Fine-grained Incident Video Retrieval with Video Similarity Learning

Georgios Kordopatis-Zilos

Submitted in partial fulfillment of the requirements of the Degree of Doctor of Philosophy

Co-supervisors: Prof. Ioannis Patras & Dr. Symeon Papadopoulos

School of of Electronic Engineering and Computer Science

Queen Mary University of London

United Kingdom

May 2020

Statement of originality

I, Georgios Kordopatis-Zilos, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author.

Signature: Georgios Kordopatis-Zilos

Date: 27/05/2020

Details of collaboration and publications:

- Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. "ViSiL: Fine-grained Spatio-Temporal Video Similarity Learning". In Proceedings of the IEEE International Conference on Computer Vision, 2019.

- Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. "FIVR: Fine-grained Incident Video Retrieval". IEEE Transactions on Multimedia, 2019.

- Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. "Finding Near-Duplicate Videos in Large-Scale Collections". Video Verification in the Fake News Era, 2019.

- Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. "Near-Duplicate Video Retrieval with Deep Metric Learning". In Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017.

- Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. "Near-Duplicate Video Retrieval by Aggregating Intermediate CNN Layers". In Proceedings of the International conference on Multimedia Modeling, 2017.

Abstract

In this thesis, we address the problem of Fine-grained Incident Video Retrieval (FIVR) using video similarity learning methods. FIVR is a video retrieval task that aims to retrieve all videos that depict the same incident given a query video – related video retrieval tasks adopt either very narrow or very broad scopes, considering only near-duplicate or same event videos. To formulate the case of same incident videos, we define three video associations taking into account the spatio-temporal spans captured by video pairs. To cover the benchmarking needs of FIVR, we construct a large-scale dataset, called FIVR-200K, consisting of 225,960 YouTube videos from major news events crawled from Wikipedia. The dataset contains four annotation labels according to FIVR definitions; hence, it can simulate several retrieval scenarios with the same video corpus. To address FIVR, we propose two video-level approaches leveraging features extracted from intermediate layers of Convolutional Neural Networks (CNN). The first is an unsupervised method that relies on a modified Bag-of-Word scheme, which generates video representations from the aggregation of the frame descriptors based on learned visual codebooks. The second is a supervised method based on Deep Metric Learning, which learns an embedding function that maps videos in a feature space where relevant video pairs are closer than the irrelevant ones. However, video-level approaches generate global video representations, losing all spatial and temporal relations between compared videos. Therefore, we propose a video similarity learning approach that captures fine-grained relations between videos for accurate similarity calculation. We train a CNN architecture to compute video-to-video similarity from refined frame-to-frame similarity matrices derived from a pairwise region-level similarity function. The proposed approaches have been extensively evaluated on FIVR-200K and other large-scale datasets, demonstrating their superiority over other video retrieval methods and highlighting the challenging aspect of the FIVR problem.

Acknowledgments

First of all, I would like to thank my supervisors Prof. Ioannis Patras and Dr. Symeon Papadopoulos, for the continuous support, advice, motivation, and patience during my studies. Also, I would like to express my gratitude to them along with Dr. Ioannis Kompatsiaris, for the great opportunity they gave me to actually start my postgraduate studies. Besides, I would like to thank the rest of the members of my supervisory team, Dr. Yi-Zhe Song and Dr. Miles Hansard, for their insightful guidance. Also, I would like to thank my former colleagues and alumni, Foteini and Christos, for their help while conducting my thesis. Last but not least, I would like to thank my friends and family, my parents and my sister, for supporting me all these years. And Eva, for her constant encouragement since the beginning of my studies.

Contents

List of Figures

List of Tables

List of abbreviations

AVR	Action Video Retrieval
BoW	Bag-of-Words
CBVR	Content-Based Video Retrieval
CNN	Convolutional Neural Network
CS	Chamfer Similarity
CSV	Complementary Scene Videos
DML	Deep Metric Learning
DNN	Deep Neural Network
DSV	Duplicate Scene Videos
EVR	Event-based Video Retrieval
FIVR	Fine-grained Incident Video Retrieval
ISV	Incident Scene Videos
MAC	Maximum Activations of Convolutions
mAP	mean Average Precision
NDV	Near-Duplicate Videos
NDVR	Near-Duplicate Video Retrieval
PCA	Principal Components Analysis
PR	Precision-Recall
SCS	Symmetric Chamfer Similarity
TD	Tensor Dot product
VCD	Video Copy Detection
ViSiL	Video Similarity Learning

Chapter 1

Introduction

Contents

1.1 Fine-grained Incident Video Retrieval

1.1.1 Problem statement and motivation

Video retrieval is a very important yet highly challenging problem that is exacerbated by the massive growth of social media applications and video sharing platforms. At the moment, YouTube reports more than two billion users, and approximately 500 hours of video content is uploaded every minute[1]. As a result of the uncontrolled number of videos published in platforms, such as YouTube, it is very common to find multiple videos about the same incident (e.g., terrorist attack, plane crash), which are either near-duplicates of some original video or simply depict the same incident from different viewpoints or at different times. Being able to efficiently retrieve all videos around an incident of interest is essential for numerous applications ranging from copy

[1] `https://www.youtube.com/yt/press/statistics.html`

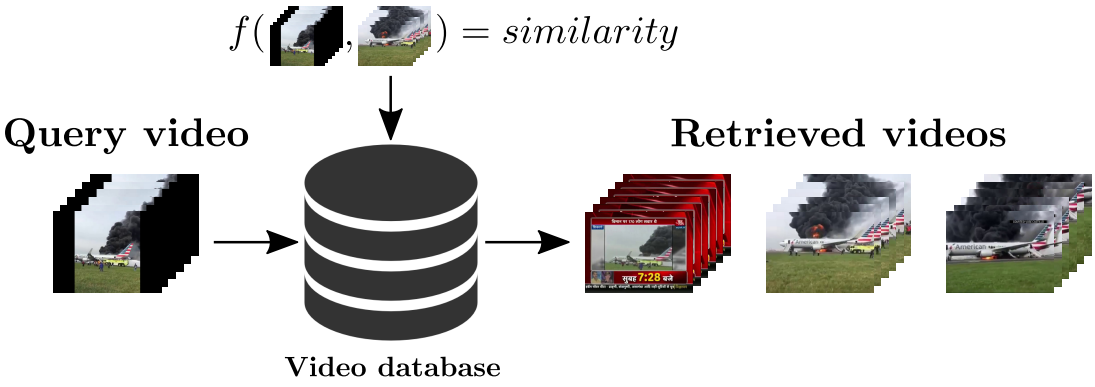


Figure 1.1: Video retrieval paradigm.

detection for copyright protection [25, 67] to event reconstruction [16, 85, 29] and news verification [54, 109].

In this thesis, we study a certain type of video retrieval, i.e., search by example, which can be formulated as follows: given a query video, retrieve all videos in a database that are related to the query. More precisely, a video retrieval system has to assess the relation of the query with all database videos and return a list of retrieved videos ranked based on their *relatedness* to the query. In the ideal scenario, all the related videos would be placed at the top ranks of the returned list, followed by the unrelated ones. However, there are several meanings and interpretations regarding related videos. In this specific line of research, the main measure considered to judge whether videos are related or not is the content-based similarity. To this end, the main challenge in the current problem setting is the definition of a pairwise function that calculates the content-based similarity between two videos and assigns higher scores to related pairs and lower to unrelated ones. Figure 1.1 illustrates an overview of the video retrieval paradigm.

However, different cases of the video retrieval problem pose different requirements. This leads to a variety of notions regarding the association between two videos and whether they are considered related to each other. For example, in the copyright protection problem, given a query video, only videos containing nearly identical copies

of the video should be retrieved. In this scenario, similar videos from the same incident (e.g., from different angles, locations, or time intervals) should be considered irrelevant and not be retrieved by the system. However, tasks such as journalistic investigations around an incident pose different requirements. In this scenario, the retrieval of videos from the same incident is of great importance. Being able to efficiently and accurately retrieve i) videos that originate from the same video source (duplicate videos) and ii) videos that capture the same incident from different viewpoints and at different times would be of great value for such tasks. In this thesis, we denote the overall problem as Fine-grained Incident Video Retrieval (FIVR) and construct a large scale dataset to simulate it. FIVR offers a single framework that contains several retrieval tasks as special cases, which derive based on the relation between videos that capture the same incident.

There are several application areas where the FIVR problem can prove relevant. A number of such relevant retrieval applications are presented in [27]. For example, news media analysis and reporting would greatly benefit from an effective solution to the FIVR problem. In a recent work, journalists from the *New York Times* [16] managed to reconstruct the timeline of the Las Vegas shootings based on content from both amateur and police videos that had been captured during the incident. In another relevant work, the research group *Forensic Architecture* [85] created a 3D video of the Grenfell Tower fire to help understand how the disaster unfolded. Moreover, Gao et al. [29] developed an approach that automatically processes a set of collected web videos and generates a short video that summarizes the storyline of an event. Other application scenarios and use cases that may benefit from solutions to the FIVR problem include safety and security applications [100, 93, 78], e.g., abnormal human behavior detection, crowd physical motion detection, forensic analysis of CCTV video. Such applications could considerably benefit from methods that, given a query video, retrieve similar videos based on the different definitions of FIVR association. Also, an adaptive video retrieval method that could be configured based on the various aspects

of the problem would facilitate such applications. The EU projects In Video Veritas (InVID)[2] and WeVerify[3] have recognised this need and put effort on that topic in order to facilitate the multimedia verification.

In this thesis, we address two fundamental associations between similar videos: a) duplicate videos and b) videos of the same incident. By duplicate videos, we refer to videos that have been captured by the same camera and depict exactly the same scene but may have undergone some visual transformations (e.g., brightness/contrast, colour, recompression, noise addition, cropping). The second type of similar videos that we consider are videos capturing the same incident. This category may be split into subcategories: a) videos that depict the same incident scene from complementary viewpoints, and b) videos that capture the same incident at different time intervals. In particular, two videos in the first category must have at least one video segment where there is temporal overlap between the depicted incident. Videos in the second subcategory need to depict the same incident but do not need to have temporal overlap. Also, the videos in the former subcategory must contain distinct visual cues, which are apparent to humans, linking two videos. However, in the latter subcategory, such cues are not mandatory to exist, and the association of two videos can be inferred through other modalities, e.g., audio signals, metadata. Moreover, even though additional modalities can be proven particularly useful in order to tackle FIVR, in this work, we focus exclusively on visual information processing and do not consider multi-modal solutions, which can be future work.

The particular use-case that we are mainly interested in is the retrieval of videos derived from breaking news events. In such scenarios, all video associations mentioned above are present, and hence all related videos have to be retrieved for practical applications. More precisely, in breaking news events, several eyewitness videos capturing the particular incident are recorded by the people involved. Such videos are usu-

[2] http://www.invid-project.eu/
[3] https://weverify.eu/

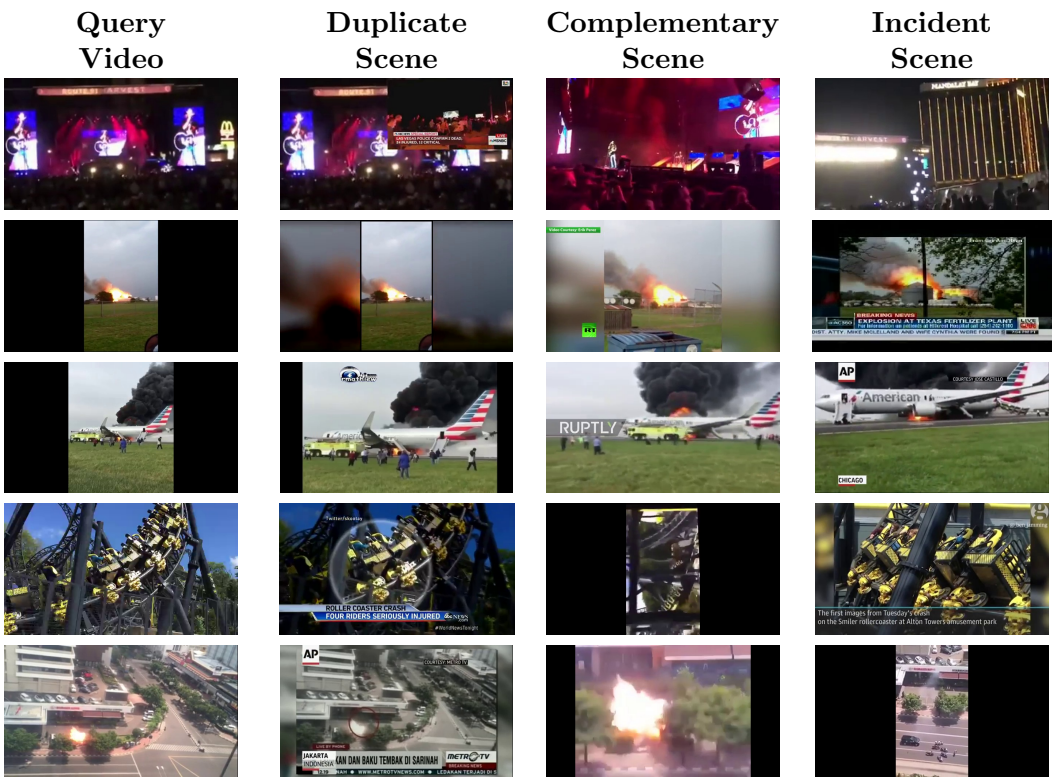


Figure 1.2: Examples of queries and retrieved associated videos from FIVR-200K.

ally reposted several times by users in social media or broadcasted by news outlets worldwide. As a result, several near-duplicate versions of the same video are generated on the web, and hence, their retrieval is of critical importance. Moreover, since many people are involved in breaking news events, there are many versions of videos capturing the same incident. Such videos are captured by different devices and are naturally from different angles or at different time intervals, but they still derive from the same incident. Therefore, their retrieval could be of high value depending on the application, e.g., news verification. Figure 1.2 illustrates several query examples from breaking news events with their related videos based on the different video associations accepted in FIVR.

1.1.2 Challenges and assumptions

There is a variety of retrieval tasks and definitions in the multimedia community in relation to the FIVR problem. These vary with respect to the degree of similarity that determines whether a pair of videos are considered related, and range from Near-Duplicate Video Retrieval (NDVR) with very narrow scope where only almost identical videos are considered positive pairs [134], to very broad definitions, such as Event-based Video Retrieval (EVR), where videos from the same event [99] or with the same semantics [13] are labelled as related. However, there does not seem to be a strong consensus among researchers about which videos are considered near-duplicate videos and none of the existing definitions addresses the retrieval of *same incident videos*. Additionally, solving the most general case of video retrieval (e.g., EVR) does not guarantee an optimal, or even satisfactory, solution for the more fine-grained cases (e.g., NDVR). Hence, in this thesis, we attempt to address these issues by providing solid definitions for all types of associations between videos related to the FIVR problem and setting up a unified framework for benchmarking the proposed methods under different retrieval settings.

Although there are a few video collections that capture different aspects of this problem, all are limited in different ways. For example, such relevant datasets include CC_WEB_VIDEO [134], VCDB [51], and EVVE [99]. The first two datasets have been collected and annotated for the problem of near-duplicate/copy detection problems, whereas the last one for the problem of event retrieval. The CC_WEB_VIDEO dataset has been used for NDVR since it comprises a number of queries that correspond to particular video subsets containing multiple near-duplicates. However, its volume is relatively small (i.e., 12,790 videos) and contains only 24 queries. Also, it lacks challenging distractors; thus, simple methods achieve close to perfect results. The VCDB [51] dataset has been compiled for partial copy detection. The main issue with this dataset is that only a limited number of videos have been annotated, i.e., there are

only 528 videos in the core dataset. The EVVE [99] simulates the event-based video retrieval problem. The definition used to determine related videos is very broad. A pair of videos is not required to have spatial and temporal consistency in order to be labelled as positives, so long they capture the same event, i.e., a happening that can be re-occurring or not. Also, the dataset contains annotation only for the videos from the same event and not for near-duplicate cases.

Moreover, the research community has invested considerable effort in the development of video retrieval methods. Several approaches have been proposed in the literature of the related retrieval fields, which can be roughly classified into three categories based on the level of similarity considered for video ranking, i.e., (i) video-level, (ii) frame-level, and (iii) filter-and-refine similarity.

The methods in the first category aggregate/pool frame-level features into a single video-level representation on which subsequently one can calculate a similarity measure [134, 113, 30, 77, 115]. These methods are very fast due to the compact size of the video representation, but they disregard the spatial and temporal structure of the visual similarity, as the aggregation of features is influenced by clutter and irrelevant content. This leads to worse performance in comparison to the methods of the other two categories. The methods in the second category extract frame-level features from videos and attempt to take into account the temporal sequence of frames in the similarity computation [119, 25, 99, 11, 28]. Such approaches lead to a significant performance gain due to the fine-grained comparison between the two videos. Their major drawback is that they are computationally expensive due to the extensive video comparison; thus, the querying process is significantly slower than the video-level methods. This might be impractical depending on the application scenario, e.g., for online retrieval systems where the similarity of newly submitted queries can not be pre-computed. In the third category, researchers sought for hybrid approaches by combining a video- and a frame-level approach in a single framework in a filter-and-refined

scheme [134, 122, 22, 138]. Typical methods first compare videos based on a video-level approach to filter the irrelevant cases and then refine the similarity estimation based on a computationally expensive frame-level approach. These methods offer a balance between retrieval performance and computational speed. However, they usually are not able to squeeze the full potential of the employed frame-level approach due to the erroneous filtering of relevant video.

Yet another limitation of the methods in the literature is the lack of flexibility, in a sense that they return only almost exact copies of the input videos, and in some cases, they are not catered for the specific requirements of the problem (e.g., when a user needs to look for partial-duplicates or videos from different viewpoints). Such property is essential in the case of the FIVR problem. Another issue of many state-of-the-art methods is that they adopt a dataset-specific approach: the same dataset is used for both development and evaluation. This leads to specialized solutions that typically exhibit poor performance when used in different video collections. Finally, a considerable limitation of the recently proposed methods is that they do not consider the spatio-temporal structure of video similarity. The exploitation of such information for similarity calculation results in significant performance improvement.

1.2 Aims and objectives

In this thesis, we aim to formally introduce the Fine-grained Incident Video Retrieval (FIVR) problem and address it with an efficient and accurate video retrieval method. While the task of video retrieval has presented much progress during the last years, it remains a timely topic with open research questions and practical application. As briefly discussed in the previous subsections, there are several related definitions proposed by the retrieval community. In addition, the performance of state-of-the-art video retrieval systems cannot be considered satisfactory, especially in FIVR settings, which highlights the difficulty of this problem and the need for developing novel ap-

proaches in this field. The challenge in building an ultimate solution for the FIVR problem is to: offer flexibility with respect to the definition of what are relevant videos, achieve very high precision and recall scores, and at the same time, provide the possibility for scalable indexing of massive multimedia collections and have low response times. Nevertheless, our goal in this work focuses on retrieval performance rather than computational time and scalability. Therefore, we aim to build an as accurate as possible retrieval system that is able to achieve very high retrieval performance.

Our first objective is to collect and annotate a challenging dataset that will serve the benchmarking needs for different variants of the problem of FIVR. To accurately represent the problem, this dataset is composed of user-generated videos related to a large number of real-world events. The events are selected to be of the same nature for the collected videos to be visually similar and thus to include more challenging distractors in the dataset. Moreover, a number of videos have been selected as benchmark queries. We set up a principled process to find queries that have several duplicates and videos from the same incident, which serve as relevant video cases. At the same time, there should also be many visually similar distractor videos from different events to make the retrieval of relevant videos more challenging.

The second objective of this research is to develop effective video retrieval approaches for FIVR. Motivated by the excellent performance of deep learning in a wide variety of multimedia problems, we develop video retrieval approaches that incorporate deep learning and can be used in different application scenarios. Initially, we focus on the development of a video-level approach, which provides a fast solution for retrieval tasks. We first build a method that does not need labelled data, and as a result, it can be applied to any video corpus. However, the developed approach has several limitations, i.e., volatile performance on unseen data, hard to be retrained with new videos, and does not provide flexibility with respect to FIVR definitions. Thus, we develop a supervised solution that is based on a learning scheme that gives the opportunity to

be trained in various scenarios. Finally, to significantly improve the system's retrieval performance, we propose a frame-level approach based on video similarity learning. This approach considers both the spatial (intra-frame) and temporal (inter-frame) structure of the visual similarity in order to assign a similarity score between two compared videos.

1.3 Thesis contributions

Our contributions are:

- The formalization of the FIVR problem and a large-scale dataset that covers its benchmarking needs, which we call FIVR-200K. We provide formal definitions for three types of video associations between related videos, i.e., duplicate, complementary, and incident scene videos, considering the spatio-temporal spans captured by videos. The dataset comprises of 225,960 videos from YouTube collected based on major news events from recent years crawled from Wikipedia. Additionally, 100 videos are selected to serve as queries selected based on an automatic pipeline that estimates the suitability as a benchmark of the videos in the dataset. The dataset contains four annotation labels derived based on FIVR definitions; thus, the benchmarked methods can be evaluated in various retrieval settings using the same video corpus. We also conduct a comprehensive experimental study comparing state-of-the-art approaches implemented with handcrafted and deep features. The study highlights the challenging aspect of the collected dataset and the difficulty of the FIVR problem.

- Two video-level approaches that generate global video representations which facilitate fast retrieval. In contrast to the common practice in video retrieval literature that used handcrafted features, we employ deep learning features extracted from intermediate layers of Convolutional Neural Networks (CNNs) to build both of our approaches. For the first method, we propose an unsupervised approach that is a variation of the traditional Bag-of-Words (BoW) scheme.

It employs a layer aggregation technique that leads to improved retrieval performance. To overcome several limitations of the BoW approach (e.g., volatile performance on unseen data), we build a second method based on a supervised solution that leverages Deep Metric Learning (DML). A significant benefit of the DML scheme is that it gives the opportunity to be trained in various scenarios; hence, it provides us with the required flexibility with respect to the FIVR definition.

- A Video Similarity Learning (ViSiL) method that considers fine-grained spatio-temporal relations between the compared videos that offers accurate similarity estimation. In contrast to the current methods in the state-of-the-art that disregard the spatial or temporal structure of videos during similarity calculation, we propose a method that considers such information, which leads to significant performance improvement. More precisely, the method consists of two carefully crafted components for frame-to-frame and video-to-video similarity calculation. For the first component, we build a function that takes into consideration region-to-region pairwise similarities during similarity computation. For the second component, we train a network that analyses the frame-to-frame similarity matrices and captures the temporal structure of the frame-level similarity to robustly establish high similarities between relevant videos. Our proposed method demonstrates significant performance gain on several video retrieval problems.

Our methods have been extensively evaluated on FIVR-200K and other large-scale datasets, demonstrating their superiority over other video retrieval methods and on a large number of video retrieval tasks. Additionally, our experimental study highlights the challenging aspect of the FIVR problem. However, we do not evaluate the scaling of our approaches in massive datasets with millions of videos; hence, this remains open for future research.

1.4 Structure of the thesis

In Chapter 2, we review the related literature in research fields related to the FIVR problem, and we provide an outline of the major trends in these fields. We present the existing definitions regarding the different types of video associations and the related research field. Then, the video retrieval approaches are classified based on the level that video similarity is calculated, and the most indicative approaches from each category are discussed. Finally, we present the evaluation datasets that are traditionally used as benchmarks.

In Chapter 3, we introduce the FIVR problem, where we provide the definitions for the related videos and the considered associations. We present all the underlying processes for the collection, annotation, and query selection built for the composition of the large-scale FIVR-200K dataset. Also, we benchmark a variety of visual descriptors and aggregation techniques that have been used by the state-of-the-art, including our proposed video-level approaches.

In Chapter 4, we present the two proposed video-level methods, BoW and DML approach, where the process for the generation of global video representations is explained in detail. Also, the feature extraction from the intermediate CNN layers is described. A comprehensive evaluation is reported with several settings of the proposed methods.

In Chapter 5, we introduce the proposed ViSiL approach. We provide an in-depth explanation and analysis for the fundamental functions for frame-to-frame and video-to-video similarity calculation and learning. Moreover, we describe the setup for the training of the network. A comprehensive experimental study is conducted on four video problems and six datasets.

The thesis concludes with Chapter 6, where we summarize the findings of the experimental studies and present our conclusions on the progress of the ongoing work. Moreover, we identify some problem aspects where there is still space for further re-

search and draw directions for future work.

Chapter 2

FIVR and Related Work in Video Retrieval

Contents

In this chapter, we review some of the most representative works in the literature of video retrieval, focusing on the research fields related to FIVR, such as Near-Duplicate Video Retrieval (NDVR). We aim to cover state-of-the-art studies in these fields, present the existing definitions of related videos, highlight the limitations of existing methods and provide a comprehensive view of the research areas for the topics addressed in this thesis. The chapter has been divided into three sections aiming to place FIVR with respect to the existing retrieval problems, highlight weaknesses of existing approaches, and show how the proposed approaches in this thesis can go beyond the state-of-the-art. Section 2.1 covers the definition and the related research fields in the literature. Section 2.2 presents a variety of video retrieval approaches, classified based on the level of video similarity calculation. Finally, Section 2.3 summarizes the

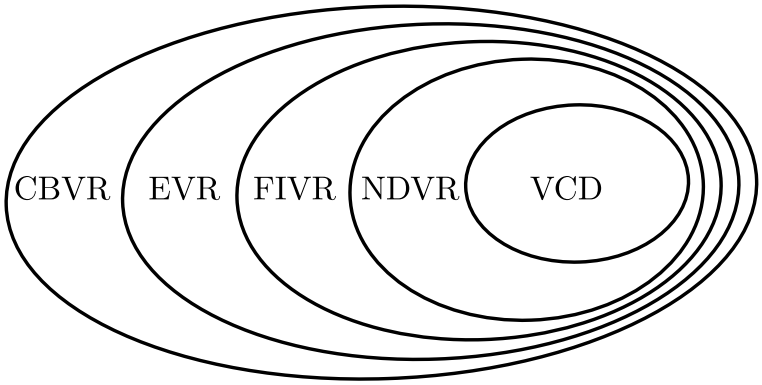


Figure 2.1: Venn diagram that illustrates the relationship between FIVR and other related retrieval tasks with respect to the considered types of association between related videos. VCD stands for Video Copy Detection, NDVR for Near-Duplicate Video Retrieval, FIVR for Fine-grained Incident Video Retrieval, EVR for Event-based Video Retrieval, and CBVR for Content-Based Video Retrieval.

existing evaluation datasets that are traditionally used as a benchmark to measure the performance of the proposed video retrieval methods.

2.1 Definition and related retrieval tasks

Video retrieval is a very challenging problem and has attracted increasing research interest in recent years. Several variations of the video retrieval paradigm have been proposed in the literature. In this section, we provide the existing definitions for four related research problems. Figure 2.1 illustrates the relationship of the related retrieval problems to FIVR with respect to the considered types of association between related videos.

Near-Duplicate Video Retrieval (NDVR) is the most closely related research field to FIVR. The scope of NDVR is the retrieval of near-duplicate videos (NDVs). However, there does not seem to be a strong consensus among researchers about which videos are considered NDVs. There is a variety of definitions and interpretations, as pointed out in [81, 107]. The representative and predominant definitions are those proposed in Wu et al. [134] and Shen et al. [106]. These vary with respect to the level of

resemblance that determines whether a pair of videos are considered to be NDVs. Wu et al. [134] adopted the most narrow scope among the definitions. In essence, NDVs were considered only those that are identical or approximately identical videos, i.e., close to being exact duplicates of each other, but different in terms of file format, encoding parameters, minor photometric variations, editing operations, length, and other modifications. By contrast, the definition in Shen et al. [106] extended this to videos with the same semantic content but different in various aspects introduced during capturing time, including photometric or geometric settings, e.g., different camera viewpoint and setting, lighting condition, background. Yet, both definitions are relatively narrow, considering only near-duplicate cases, and does not cover the retrieval of the same incident videos adequately.

Additionally, Video Copy Detection (VCD) [69] is closely related to NDVR and, as a result, to FIVR. The definition of video copies in VCD is very close to the one of NDVR, yet it is slightly narrower. Videos derived from the same source video and differing only with respect to photometric or geometric transformations are considered as copies based on Law-To et al. [69]. Also, the objective of a VCD approach is to identify the copied videos and detect the particular video segments that have been copied. Thus, the proposed VCD solutions might be inapplicable to video retrieval settings. A comprehensive overview of VCD approaches is provided in [130].

Another related research field is the Event-based Video Retrieval (EVR) problem. The problem was formulated in terms of definition and dataset by Revaud et al. [99]. The objective of this problem is the retrieval of the videos that captures the same event. However, the definition of the same event videos is very broad, including videos that have either spatial or temporal relationships. Based on our definition provided in Chapter 3, two videos are considered related when they originate from the same spatio-temporal span, i.e., have to be spatially and temporally related. Hence, the proposed definition and dataset do not serve our needs.

Content-Based Video Retrieval adopts the broadest definition among the ones in the literature. The problem was introduced in Basharat et al. [13] and considered that two videos are related when they depict the same semantic concept, which may occur under different illumination, appearance, scene settings, camera motion, etc. For example, videos that illustrate a person riding a bicycle are considered related, even if there are variations such as different viewpoints, sizes, appearances, bicycle types, and camera motions.

At this point, one may wonder whether a solution to the most general problem would ultimately solve all other narrower problems. However, solving the most general case of video retrieval (e.g., CBVR) does not guarantee an optimal, or even satisfactory, solution for the more fine-grained cases (e.g., FIVR or NDVR). For example, a solution built to generate global video representations that finely encode the videos based on the depicted concepts would have a very competitive performance for CBVR. However, it would fail in a more fine-grained scenario such as NDVR, where there are long near-duplicate videos with only short overlapping content. In this case, the videos depicting the same concept would distract the retrieval process, whose goal is to retrieve only the near-duplicates. Thus, a specialized solution that processes the videos in frame-level would be more suitable. On the other hand, a rigorous system that detects only near-duplicate content would not work for more general cases, where the objective is to retrieve videos with more abstract relations. To this end, we do not search for a universal solution that solves all retrieval problems at once.

2.2 Video retrieval approaches

Based onThe video retrieval approaches can be classified based on the level of similarity considered to determine the video ranking into video-level (Section 2.2.1), frame-level (Section 2.2.2) and filter-and-refine similarity (Section 2.2.3)

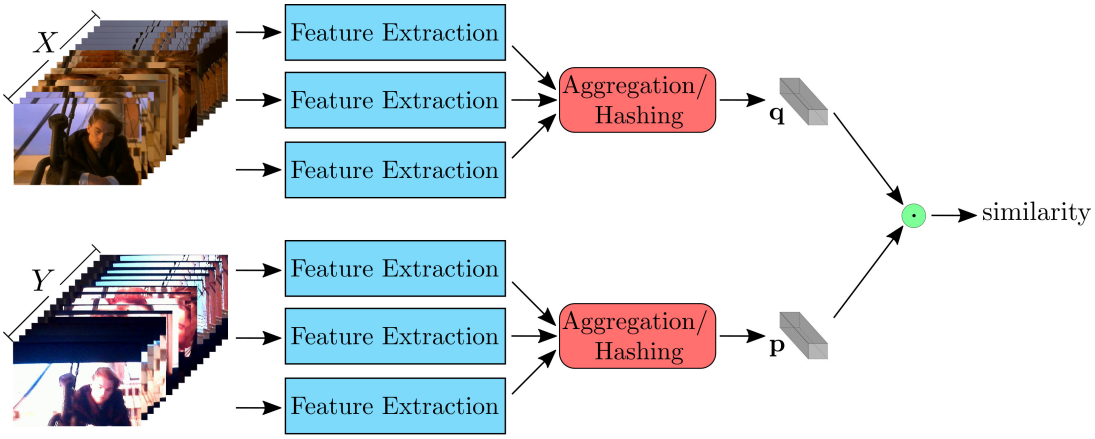


Figure 2.2: Overview of video-level approaches. After the feature extraction, the frame-level features are combined into a global video representation through an aggregation scheme or a hashing function, which is used for similarity calculation.

2.2.1 Video-level similarity

Video-level approaches have been developed to deal with web-scale retrieval. In such approaches, videos are usually represented with a global signature, such as an aggregated real-value feature vector or a binary hash code. The video similarity is computed based on the global video representations. Figure 2.2 provides an overview of a typical video-level approach.

A common process to generate a global video representation is by the combination of visual features extracted from video frames into a single feature vector. The global representations derive from the application of an aggregation/pooling function, and the video similarity is usually calculated based on Euclidean distance, cosine similarity, or Jaccard similarity. Several research works have employed this similarity calculation scheme [134, 82, 106, 103, 43, 59, 48, 30, 31, 71, 143, 70]. Wu et al. [134] introduced a simple approach for the video signature generation. They extracted HSV features from the video keyframes and averaged them to create a single vector. The distance between two video signatures was computed based on their Euclidean distance. Huang et al. [43] employed Principal Component Analysis (PCA) [133] over the colour histograms of the video frames and generated a video representation model

called Bounded Coordinate System (BCS). The scaling and rotations of the BCS vectors were considered for the similarity calculation between two videos. Shang et al. [103] introduced compact spatio-temporal features based on Local Binary Patterns (LBP) [145], called STF-LBP, to represent videos and constructed a modified inverted file index based on a Bag-of-Word (BoW) scheme [18]. These spatio-temporal features were extracted based on a feature selection and w-shingling scheme, which uses sets of unique n-grams, each of which is composed of consecutive visual words. They adopted Jaccard similarity to rank videos. Cai et al. [18] presented a large-scale BoW approach by applying a scalable K-means clustering technique on the color correlograms [41] of a sample of frames and using inverted file indexing [111] for the fast retrieval of candidate videos. They used cosine similarity to measure the similarity between two candidate videos. Kim et al. [59] propose a video fingerprint based on region binary patterns, which is robust against rotation and flipping transformations. They extract two complementary region binary patterns from several rings in keyframes, which are combined into a single video fingerprint used to measure similarity. Jiang et al. [48] extracted frame representations from the fully connected layer of a CNN, which are aggregated to a video-level signature by applying global average pooling. The Euclidean distance measures the similarity between videos. Goa et al. [30, 31] extracted a video imprint for the entire video based on an alignment procedure of CNN features that exploits the temporal correlations and removes feature redundancies across frames. They sum-aggregate the entire video imprint to extract a global vector and use cosine similarity to measure similarity. Lee et al. [71] proposed a deep learning architecture that maps videos based on their audio-visual content, onto an embedding space that preserves video-to-video relationships. They experimented with different fusion schemes to combine video and audio features. Videos were ranked based on the dot product of the video embeddings, and the network was trained by optimizing the triplet loss function. In [70], they extended their work by building a training scheme based on hierarchical graph clusters, which are used for negative sampling and

pseudo-classification labeling.

A popular direction is the generation of a hash code for the entire video. Similar to the real-value approaches, video hashing methods apply a hashing function that aggregates the video information and generates a binary hash code for the entire video. The Hamming distance is utilized as a similarity measure. Many video hashing methods have been proposed in the literature [113, 114, 91, 77, 35, 34, 55, 115, 112, 90, 73, 141]. Song et al. [113] presented an approach for Multiple Feature Hashing (MFH) based on an unsupervised method that employed multiple frame features (i.e., LBP and HSV features) and learned a group of hash functions that map the video frame descriptors into the Hamming space. The video signatures were generated by averaging the keyframe hash codes. They extended their approach in [114] by including information of the frame groups into the objective function, so as to introduce temporal information in the learning process of the hash functions, which led to a marginal performance increase. Hao et al. [35] combined multiple frame features to learn a group of mapping functions by minimizing the difference of the probability distribution of frame adjacencies between the original and embedded Hamming space based on the Kullback-Leibler (KL) divergence. They extended their work in [34] by employing t-distribution to generate relaxed hash codes. Jing et al. [55] proposed a supervised hashing method called Global-View Hashing (GVH), which utilized relations among multiple features of video keyframes. They projected all features into a common space and learned multi-bit hash codes for each video using only one hash function. Liong et al. [77] employed a CNN architecture to learn binary codes for the entire video and trained it end-to-end based on the pair-wise distance of the generated hash codes and video class labels. Song et al. [115] built a self-supervised video hashing system, able to capture the temporal relation between frames using an encoder-decoder scheme based on a Recurrent Neural Network (RNN). The network is trained with reconstruction loss on the encoder-decoder setup and a neighborhood loss that enforces the preservation of the neighborhood structure. Nie et al. [90] proposed a supervised hashing scheme that

jointly learns multiple hashing functions that preserve the global and local structures of multiple features in the Hamming space. A multi-bit hash function is learned based on generalized eigenvalue decomposition that learns multiple hash functions within a single step. Li et al. [73] proposed a neighborhood attention mechanism which focuses on useful content in video frame conditioned with the neighborhood information. They employed an RNN-based reconstruction scheme to implement neighborhood attention, which learns a hashing function that maps similar videos to similar binary codes. Finally, Yuan et al. [141] proposed the central similarity, which is a global similarity metric, that encourages the hash codes of similar data pairs to approach a common center and those for dissimilar pairs to converge to different centers. They trained end-to-end a CNN network with a hashing layer by optimizing their central similarity, in order to generate video hash codes.

To sum up, video-level approaches capture the overall video information and facilitates video indexing and searching due to the compact video representation. Thus, video querying is very fast in such methods. However, the loss of local information in the video-level global signature has a considerable impact on the performance of such approaches [134, 22, 11, 64, 62, 104], making it difficult to distinguish two irrelevant videos with similar content. Hence very different videos may have similar global signatures, which may result in misleading decisions. These methods are typically outperformed by the ones of the other two categories. Also, methods that use deep learning, either for feature extraction or generation of global video representation, were limited by the time that we conducted research for this thesis. Besides, only a few recent works proposed supervised learning solutions [77, 71, 70] for the video retrieval problem. Supervised solutions provide flexibility with respect to the definition of related videos and offer a more robust solution when applied on unseen data. Finally, deep learning methods [70, 73, 141] achieve the best results among video-level methods in the related video retrieval fields.

To this end, in this thesis, we build one unsupervised and one supervised video-level method based on global features that leverage deep learning and outperform the state-of-the-art of video-level methods with a significant margin (Chapter 4). The first method is an unsupervised approach based on a BoW scheme [111]. We extract global frame representations for the video frames from the intermediate layers [124, 146] of a pretrained CNN. Then, we devise two aggregation schemes, i.e., a vector and a layer aggregation, to generate global video representations. Our approach shares many similarities with the method presented in [18] since both of them employ global frame representations and generate global video representations from BoW. However, in contrast to [18], we use CNN features. At the time of publication, it was the first time that such features were successfully employed in the context of the related video retrieval tasks. Moreover, the layer aggregation is a modified version of the traditional BoW, where multiple codebooks are utilized to map the feature vector extracted from the CNN layers into multiple visual words. This modification leads to a significant performance increase. The second method is a supervised approach that employs Deep Metric Learning (DML). Again, we utilize the same features from intermediate CNN layers concatenated in the channel dimension. We train a Deep Neural Network (DNN) to approximate an embedding function that maps videos to a feature space where the related videos are closer than the irrelevant ones. We train our network with triplet loss function and a triplet-generation process based on hard negative examples. At the time of publication, the DML scheme had been employed in other similar computer vision problems, e.g., image retrieval [128, 95], face recognition/retrieval [101], but not in one of the related fields, being a recent trend in state-of-the-art by the time of writing [77, 71, 11, 64, 70, 141, 104]. Therefore, our main novelty is the adaptation of the DML pipeline to the domain of video processing.

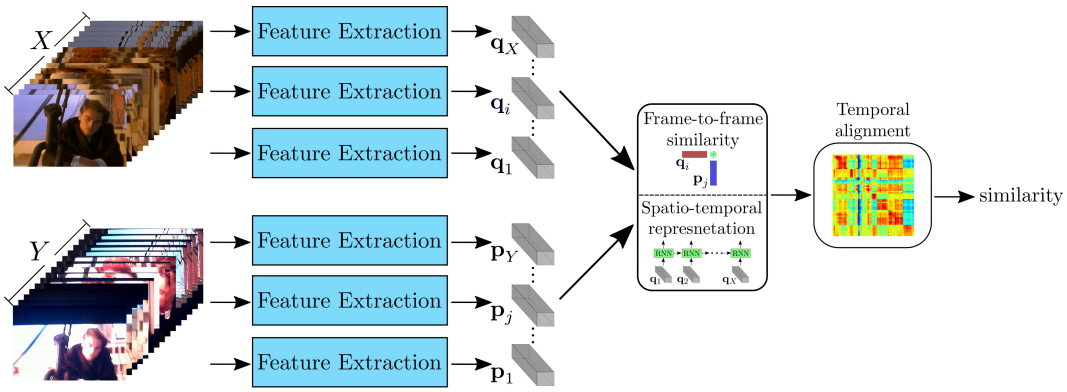


Figure 2.3: Overview of frame-level approaches. After the feature extraction, global frame representations are generated, and a frame-to-frame similarity calculation step is performed, or a spatio-temporal representation is extracted. Then, temporal alignment is applied that assesses the video similarity.

2.2.2 Frame-level similarity

In the case of frame-level approaches, the video similarity is determined by the comparison between individual video frames or sequences. Typical frame-level approaches calculate the similarity between videos based on frame-to-frame or spatio-temporal similarity functions. Figure 2.3 provides an overview of a typical frame-level approach.

Methods that employ frame-to-frame similarity calculation usually extract global representations for video frames and then employ temporal alignment algorithms to compute similarity at the video-level. Some typical methods are [21, 140, 119, 25, 132, 80, 51, 58, 52, 129, 86, 76, 74, 104]. Tan et al. [119] proposed a graph-based Temporal Network (TN) structure generated through keypoint frame matching. They embedded temporal constraints into a network structure and formulated the partial video alignment problem into a network flow problem. The similarity between two compared videos was calculated based on the longest path in the generated temporal network. Douze et al. [25] proposed an approach to align matched frames by means of a temporal Hough transform. They extracted SIFT [83] and CS-LBP [37] descriptors based on Hessian-Affine regions [88], to create a BoW codebook [111] for Hamming Embedding with weak geometric consistency [45]. Using post-filtering, they verified retrieved

matches with spatio-temporal constraints and devised the so-called Temporal Hough Voting (THV). Several recent works have employed modifications of the two aforementioned approaches for the problem of partial-copy detection, combining it with CNN features [52, 129, 40]. Jiang et al. [52] employed a pre-trained CNN to extract global features for the video frames, and they also trained another CNN with pairs of image patches that captures the local information of frames. They experimented with TN and THV in order to detect the copied video segments. Wang et al. [129] proposed a compact video representation by combining features extracted from pre-trained CNN architectures with sparse coding to encode them into a fixed-length vector. To determine the copied video segments, they constructed TNs based on the Euclidean distance between the extracted features. Hu et al. [40] trained a siamese architecture consisting of a CNN+RNN with contrastive loss function and employed TNs to calculate video similarity. Another popular solution is based on Dynamic Programming (DP) [22, 79]. Such works calculated the similarity matrix between all frame pairs and extracted the diagonal blocks with the largest similarity. To increase flexibility, they also allowed limited horizontal and vertical movements. In some recent works, Guzman-Zavaleta et al. [32] trained a reinforcement learning system for video copy detection. They developed a decision strategy by adapting the Q-learning algorithm [131] to detect the boundings of the overlapping segments. Liang et al. [74] proposed an unsupervised teacher-student set up to train a feature extraction CNN on the target dataset. They developed an algorithm to assess the similarity between the query and database videos based on the most similar database frames to the query ones. Finally, Shao et al. [104] proposed a temporal context aggregation framework for video representation learning that captures long-range temporal information between frame-level features. They use the transformer architecture [127], which is based on self-attention mechanism, and train it with contrasting learning. They evaluate their network in both video-level and frame-level settings, achieving their best performance with the use of Chamfer Similarity, proposed in our work.

Another line of research considers spatio-temporal video representation and matching to improve the performance by exploiting not only the spatial information of frames but also the temporal relations to generate video representations and calculate similarity. Indicative spatio-temporal works include [42, 144, 98, 99, 135, 94, 149, 11, 28, 72, 3]. In some early works, Huang et al. [42] proposed a one-dimensional Video Distance Trajectory (VDT) to monitor the continuous changes of consecutive frames with respect to a reference point, which is further segmented and represented by a sequence of compact signatures called Linear Smoothing Functions (LSFs). They measured video similarity with a scheme that extends edit distance, which was applied to the extracted representations. Wu and Aizawa [135] proposed a self-similarity-based feature representation called Self-Similarity Belt (SSBelt), which derived from self-similarity matrices. They measure video similarity via coarse histogram matching of the video representations and a refinement step based on flip-invariant feature representations. A popular direction is to use the Fourier transform in a way that accounts for the temporal structure of video similarity [99, 94, 11, 136]. Revaud et al. [99] proposed the Circulant Temporal Encoding (CTE) that encodes the frame features in a spatio-temporal representation with Fourier transform and thus compares videos in the frequency domain based on the properties of circulant matrices. Poullot et al. [94] introduced the Temporal Matching Kernel (TMK) that encodes sequences of frames with periodic kernels that take into account the frame descriptor and timestamp. A score function was introduced for video matching that maximizes both the similarity score and the relative time offset by considering all possible relative timestamps. Baraldi et al. [11] built a deep learning layer component based on TMK and set up a training process to learn the feature transform coefficients in the Fourier domain using a triplet loss that takes into account both the video similarity score and the temporal alignment. Finally, some recent works [28, 72, 3] employ deep learning to solve the problem of the detection of overlapping video segments. Feng et al. [28] developed an approach based on cross gated bilinear matching for video re-localization. They employed C3D features [125]

and built a multi-layer recurrent architecture that matches videos through attention weighting and factorized bilinear matching to locate related video parts. Li et al. [72] built multiple two-class classifiers based on a 3D-CNN architecture for video copy detection. They trained several binary classifiers that constitute a parallel classification model that detects different video transformations. Finally, Abobeah et al. [3] proposed a bi-directional attention model for video alignment. They extracted CNN features aggregated with location-aware VLAD [84], and then built a bidirectional Long Short-Term Memory (bi-LSTM) [39] weighed with an attention mechanism [102] to automatically detect the starting and ending point of the overlapping segments between two videos.

Overall, the frame-level similarity approaches extract fine-grained information from videos, leading to a significant performance increase in comparison to video-level methods. Their major drawback is that they are computationally expensive due to the extensive comparison of all video frames or sequences of video pairs; thus, the retrieval process is significantly slower than the video-level matching approaches. Additionally, a promising direction is exploiting better the spatial and temporal structure of videos in the similarity calculation [25, 51, 52, 3]. However, recent approaches either focus on the spatial processing of frames and completely disregard temporal information [86, 76, 74], or consider global frame representations (essentially discarding spatial information) and then consider the temporal alignment among such frame representations [11, 28, 74]. None existing work proposes spatio-temporal solutions that considered both the spatial structure of frames and temporal structures of videos.

In this thesis, we attempt to overcome these limitations and propose a frame-to-frame method that considers fine-grained spatio-temporal relation of videos during similarity computation (Chapter 5). We devise a novel frame-to-frame similarity calculation scheme that captures similarities at the region level, which leads to significant performance improvement. Also, we build a supervised video-to-video similarity cal-

culation scheme that analyses the frame-to-frame similarity matrix through a CNN network, which robustly establishes high similarities between video segments of the compared videos. Some works in the state-of-the-art proposed approaches with the utilization of similar frame-to-frame similarity matrices [22, 79]. These solutions are not capable of capturing a large variety of temporal similarity patterns due to their rigid aggregation approach. By contrast, the proposed approach learns the similarity patterns with a CNN subnet that operates on the similarity matrix between the frame pairs. However, since computational efficiency is not our primary goal in this work, the proposed method does not address the main disadvantage of frame-level methods in comparison to video-level ones, which is the high computational time. Yet, it achieves state-of-the-art performance on several retrieval tasks and datasets.

2.2.3 Filter-and-refine

To overcome the bottleneck of video-level approaches and to achieve efficient video retrieval implementations, researchers developed hybrid approaches by combining the advantages of frame-level and video-level methods.

Typical filter-and-refine methods deploy a video-level method to quickly discard videos with low similarity scores as they considered irrelevants and then apply on the remaining videos a frame-level algorithm for refined similarity calculation. One of the earliest methods is the [134], where the author generated video signatures by averaging the HSV histograms of keyframes. Then, they applied a hierarchical scheme to filter out irrelevant videos and apply a computationally heavy similarity scheme based on local feature descriptors. Zhou et al. [147] proposed a video representation based on a 3D structure tensor called Adaptive Structure Video Tensor (ASVT) that is used to calculate similarity based on a Hamming distance extension. For the filtering step, they devised a dimensionality reduction technique for efficient indexing. Several filter-and-refine methods in the literature extracts multimodal features from videos [123, 49, 121, 122]. These methods employed various features, i.e., local visual features (SIFT

[83], SURF [14]), global visual features (DCT [5]), and audio features (WASF [20]). Also, they used Bag-of-Words (BoW) [111] scheme and Locality Sensitive Hashing (LSH) [23] for aggregation. Jiang et al. [49] presented a soft cascade framework based on the hashed features to filter out irrelevant videos. Then, they applied a temporal pyramid matching algorithm to determine the similarity between video sequences. Tian et al. [122] extended the multimodal cascading framework, including the concept of transformation-awareness, copy units, and soft-decision boundary. Moreover, many researchers combine a video-level BoW scheme as the filtering step with frame-to-frame similarity calculation as the refinement [22, 79, 6, 148, 75]. Chou et al. [22] proposed a spatio-temporal indexing structure utilizing index patterns, termed Pattern-based Index Tree (PI-tree), to filter irrelevant videos. In the refining stage, an Dynamic Programming scheme was devised to localize near-duplicate segments and to re-rank results of the filter stage. Yang et al. [138] proposed a multi-scale video sequence matching method, which gradually detected and located similar segments between videos from coarse to fine scales. Given a query, they used a maximum weight matching algorithm to select candidate videos in the coarser scale, and then they extract the similar video segments in the middle scale. In the fine scale, they used bi-directional scanning to check the matching similarity of video parts to localize near-duplicate segments. Zhou et al. [148] extracted spatial and temporal representations for the video sequences based on CNN features. They organized videos in a BoW scheme and an inverted file structure [111] for the filtering step. They refine similarity calculation based on the cosine distance between the extracted features. Finally, Liang et al. [75] employed a filtering stage based on the concepts depicted in the video frames that derive from a trained classifier. Then, they measure similarity based on a BoW scheme that uses CNN features pooled in the temporal dimension.

In this way, filter-and-refine methods take advantage of the fast retrieval of the video-level approaches to filter a large number of videos, considering them as dissimilar, and then apply computationally expensive frame-level similarity calculation techniques

Table 2.1: Comparison of FIVR with existing datasets and retrieval tasks. UGV stands for User-Generated Videos.

Dataset	Queries	Videos	Hours	UGV	Task
CC_WEB_VIDEO [134]	24	12,790	551	✓	NDVR
UQ_VIDEO [113]	24	169,952	N/A	✓	NDVR
SVD [50]	1,206	562,013	2,704	✓	NDVR
MUSCLE-VCD [69]	18	101	100	✗	VCD
TRECVID-CBCD [67]	11,256	11,503	420	✗	VCD
VCDB [51]	528	100,528	2,038	✓	VCD
EVVE [99]	620	102,375	5,536	✓	EVR
FIVR-200K	100	225,960	7,100	✓	FIVR

for improved retrieval performance. Such approaches may sacrifice performance for faster video retrieval. Hence, they offer a balance between speed and accuracy. Yet, they heavily depend on the performance of the individual video-level and frame-level approach. Also, the adaptation of the developed methods with a filter-and-refine setup can be relatively straightforward, e.g., by setting a similarity threshold. Thus, we do not invest any effort in the development of such a method in this thesis.

2.3 Benchmark datasets

Although there are a few video collections that capture different aspects of this problem, all of them are limited in different ways. More specifically, related datasets include CC_WEB_VIDEO [134], UQ_VIDEO [113], SVD [50], MUSCLE-VCD [69], TRECVID-CBCD [67], VCDB [51] and EVVE [99]. The first three datasets were collected for the problem of near-duplicate video retrieval, the next three for the video copy detection problem, and the last one for the problem of event retrieval. The query videos for the MUSCLE-VCD and TRECVID-CBCD datasets were artificially generated, i.e., the queries have been synthetically generated with the manual application of predefined transformations. In contrast, the rest of the datasets contain actual user-generated videos as queries. Table 2.1 provides an overview of the aforementioned datasets and associated retrieval tasks.

The most relevant and widely used dataset is the CC_WEB_VIDEO [134]. The dataset consists of user-generated videos collected from the Internet. In particular, it contains a total of 12,790 videos consisting of 397,965 keyframes. The videos were collected by submitting 24 popular text queries to popular video sharing websites (YouTube, Google Video, and Yahoo! Video). For every query, a set of video clips were aggregated, and the most popular video was considered as the query video. Subsequently, all retrieved videos in the video sets were manually annotated by three annotators based on their near-duplicate relation to the query video. The near-duplicate rate of the collected sets ranges from 6% to 93%. On average, 27% of the videos in each set are considered near-duplicates. The main limitation of the dataset is that its volume and query set are relatively small (12,790 videos and 24 queries). Also, it lacks challenging distractors, given that the queries are very different from each other, resulting in relatively simple approaches achieving close to perfect performance, which can be misleading.

Several variations of the CC_WEB_VIDEO dataset have been developed by researchers in the NDVR field [103, 113, 18, 22]. To make the NDVR problem more challenging and benchmark the scalability of their approaches, researchers usually extend the core CC_WEB_VIDEO dataset with thousands of distractor videos [113, 22]. The most well-known and publicly available dataset that has been created through this process is UQ_VIDEO [113]. For the composition of the background dataset, they chose the 400 most popular queries based on Google Zeitgeist Archives from the years 2004 to 2009. Each query was submitted to YouTube, and up to 1,000 video results were collected. After filtering out videos of duration longer than 10 minutes, the combined dataset is composed of 169,952 videos (including those of the CC_WEB_VIDEO) comprising 3,305,525 keyframes. The same 24 query videos contained in CC_WEB_VIDEO are used for benchmarking. Unfortunately, only the HSV and LBP histograms of the video keyframes are provided by the authors. Similar to the CC_WEB_VIDEO, many outdated approaches report competitive performance, indicating that the dataset is

not challenging enough.

A recently published dataset is the SVD [50]. This dataset is tailored to cover the need of NDVR of short videos in particular. It consists of 562,013 short videos crawled from a large video-sharing website, namely Douyin[1]. The average length of the collected videos is 17.33 seconds. The videos with more than 30,000 likes were selected to serve as queries. Candidate videos were selected and annotated based on a three-step retrieval process. A large number of probably negative unlabelled videos were also included to serve as distractors. Hence, the final dataset consists of 1,206 queries with 34,020 labelled video pairs and 526,787 unlabelled videos. The queries are split into two sets, i.e., training and test set with 1,000 and 206 queries, respectively. However, the dataset consists of solely short videos that usually are single-shots, which does not generalize to the retrieval of long untrimmed cases. Also, it includes annotations only for the near-duplicate video pairs.

Another popular public dataset is the MUSCLE-VCD, created by Law-To et al. [69]. This dataset was created for the problem of video copy detection. It consists of 100 hours of videos, including Web video clips, TV archives, and movies of different bitrates, resolutions and video formats. A set of original videos and their corresponding transformed queries are given for evaluation. Two types of transformation are applied on the queries: a) ST1: copy of the entire video with a single transformation, where the videos may be slightly recoded and/or subjected to noise addition; b) ST2: partial copy of videos, where two videos share one or more video segments. Both transformations were artificially applied using video-editing software. The transformed videos or segments were used as queries to search their original versions in the dataset. Due to its small size and the limited number of transformations applied to the original videos, this dataset does not serve the needs of large-scale and more general problems such as FIVR.

[1] `http://www.douyin.com`

The annual TRECVID [1] evaluation included a task on Content-Based Copy Detection (CBCD) in years 2008 to 2011. Each year a benchmark dataset was generated and released only to the registered participants of the task. The TRECVID-CBCD datasets were constructed following the same process as the MUSCLE-VCD dataset. The latest edition of the dataset [67] contains 11,503 reference videos of over 420 hours and 11,256 queries. Query videos are categorized into three types: a reference video only, a reference video embedded into a non-reference video, and a non-reference video only. Only the first two types of query videos are copies of videos in the dataset. The queries were automatically generated by randomly extracting a segment from a dataset video and imposing a few predefined transformations. The contestants were asked to find the original videos and detect the copied segment. However, the TRECVID-CBCD task has not been organized since 2011 due to the near-perfect performance of the submitted methods. This is misleading since the performance of the same methods on other user-generated datasets that have been developed for similar problems is far from satisfactory. This fact reveals that the developed dataset can not simulate real-world scenarios where an arbitrary number of transformations might have been applied to the original videos by users.

A more recent dataset that is relevant to our problem is VCDB [51]. It is composed of videos from popular video platforms (YouTube and Metacafe) and has been compiled and annotated as a benchmark for the partial copy detection problem. VCDB contains two subsets, the *core* and *distractor*. The core subset contains 28 discrete sets of videos composed of 528 videos with over 9,000 pairs of partial copies. Each video set was manually annotated by seven annotators, and the video chunks of the video copies were extracted. The distractor subset is a corpus of approximately 100,000 distractor videos, which is used to make the video copy detection problem more challenging. In total, VCDB contains 100,528 videos amounting to more than 2,000 hours of video. Its main limitation is that only a limited number of its videos have been annotated (528 videos in the core dataset), so it can not cover the need for large-scale video retrieval.

Another relevant dataset is the EVVE dataset [99], which was developed for the problem of event video retrieval. The main task of this dataset is the retrieval of all videos that capture the event depicted by a query video. The dataset contains 13 major events that were provided as queries to YouTube. A total of 2,995 videos were collected, and 620 of them were selected as queries. Each event was annotated by one annotator, who first produced a precise definition of the event. In addition to the videos collected for the specific events, the authors also retrieved a set of 100,000 distractor videos by querying YouTube with unrelated terms. These videos were all collected before a certain date, which ensures that the distractor set does not contain relevant events since all EVVE events occurred after that date. Nevertheless, the definition of the related videos is much broader than the one we consider in FIVR, and additionally, the dataset contains annotations only for videos from the same event and not for its near-duplicates.

Finally, there are several relevant video datasets that have been used for content-based videos retrieval, i.e., Youtube-8M [4], YFCC100M [120], ActivityNet [17], FCVID [53]. Such datasets contain information regarding a set of classes related to the concepts or actions depicted in the videos. Since they represent the most general video retrieval scenario and contain only class-level annotation, they are not suitable for our purpose as they do not cover our requirements for FIVR, and thus, they can not be exploited.

In conclusion, all of the datasets mentioned above have several limitations. The most important limitations are: i) Many datasets are saturated and do not pose a challenge as old and outdated methods achieve near-perfect results (i.e., CC_WEB_VIDEO, UQ_VIDEO, MUSCLE-VCD, TRECVID-CBCD). ii) Others are relatively small in size, so they can not simulate large-scale retrieval scenarios (i.e., CC_WEB_VIDEO, MUSCLE-VCD, TRECVID-CBCD, VCDB). iii) Some datasets either simulate limited aspects of the problem (i.e., SVD, MUSCLE-VCD, TRECVID-CBCD) or con-

sider only very broad definitions to determine the related videos (i.e., EVVE). Finally, iii) no dataset contains proper annotations that cover the case of the same incident videos. In short, none of the above datasets can satisfy the requirements posed by the FIVR problem. For that reason, we built a new large-scale video dataset (FIVR-200K) according to the FIVR definitions. The dataset consists of videos depicting a variety of real-world news events, challenging cases of positive video pairs, and a large number of distractor videos.

2.4 Conclusion, limitations and novelty

In this chapter, we presented the most important works in the video retrieval literature related to FIVR. We began by discussing the most relevant retrieval tasks, along with the existing definitions. We continued with the description of several state-of-the-art works classified based on the level of similarity employed during computation. We followed with the presentation of the video datasets that have been composed to simulate relevant retrieval tasks.

We have drawn several conclusions from our literature review. First, the existing definitions can range from very narrow, i.e., only video copies or near-duplicates are considered, to very broad, where videos depicting the same semantic concept are considered related. Also, dealing with the most general problem does not necessarily address the more fine-grained ones. Regarding the retrieval methods, there are three main categories of methods classified based on the level of the calculated similarity, i.e., video-level, frame-level, and filter-and-refine. For all of the categories, the most prominent solutions that achieve state-of-the-art results employ deep learning [28, 148, 74, 3, 75], and especially in deep metric learning settings [115, 11, 70, 141, 104]. Video-level methods are employed mainly in the general aspect of the retrieval problem, i.e., for CBVR [115, 73, 70, 141]. However, frame-level methods are used to tackle more fine-grained problems, i.e., VCD [52, 148, 3], NDVR [22, 75, 104], FIVR

[74, 104], or EVR [94, 11, 104]. Finally, for the simulation of the video retrieval problem, large-scale datasets with user-generated videos datasets are required. There are various datasets covering all the related retrieval problems, but none of them can be employed to simulate FIVR.

To this end, in this thesis, we formulate the FIVR problem, and we build retrieval approaches that tackle the limitations and go beyond the state-of-the-art, as described below:

Firstly, since there is no proper definition that covers our problem, we introduce the FIVR problem with the composition of a large-scale video dataset that has been crafted and annotated to serve as an evaluation testbed for the benchmarked approaches (Chapter 3). We propose formal definitions for the association types considered in the dataset, i.e., near-duplicate videos, videos captured from different viewpoints, and the same incident videos. The definitions proposed in the field of NDVR ([134] and [106]) had a significant influence on the formulation of the first two. In the case of the same incident videos, we considered the EVR definition proposed in [99] for the proper separation of the two associations. Our main novelty is the definitions of the video associations for FIVR and the composition of a large-scale dataset that simulates the problem.

Secondly, at the time of publication of our early work, only limited methods in the literature of video retrieval employed solutions based on deep learning. Hence, we present two video-level methods that leverage deep learning (Chapter 4). Our first method is an unsupervised approach based on a BoW scheme [111]. Our novelty is the utilization of deep learning features from the intermediate convolutional layers [124, 146], and a layer aggregation scheme for the generation of global video representations. Our second method is a supervised approach that employs Deep Metric Learning (DML) to train a network to approximate an embedding function that maps videos to a feature space where the related videos are closer than the irrelevant ones. Our

main novelty is the problem formulation for the adaptation of the DML pipeline to the domain of video retrieval.

Finally, there is a lack of methods in the literature that exploits both spatial and temporal structure of video similarity. State-of-the-art methods either focus on the spatial processing of frames (disregarding temporal information) or build temporal alignment schemes based on global frame representations (discarding spatial information), which are usually not capable of capturing a large variety of temporal similarity patterns due to their rigid aggregation approach. To this end, to tackle these limitations, we propose a video similarity learning architecture that considers fine-grained spatio-temporal information during the similarity computation (Chapter 5). We devise a novel frame-to-frame similarity computation scheme that captures the intra-frame relations between frames, and we train a CNN network for video-to-video similarity calculation, which captures the inter-frame relations. Our main novelty is the composition of a fine-grained spatio-temporal model for video similarity learning.

Chapter 3

Fine-grained Incident Video Retrieval: new problem and dataset

Contents

In this chapter, we formulate the retrieval task of Fine-grained Incident Video Retrieval (FIVR). Our objective in FIVR is to retrieve all videos that depict the same incident given a query video – related video retrieval tasks adopt either very narrow scopes, considering only near-duplicate videos, or very broad, considering videos from the same event. To formulate the case of same incident videos, we definite three video associations, i.e., duplicate, complementary, and incident scene videos, taking into account the spatio-temporal spans captured by video pairs. We construct and present a large-scale annotated video dataset to address the benchmarking needs of all such tasks, which we call FIVR-200K, and it comprises 225,960 videos. To create the dataset, we devise a process for the collection of YouTube videos based on major news events from recent years crawled from Wikipedia and deploy a retrieval pipeline for

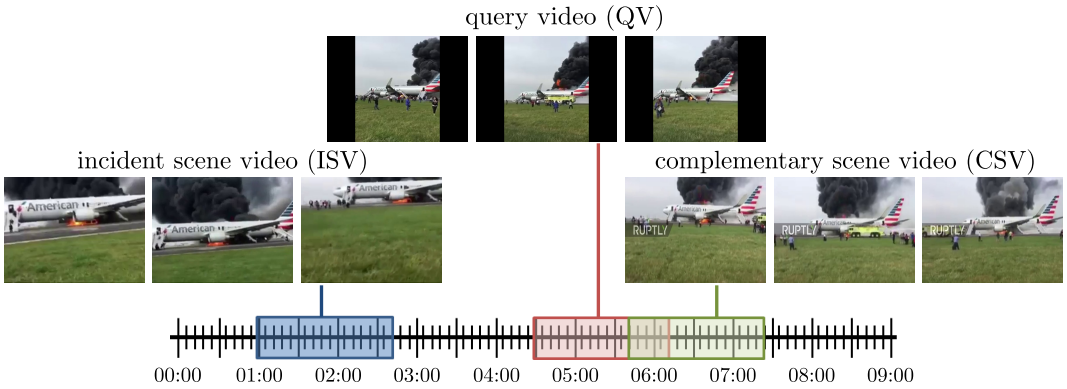


Figure 3.1: Examples of a query video (QV) with one *complementary scene video* (CSV) and one *incident scene video* (ISV) on the timeline of an incident. The following colour coding is used: i) red for QV, ii) green for CSV, and ii) blue for ISV.

the automatic selection of query videos based on their estimated suitability as benchmarks. We also devise a protocol for the annotation of the dataset with respect to the four types of video associations defined in FIVR, which facilitates the evaluation of the benchmarked methods in different retrieval settings using the same dataset – in contrast to other related datasets that can simulate only a single retrieval scenario. We also conduct a comprehensive experimental study comparing state-of-the-art approaches with handcrafted and deep features, highlighting the challenging aspect of the collected dataset and the difficulty of the FIVR problem.

More precisely, we address two fundamental associations between similar videos: a) duplicate videos and b) videos of the same incident. By duplicate videos, we refer to videos that have been captured by the same camera and depict exactly the same scene, but may have undergone some visual transformations (e.g., brightness/contrast, colour, recompression, noise addition, cropping). The second type of similar videos that we consider are videos capturing the same incident. This category may be split into subcategories: a) videos that depict the same incident scene from complementary viewpoints, and b) videos that capture the same incident at different time intervals. In particular, two videos in the first category must have at least one video segment where there is temporal overlap between the depicted incident. Videos in the second

subcategory need to depict the same incident but do not need to have temporal overlap. Figure 3.1 illustrates three example videos that capture the same incident along with their FIVR associations.

The goal of this chapter is to propose and formulate the Fine-grained Incident Video Retrieval (FIVR) problem through the composition of a challenging dataset that will serve the benchmarking needs for different variants of the problem. The main contributions of this work can be summarized in the following:

- The introduction of the Fine-grained Incident Video Retrieval (FIVR) problem and the definition of different associations between pairs of videos.
- The creation and availability of a large-scale dataset (FIVR-200K)[1] consisting of 225,960 videos, derived from a wide variety of real-wold news events, which leads to challenging retrieval cases.
- The development of a process for the collection and annotation of videos based on major news events crawled from Wikipedia and a principled process for the automatic selection of suitable video queries.
- A comprehensive experimental study comparing five state-of-the-art approaches implemented with several visual descriptors (handcrafted and deep features).

The chapter is organized as follows: Section 3.1 introduces the necessary notation and definitions that formulate the FIVR problem. Section 3.2 describes the dataset construction process, including the video collection, query selection, and video annotation. Section 3.3 reports on the results of the experimental study on the dataset. Section 3.4 concludes the chapter.

Table 3.1: Background notation and definitions.

Term	Description
$\mathbf{x}$	an arbitrary video
x_i	i^{th} scene of $\mathbf{x}$
z_i^x	spatio-temporal span of the i^{th} scene of $\mathbf{x}$
v_i^x	viewpoint of the i^{th} scene of $\mathbf{x}$
h_i^x	incident captured in the i^{th} scene of $\mathbf{x}$
$\mathbf{z}^x$	spatio-temporal span of the entire video $\mathbf{x}$
$\mathbf{v}^x$	viewpoints of the entire video $\mathbf{x}$
$\mathbf{h}^x$	incidents captured in the entire video $\mathbf{x}$
S	space of scenes
Z	space of spatio-temporal span
V	space of viewpoint
H	space of incidents
f	function that maps an incident to a unique spatio-temporal span
g	function that, given a viewpoint, maps a spatio-temporal span to a scene

3.1 Problem definition

We consider that a real-world incident determines a unique spatio-temporal span, i.e., there is a function $f : H \rightarrow Z$ that maps the incidents from an incident space H to a continuous spatio-temporal space Z, which can be understood as the specific place and time interval that an incident takes place. Furthermore, a video can be perceived as the mapping of the real world to a sequence of two-dimensional raster images with three colour channels. Additionally, as defined in the field of temporal video segmentation [33], a video can be decomposed in a sequence of *scenes*, which are temporal segments that cover either a single event or several related events taking place in parallel. Thus, an arbitrary video $\mathbf{x}$ with a sequence of n non-overlapping scenes may be denoted as $\mathbf{x} = [x_1\ x_2\ ...\ x_n]$, where $x_i \in S$ and S is the space of scenes. We may also consider a function $g : Z, V \rightarrow S$ that maps a real-world spatio-temporal span from space Z and given a specific viewpoint from space V, where V is the viewpoint space, to a video scene. Note that knowing functions f and g is not our objective; instead, they are

[1] `http://ndd.iti.gr/fivr/`, `https://github.com/MKLab-ITI/FIVR-200K`

Table 3.2: Definitions of the different types of associations between video pairs.

Duplicate Scene Videos (DSV)	Videos that share at least one scene (captured by the same camera) regardless of any applied transformation.	*Definition 1*: Given a query video $\mathbf{q}$ with a number of n scenes $\mathbf{q} = [q_1\ q_2\ ...\ q_n]$, spatio-temporal span $\mathbf{z}^q$ and viewpoints $\mathbf{v}^q$, and a candidate video p with a number of m scenes $\mathbf{p} = [p_1\ p_2\ ...\ p_m]$, spatio-temporal span $\mathbf{z}^p$ and viewpoints $\mathbf{v}^p$, there is a binary function $\mathrm{DS}(\cdot,\cdot)$ that indicates whether the two videos are DSVs $$\mathrm{DS}(q,p) = \begin{cases} 1 & \exists i \in [1,m]\ (z_i^p \subseteq \mathbf{z}^q \wedge v_i^p \in \mathbf{v}^q) \\ 0 & \text{otherwise} \end{cases} \quad (3.1)$$
Complementary Scene Videos (CSV)	Videos that contain part of the same spatio-temporal segment, but captured from different viewpoints.	*Definition 2*: Given a query video $\mathbf{q}$ with a number of n scenes $\mathbf{q} = [q_1\ q_2\ ...\ q_n]$, spatio-temporal span $\mathbf{z}^q$ and viewpoints $\mathbf{v}^q$, and a candidate video p with a number of m scenes $\mathbf{p} = [p_1\ p_2\ ...\ p_m]$, spatio-temporal span $\mathbf{z}^p$ and viewpoints $\mathbf{v}^p$, there is a binary function $\mathrm{CS}(\cdot,\cdot)$ that indicates whether the two videos are CSVs. $$\mathrm{CS}(q,p) = \begin{cases} 1 & \exists i \in [1,m]\ (z_i^p \subseteq \mathbf{z}^q \wedge v_i^p \notin \mathbf{v}^q) \\ 0 & \text{otherwise} \end{cases} \quad (3.2)$$
Incident Scene Videos (ISV)	Videos that capture the same incident, i.e. they are spatially and temporally close, but have no overlap.	*Definition 3*: Given a query video $\mathbf{q}$ with a number of n scenes $\mathbf{q} = [q_1\ q_2\ ...\ q_n]$, spatio-temporal span $\mathbf{z}^q$ and incidents $\mathbf{h}^q$, and a candidate video p with a number of m scenes $\mathbf{p} = [p_1\ p_2\ ...\ p_m]$, spatio-temporal span $\mathbf{z}^p$ and incidents $\mathbf{h}^p$, there is a binary function $\mathrm{IS}(\cdot,\cdot)$ that indicates whether the two videos are ISVs. $$\mathrm{IS}(q,p) = \begin{cases} 1 & \exists i \in [1,m]\ h_i^p \in \mathbf{h}^q \wedge \nexists j \in [1,n]\ z_j^p \subseteq \mathbf{z}^q \\ 0 & \text{otherwise} \end{cases} \quad (3.3)$$

solely used for the proper formulation of our problem.

For the accurate definition of the associations between videos, we consider that each scene x_i of an arbitrary video $\mathbf{x}$ has the corresponding attributes: the captured spatio-temporal span $z_i^x \in Z$, the viewpoint $v_i^x \in V$ of the camera and the incident $h_i^x \in H$ that corresponds to the captured spatio-temporal span. By aggregating all attributes of the scenes of video $\mathbf{x}$, we can derive the attributes for the entire video: the entire captured spatio-temporal span $\mathbf{z}^x \in Z$, all viewpoints $\mathbf{v}^x \in V$ of the video scenes and the different incidents $\mathbf{h}^x \in H$ occurring during the captured spatio-temporal span.

To properly define the relations between videos, we consider three fundamental types

of association between videos, summarized in Table 3.2. These are defined based on the relation between the viewpoints and spatio-temporal spans of the compared videos.

We denote as **Duplicate Scene Videos** (DSVs), two videos that share at least one scene (as captured by the same camera) regardless of any applied transformation. The shared scenes must be close to exact duplicates of each other but can be different in terms of photometric variations, editing operations, length, and other modifications. More precisely, they have to originate from the same spatio-temporal span and viewpoint. Videos that contain semantically similar scenes are not considered DSVs. Definition 1 provides a formal definition of the DSVs. A special case of Definition 1 is when Equation 3.1 is valid for all scenes of the candidate video. Such cases are denoted as **Near-Duplicate Videos** (NDVs).

Videos in the second category have to share at least one common segment of the same incident. These are denoted as **Complementary Scene Videos** (CSVs). The term complementary is referred to the different viewpoint, i.e., different angle, of two videos that captures the same incident from different devices at the same time. In particular, each of the two videos of a CSV pair needs to contain a spatio-temporal segment that is temporally overlapping with the spatio-temporal segment of the other. However, to be included in this category, the two video segments need to be captured from different cameras, and hence, offer complementary viewpoints of the incident. Since the identification of temporal overlap is a challenging task, any audio or visual cue may be taken into consideration to make such an inference. The formal definition of CSVs is provided in Definition 2.

Videos in the third category depict the same incident but have no temporal overlap. These are referred to as **Incident Scene Videos** (ISVs), and they are formalized in Definition 3. Such videos still need to be spatially and temporally related, i.e., their spatio-temporal span should originate from the same incident. Additionally, if the query depicts a particular incident in a long event or a sequence of incidents,

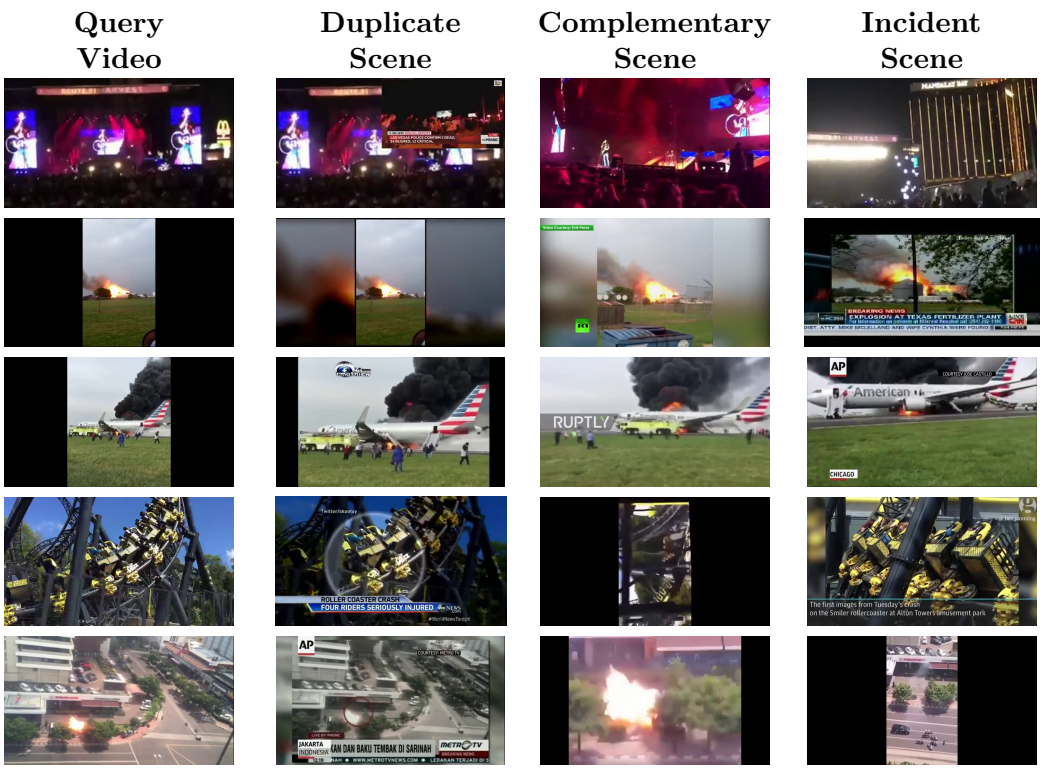


Figure 3.2: Examples of queries and retrieved associated videos from FIVR-200K.

then only the videos that capture the particular incident are included in this category. Additionally, the inference that two videos originate from the same incident may derive from video metadata (e.g., title, description) or audio, i.e., it is not necessary to associate the two videos with the event solely on the basis of their visual content. There are some rare cases where ISVs have no obvious visual cues linking them to each other, and no such inference can be made without outside knowledge. An example is a case where the query captures an incident from the outside of a building, and there are ISVs from the inside of the same building captured during the same incident.

Figure 3.1 illustrates selected frames of a query video and one candidate video from each category (CSV, ISV). The video fragments have been coloured accordingly, with red indicating the query video, green the CSV, and blue the ISV. Also, a sample timeline is presented to illustrate the time span where each type of video occurs. The example video depicts the fire in the American Airlines flight 383 at Chicago O'Hare

airport[2]. There are a number of videos in FIVR-200K from that incident, capturing various viewpoints and different time spans. The query video depicts the passengers standing outside the plane and the firefighters trying to put out the fire. The CSV is captured from a slightly different viewpoint. The overlap between the two videos can be determined by the movement of the firefighter truck passing in front of the plane and the position of the people. The ISV is in a distinct time span relative to the query. It is captured before the query video, at the moment when the passengers exit the plane through the emergency exits. Figure 3.2 illustrates some additional examples of FIVR associations.

3.2 Dataset generation process

In this section, we describe the pipeline that we developed for the composition of the FIVR-200K dataset. As explained in Section 1.1.1, the particular use-case that we are interested in is the retrieval of breaking news videos. We first present the process for the collection of the dataset (Section 3.2.1). Then, we explain the principled process for the selection of the query videos based on their suitability (Section 3.2.2). We also report the protocol that we followed for the annotation of the dataset (Section 3.2.3). Finally, we provide some basic statistics for the composed dataset (3.2.5).

3.2.1 Video Collection

The FIVR-200K dataset was designed with the following goals in mind: a) the videos should be associated with a large number of news events, b) the categories of these news events should be the same, and c) the dataset size needs to be sufficiently large to make retrieval of relevant results challenging.

Based on the above requirements, we set up the process depicted in Figure 3.3 to retrieve videos about major news events that took place during recent years. First,

[2] https://en.wikipedia.org/wiki/American_Airlines_Flight_383_(2016)

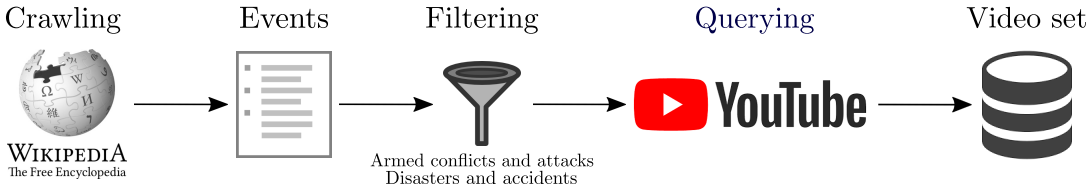


Figure 3.3: Overview of the video collection process.

Table 3.3: Examples of crawled news events.

Headline	**Date**	**Category**	**Text**	**Source**
Syrian civil war	2013-01-01	Conflicts	Fierce clashes erupt ...	BBC
Greek debt crisis	2015-07-07	Business	Eurozone leaders hold ...	Reuters
Hurricane Harvey	2017-08-29	Disasters	The death toll from ...	NY Times
US elections	2016-11-08	Politics	Voters in the United ...	ABC
Artificial intelligence	2016-01-27	Science	A computer program ...	MIT Rev.
Boston Mar. Bombing	2014-07-21	Law & Crime	Azamat Tazhayakov ...	MSN News
2016 Sum. Olympics	2016-08-12	Sports	Singaporean swimmer ...	NY Times

we crawled Wikipedia's 'Current Event' page[3] to build a collection of the major news events since the beginning of 2013. Each news event is associated with a topic, headline, text, date, and hyperlinks. Five examples of collected news events are displayed in Table 3.3. For the remaining steps of the process, we retained only news events categorized as 'Armed conflicts and attacks' or 'Disasters and accidents'. We selected these two categories to find multiple videos on YouTube that report on the same news event, and ultimately to collect numerous pairs of videos that are associated with each other through the relations of interest (DSV, CSV, and ISV). The time interval used for crawling the news events was from January 1st 2013 to December 31st 2017. A total of 9,431 news events were collected, and 4,687 news events were retained after filtering.

In the next step, the public YouTube API[4] was used to collect videos by providing event headlines as queries. The results were filtered to contain only videos published at the corresponding event start date and up to one week after the event. Furthermore, they were filtered to contain only videos with a duration of up to five minutes, which resulted in the collection of 225,960 videos (∼48 videos/event). At this point, it is

[3] `https://en.wikipedia.org/wiki/Portal:Current_events`
[4] `https://developers.google.com/youtube/`

worth noting that several of the news event headlines in Wikipedia describe long-running news events (e.g., Syrian civil war). However, we are interested in collecting specific/particular news events within longer-term ones. Yet, this is not an issue for our data collection process since the combination of the general event headline with the date of the particular news event is often sufficient to retrieve a variety of videos that depict the incidents of interest that are alluded by the respective Wikipedia entries.

3.2.2 Query Selection

Selecting "appropriate" queries is important for ensuring that the dataset annotations and evaluation protocol are representative of the challenges arising in real-world search tasks. To this end, the query selection process was designed with two goals in mind: a) to generate challenging queries, i.e., queries that will lead to many distractor videos and challenge content-based retrieval systems, and b) to find queries that will lead to the retrieval of videos with various modifications that will be not only trivial NDV cases but also contain interesting variations (e.g., different viewpoints of the same scene), i.e., CSV and ISV. To achieve those two goals, we implemented a largely automatic process that is described below.

First, the visual similarity between videos was computed as the cosine similarity between the *term frequency-inverse document frequency* (tf-idf) representations derived from visual words extracted from their visual content. The visual words are extracted based on the NDVR method described in [66] and aggregated based on a Bag-of-Word (BoW) scheme. We sample one frame per second and extract the embedding vectors using a trained Deep Metric Learning (DML) network, which are then mapped and aggregated to the three closest visual words from a codebook of size 10k. The DML network was trained on the VCDB dataset [51], and the visual codebook was built by sampling one frame per video in the dataset and extracting the corresponding embedding vector. Next, the textual similarity between videos was computed as the cosine similarity between the tf-idf representations of their titles. To perform

the similarity calculation, we first pre-processed video titles with the NLTK toolkit [15], applying part-of-speech (PoS) tagging, removing all verbs (which we found to introduce unnecessary noise) and providing the results to the NLTK WordNet-based lemmatizer to extract the lemmas, which constitute the word-based representation of the titles. The overall video similarity derives from the average of the visual and textual similarity. We expect that the visual similarity will opt for DSV cases, while textual for CSV and ISV. BoW representation was selected for both visual and text words because of its sparsity, which was practical for fast similarity calculation and efficient dataset annotation.

In the next step, we computed all non-zero similarities between video pairs. Only video pairs that share at least one visual or text word were considered, which resulted in complexity much lower than $O(n^2)$. Afterwards, we created a video graph G by connecting with an edge video pairs with similarity greater than a certain threshold t_s (empirically set to 0.7). To identify meaningful video groups, we extracted the connected components C of the video graph G with more than two videos. Then, we defined the uploader ratio r_c of each component $c \in C$ using Equation 3.4.

$$r_c = \frac{|\{u_v | v \in c, u_v \in U\}|}{N_c} \tag{3.4}$$

where the numerator is the number of unique uploaders in the component, v is a video in the component, u_v is the uploader of video v, U is the set of uploaders in the dataset, and N_c is the number of videos in the component. We empirically found that components with a low uploader ratio usually contain videos from a single specific channel (e.g., news channel) with titles that are very similar (e.g., exactly the same title with a different date) or with content that is visually highly similar (e.g., the same presenter reporting news in the same background). However, based on our definition, such videos are neither considered DSV nor CSV or ISV. For that reason, we discard components with an uploader ratio of less than a threshold t_r (empirically set to 0.75). Additionally, since we need components consisting of videos that refer

to the same incident, we applied another criterion on the component set based on the publication date of their videos and retained only components consisting of videos that were published within a time window of two weeks.

Our goal was to find queries that lead to result sets with many DSV, CSV, and ISV. Intuitively, large components with many (visually and textually) similar videos have a better chance of containing such videos. For that reason, we ranked connected components based on their size and selected one query video per component. Keep in mind that the components have been formed based on visual-textual similarity, i.e., the visual similarity will derive DSV cases, while textual for CSV and ISV. Also, we considered that short videos with few shots were the most suitable candidates for having been modified and reposted several times (both as single videos or as part of mash-ups). Therefore, we selected videos with a duration of less than a threshold t_d (empirically set to 90 seconds). Attempting to find the original version of videos in each cluster, we chose the video that was published earliest as the query video.

The total number of queries using the above process was 635. Since it would be overly time-consuming to annotate all of them, we selected the top 100 as the final query set (ranked based on the size of the corresponding graph component).

3.2.3 Annotation Process

Figure 3.4 depicts the annotation process for one query, carried out in three stages. Given the query, two video groups are retrieved, one based on visual similarity and one based on textual similarity. All videos are annotated based on their relation to the query according to our definitions. In the first stage, we annotate the videos in the "visual" group, ranked based on their visual similarity to the query. The end of the first stage occurs when a total number of 100 irrelevant videos have been annotated after the last relevant result (i.e., annotated as NDV, DSV, CSV, or ISV) or after the annotators have gone through the first 1000 videos (whichever of the two criteria applies first).

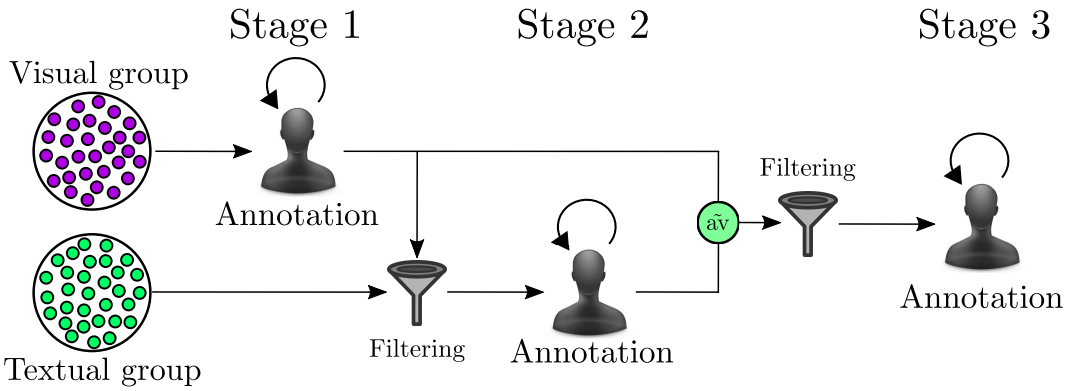


Figure 3.4: Overview of the annotation process. Two groups of videos are created based on their visual and textual similarity to the query. Three annotation phases take place, and two filtering steps are applied. $\tilde{av}$ stands for the average of visual and textual similarity between videos.

In the second stage, we remove all videos from the "textual" group that have already been annotated. The annotation process continues with the remaining videos in the textual group, ranked based on their textual similarity to the query. Similarly, this stage ends when a total number of 100 irrelevant videos have been annotated after the last relevant one or after the annotators have gone through 1000 videos. In the third and final stage, the remaining videos of the two groups are merged and filtered based on their publication date to minimize the possibility of having missed relevant videos. We retained those published within a time window of a week before and after the query's publication date. These were ranked based on the average visual-textual similarity, and the annotation proceeded until either 100 irrelevant videos were found after the last relevant video or no videos left in the merged group. The entire process is repeated for each one of the 100 selected queries.

The annotations are in video-level, i.e., we do not annotate the particular segments that the two videos are related. Also, he annotation labels used by the annotators, along with the corresponding definitions, are as follows:

- **Near-Duplicate (ND)**: These are a special case of DSVs, as specified in the Definition 1 in Table 3.2. Videos annotated with this label share all scenes (captured by the same camera) regardless of any applied transformation.
- **Duplicate Scene (DS)**: DSVs are annotated with this label based on Definition 1 in Table 3.2. Videos annotated with this label share at least one scene (captured by the same camera) regardless of any applied transformation.
- **Complementary Scene (CS)**: CSVs are annotated with this label based on Definition 2 in Table 3.2. Videos annotated with this label contain part of the same spatio-temporal segment but captured from different viewpoints.
- **Incident Scene (IS)**: ISVs are annotated with this label based on Definition 3 in Table 3.2. Videos annotated with this label capture the same incident, i.e., they are spatially and temporally close, but have no overlap.
- **Distractors (DI)**: Videos that do not fall in any of the above cases are annotated as distractors.

For the annotation of the dataset, the extracted queries were split into two parts, each assigned to two different annotators with expertise in multimedia-related fields. After the end of the annotation process, all annotated videos (excluding those labelled as DI) were revisited and tested for their consistency to the definitions by the author. For all 100 queries, the total number of unique videos annotated (including DIs) was approximately 140 thousand, i.e., the annotators went through approximately 1.4 thousand videos per query. Some videos were annotated multiple times because they had different labels for different queries. The entire annotation process needed approximately two months for its completion, with both annotators working full-time on this task.

3.2.4 Annotation Tool

To alleviate the annotation effort and facilitate annotators, we have developed an annotation tool that covers annotators' needs and equip them with several useful features and functionalities. The tool can be unveiled into two distinct modules: i) the back-end service and ii) the front-end user interface. The former is responsible for executing the video indexing and the retrieval of the provided query videos. The back-end service has been implemented with Spring [2] framework in Java, and Tensorflow [2] library in Python. The latter is responsible for the display of the results of the retrieval process and provides all the required options to the annotators to manage the results (e.g., submit/delete an annotation, delete video). The user interface has been implemented with the jQuery [56] library in JavaScript.

Figure 3.5 illustrates a screenshot of the annotation tool. Initially, the users provide the URL of the query video at the top of the screen and a similarity threshold to limit the results. After submitting the URL, the retrieved videos are displayed in separate windows, ranked based on their similarity to the query in descending order. Each window contains the actual video, which can be watched directly in the user interface from Youtube. Alongside the Youtube video, several useful information about the video is provided, i.e., the similarity to the query, the rank in the results, the title, the views, the publication date, the upload channel, the duration, and the category. A visual example of the result window is video #1 in the screenshot. The annotator can select the appropriate annotation for the retrieved video and submit it to the system. After submission, the corresponding label appears in the down left corner of the window under the Youtube video. In case that the annotator wants to delete its submission, there is a dedicated button under the label. Whether a video has been removed from Youtube, the annotators have to delete it from the database using the corresponding button on the right of the Youtube video. A handy feature of the user interface is the comparison button on the right side of the window. By pressing it, a

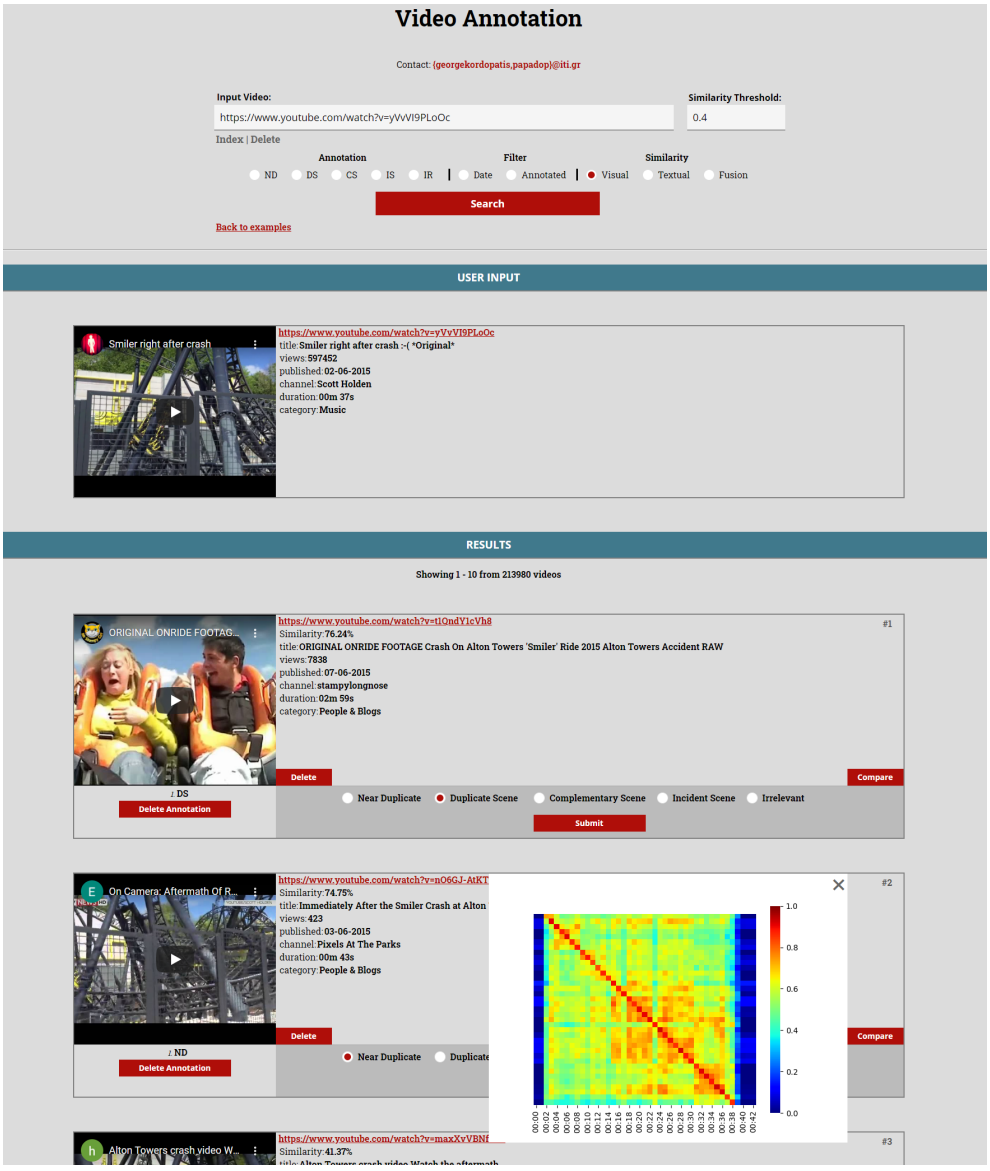


Figure 3.5: Screenshot of the annotation tool. From top to bottom, the following are displayed: the query field where the video URL is provided, several options for the search process, the query video with its information, and the retrieved videos with their information, started from video #1.

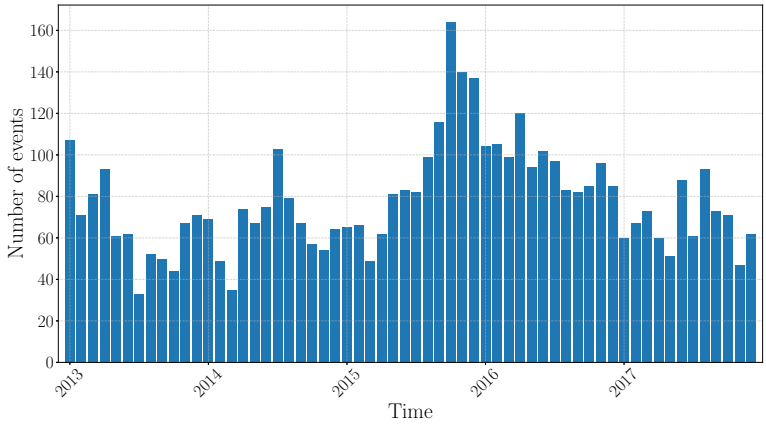
small window pops-up that illustrates the similarity matrix (i.e., contains the frame-to-frame similarity) between the query and the candidate video, as in video #2 in the screenshot. The similarity matrix is coloured appropriately so as the video segments with high similarity to easily distinguishable. Additionally, there are several options

to perform a query search. First, the similarity type (i.e., Visual, Textual, Fusion) used for retrieval can be selected. The Visual similarity is selected by default. Also, there are the Date and Annotated filters, whose functionality is as described in the previous section. Finally, the annotators can select a specific label to retrieve only the videos that have been annotated with that particular label.

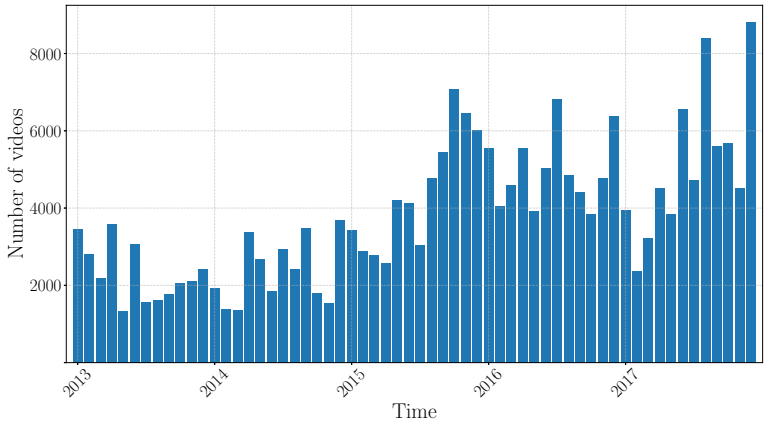
3.2.5 Dataset Statistics

In total, the dataset comprises 225,960 videos associated with 4,687 Wikipedia news events and 100 selected video queries. Figure 3.6 illustrates the monthly distribution of the collected news events, videos, and queries. There is a noteworthy peak of news events during the last quarter of 2015. During that period, major wars (e.g., the Syrian civil war, the war in Afghanistan, the Yemeni civil war), and a number of devastating natural disasters (e.g., hurricane Joaquin, Hindu Kush earthquake and an intense Pacific typhoon season) took place leading to daily newsworthy incidents. From the temporal video distribution, one may notice an increase in video sharing in the last two years, which does not correspond to the trend in the timeline of major news events. A possible explanation may be the increasing trend in video capturing and sharing on YouTube. Finally, it is noteworthy that the temporal distribution of queries approximately follows the one of videos over time with more query videos published during the last two years of the dataset. This confirms that the employed query selection process does not introduce temporal bias.

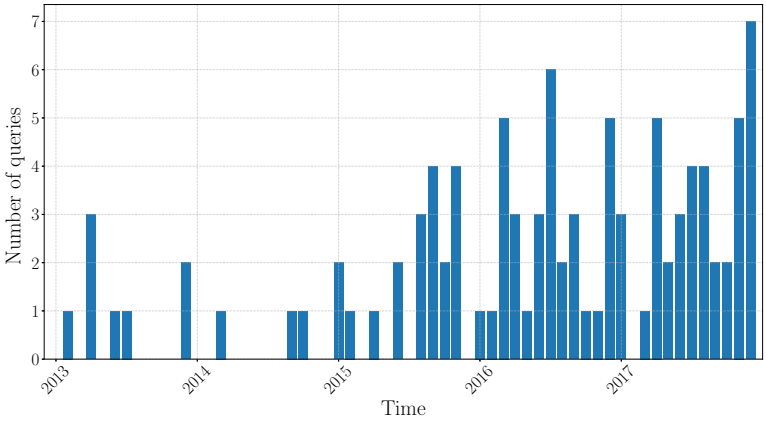
Table 3.4 presents the top news events based on their duration and number of collected videos. The duration of a news event is computed as the total number of days when it occurred in the collection. As expected, the longest news events, are wars or war-related events that usually last several years. The longest news event was the Syrian Civil War, which covered almost 500 days. However, news events with the most collected videos are breaking news events with large media coverage and live footage from multiple sources. The news event with the most collected videos was the



(a) news events



(b) videos



(c) queries

Figure 3.6: Monthly distribution of a) news events, b) videos and c) queries.

Table 3.4: (left) the top 10 longest news events (right) the top 10 news events with the most videos.

Long-running news events	days	**Breaking news events**	videos
Syrian civil war	499	November 2015 Paris attacks	651
War in Afghanistan	250	2017 Atlantic hurricane season	572
Iraqi insurgency	137	Charlottesville riots	569
War in NW Pakistan	118	Charlie Hebdo shooting	546
Iraqi civil war	116	2017 Las Vegas shooting	542
War in Somalia	101	Umpqua College shooting	486
Yemeni civil war	89	Assassination of Andrei Karlov	476
Israel-Palestine conflict	64	2016 central Italy earthquake	475
War in Donbass	62	2014 Peshawar school massacre	459
Libyan civil war	61	2017 Manchester arena bombing	457

terrorist attack in Paris, France, on 13 November 2015, where multiple suicide bombers struck, followed by several mass shootings. Figure 3.7 illustrates the distributions of video categories and duration. From the first, it is evident that the majority of collected videos are news-related, which was expected due to the nature of the searched events. Additionally, the 'People' category has a sizable portion of the collected videos. Regarding video duration, the majority of videos have a length between 30 to 120 seconds.

To further delve into the dataset content, we processed the video titles and extracted summary statistics. Initially, the language of the titles was detected using the detection approach by [108]. As expected, the predominant language was English, with 81.16%, followed by German with 2.58%. It is noteworthy that Indonesian ranked third with 1.74%, possibly due to several terrorist attacks that occurred in the region during the period of interest. Additionally, the most used nouns and locations are reported in Table 3.5. We extracted the nouns using the NLTK toolkit [15] and the mentioned countries using the method described in [61]. Unsurprisingly, the most used nouns were the ones related to wars and natural disasters, as well as the general words 'news' and 'video'. The most frequently mentioned countries were the ones related to long-lasting wars or major incidents with considerable media coverage.

In terms of the content source, the dataset contains videos from 66,919 unique

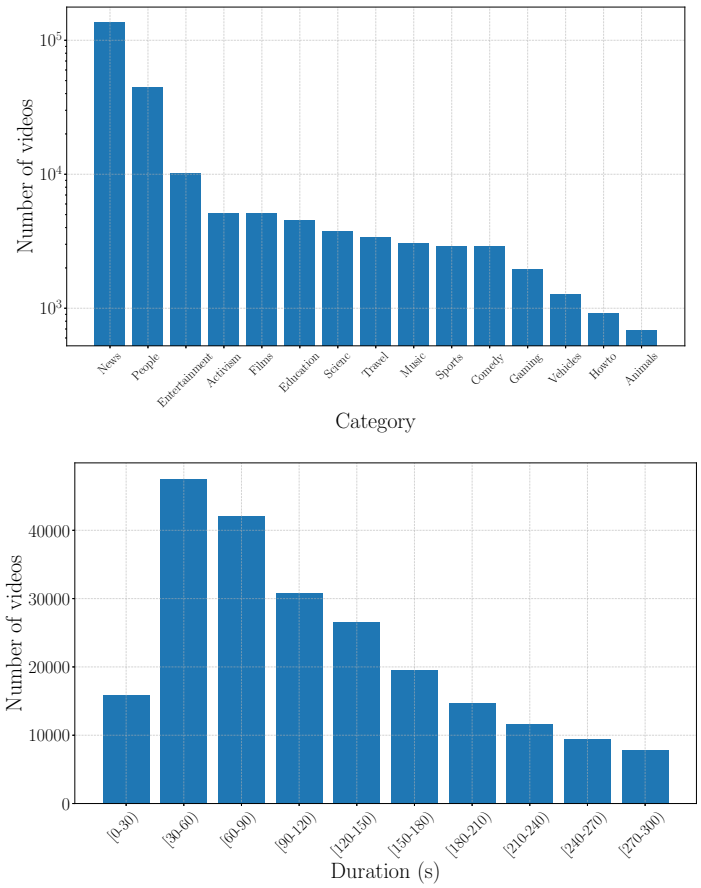


Figure 3.7: Distribution of videos based on their category and duration.

channels. As expected, the most prolific channels are news-related, including Wochit News, Ruptly, AP, and Al Jazeera, which regularly upload breaking-news content. Additionally, we grouped videos based on year of publication and found that the median of views per video remained approximately the same through the years.

Regarding the annotation labels, we found that the selected queries have, on average, 13 NDV, 57 DSV, 18 CSV, and 35 ISV. Figure 3.8 illustrates the distribution of annotation labels per query. Queries were ranked by the size of the cluster they were associated with (cf. Section 3.2.2). As expected, there was a considerable correlation (Pearson correlation=0.62) between cluster size and the number of videos annotated with one of the four relevant labels.

Table 3.5: (left) the top 10 most used nouns (right) the top 10 most refereed countries.

Nouns	videos
attack	18192
news	12133
earthquake	8016
fire	7121
hurricane	6447
crash	6304
video	5790
flood	5394
force	4702
army	4464

Locations	videos
Syria	13952
Ukraine	4545
Iraq	4545
Russia	3990
Yemen	3988
Turkey	3653
Israel	2776
Afghanistan	2691
China	2604
Egypt	2306

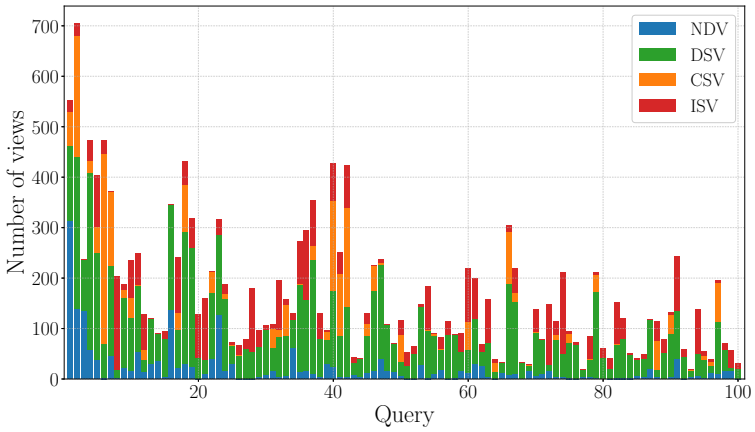


Figure 3.8: Distribution of annotation labels per query (best viewed in colour).

3.3 Comparative Study

3.3.1 Experimental Setup

In this section, we conduct a comparative study to evaluate the performance of several state-of-the-art video retrieval systems. We compare five state-of-the-art approaches based on different feature extraction, aggregation, and similarity calculation schemes. Additionally, three tasks are defined based on the labels that are considered relevant per task.

Evaluation Metrics

To evaluate retrieval performance, we build on the evaluation scheme described in [134]. We first employ the interpolated *Precision-Recall* (PR) curve. *Precision* is determined as the fraction of retrieved videos that are relevant to the query, while *Recall* is the fraction of the total relevant videos retrieved (Equation 3.5).

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN} \tag{3.5}$$

where TP, FP and FN are the true positives (correctly retrieved), false positives (incorrectly retrieved) and false negatives (missed matches), respectively. The interpolated PR-curve derives from averaging the Precision scores of all queries for given Recall ranges. The maximum Precision score is selected as the representative value for each Recall range. We further use *mean Average Precision* (mAP) as defined in [134] to evaluate the quality of video rankings. For each query, the *Average Precision* (AP) is calculated based on Equation 3.6.

$$AP = \frac{1}{n} \sum_{i=0}^{n} \frac{i}{r_i} \tag{3.6}$$

where n is the number of relevant videos to the query video, and r_i is the rank of the i-th retrieved relevant video. The mAP is computed by averaging the AP scores across all queries.

Benchmarked Approaches

Of the feature aggregation and similarity calculation techniques described in Section 2.2.1, we benchmark the following state-of-the-art approaches:

- **Global Vectors**: In the approach by [134], the HSV histograms are extracted for each video frame, and all frame descriptors are averaged to a single vector for the entire video. Video similarity is calculated based on the dot product between the respective vectors. This approach is denoted as GV.

- **Bag-of-Words**: We select two methods using this feature aggregation scheme. The first [18] is a traditional BoW approach that employs the ACC [41] features as frame descriptors. Every frame descriptor is mapped to a single visual word of a visual codebook. The second approach [65] is our variant of the traditional BoW scheme based on the intermediate CNN features. The feature vectors extracted from each convolutional layer are mapped to a word of a visual codebook (i.e., multiple codebooks are used, one codebook per layer). For both methods, the final video representation is the *tf-idf* representation of these visual words. Video ranking is performed based on the cosine similarity between the tf-idf video representations. The two methods are denoted as BoW and Layer BoW (LBoW), respectively.

- **Deep Metric Learning**: Our approach in [66] is selected as representative of this feature aggregation scheme. The intermediate CNN features [65] are extracted from the video frames and combined into global video descriptors, similar to GV. These descriptors are fed to a DML network to calculate video embeddings. Video similarity is calculated based on the Euclidean distance between these embeddings. This approach is denoted as DML.

- **Hashing Codes**: The approach by [114] is selected as representative of this feature aggregation scheme. Multiple frame features are extracted, i.e., HSV and LBP [145], and used to learn a group of hash functions that project frames into the Hamming space and combine them to a single video representation. Videos similarity is calculated based on Hamming distance. We use the public implementation provided by the authors. This approach is denoted as HC.

For all methods, we extract one frame per second to generate the frame descriptors. For the Bag-of-Words methods, the codebooks are created by sampling one frame per video in the dataset and extracting their visual descriptors. The selection of appropriate codebook size is important, so we experimented with 1K and 10K visual

words per codebook. Only the 10K codebook size results are presented since there is a large performance gap in favour of 10K words. For the DML and HC, the most important tuning parameter is the dimensionality of the output vectors. Yet, from our experiments, we concluded that it does not have a decisive impact on the performance of the approach. The DML is a supervised approach, so it is trained on the VCDB [51] dataset. The HC is an unsupervised approach, but a sample of 50K frame descriptors is still required to learn a set of hash functions. An extensive evaluation of the sensitivity to the parameters of the benchmarked methods is beyond the scope of this work; hence we selected those parameter values suggested by the authors or ones that gave better results in our initial experiments.

We should note that the LBoW and DML methods are part of our work in this thesis and are presented in further detail regarding their functionality and architecture choices in Chapter 4.

Visual Descriptors

For a more comprehensive and fair comparison, we also implemented the benchmarked approaches with the following visual descriptors.

- **Handcrafted Features**: We perform experiments with four widely used handcrafted features in the literature: HSV histograms, LBP [145], ACC [41], and VLAD-SURF [46].
- **Intermediate CNN Features**: We employ three popular architectures for the extraction of intermediate CNN features [65]: VGG-16 (VGG) [110], ResNet-152 (RES) [36] and Inception-V4 (INC) [117]. All of them are trained on ImageNet [24]. This feature extraction scheme is in detail presented in Section 4.1.
- **3D CNN Features**: We employ two popular architectures for the extraction of 3D CNN features: C3D [125] and I3D [19]. Both are trained on datasets

Table 3.6: Positive labels for each evaluation setup.

	Accepted Labels			
Task	**ND**	**DS**	**CS**	**IS**
DSVR	✓	✓		
CSVR	✓	✓	✓	
ISVR	✓	✓	✓	✓

annotated based on the display actions in the videos, i.e., UCF101 [116] and Kinetics 400 [19], respectively. To extract one visual descriptor per second, we feed the network with the corresponding number of frames suggested by the authors. We extract features with two techniques: (i) from the activations of the first fully connected layer after the convolutional layers, and (ii) from the intermediate 3D convolutional layers by applying MAC [124] pooling in the spatial (similar to the CNN features) and temporal axis.

The ResNet, Inception, and I3D architectures are very deep, which made the utilization of all convolutional layers impractical. Hence, we extracted features from the activations of the convolutions before max-pooling. For the HC method, we set up three runs based on i) handcrafted features, ii) CNN features extracted from the three architectures, and iii) 3D CNN features extracted from the two architectures.

Retrieval tasks

We evaluate three retrieval tasks. Table 3.6 indicates the positive labels per task.

- **Duplicate Scene Video Retrieval (DSVR)**: this task represents the NDVR problem, so it only accepts the videos annotated with ND or DS as relevant.
- **Complementary Scene Video Retrieval (CSVR)**: this scenario is a strict variation of the FIVR problem where only the ND, DS, and CS are accepted.
- **Incident Scene Video Retrieval (ISVR)**: this represents the general FIVR problem, and all labels (with the exception of DI) are considered relevant.

Table 3.7: mAP of the benchmarked approaches for the three retrieval tasks and the CC_WEB_VIDEO dataset.

Run	DSVR	CSVR	ISVR	CC_WEB
GV [134]	0.165	0.153	0.118	0.892
BoW [18]	0.240	0.220	0.171	0.944
LBoW [65]	**0.710**	**0.675**	**0.572**	**0.976**
DML [66]	0.398	0.378	0.309	0.971
HC [114]	0.265	0.247	0.193	0.958

3.3.2 Experiments

Benchmarked approaches

In this paragraph, we evaluate the performance of the five compared approaches. Table 3.7 illustrates the mAP of the benchmarked approaches on the three evaluation tasks of the FIVR-200K dataset and the CC_WEB_VIDEO dataset. LBoW outperforms all other approaches in all cases by a considerable margin. The second-best performance is achieved by DML, followed by HC and BoW. GV had the worst results in all cases. In particular, LBoW achieves a mAP score of 0.710 in the DSVR task, followed by DML and HC with 0.398 and 0.265, respectively. BoW and GV are the two worst-performing approaches with 0.240 and 0.165 mAP values, respectively. For the CSVR task, all approaches exhibit a drop in mAP, between 0.018 and 0.04. The performance is significantly worse in the ISVR task for all benchmarked approaches. The best method (LBoW) achieves a mAP score of 0.572, whereas the worst (GV) only 0.118.

The DSVR task of the proposed framework is closely related to the NDVR problem that is simulated by the CC_WEB_VIDEO dataset. The results make clear that the performances of all methods on FIVR-200K are significantly lower compared to CC_WEB_VIDEO, highlighting that the newly proposed dataset is much more challenging. All methods report very high mAP scores on CC_WEB_VIDEO, achieving values as high as 0.976. Even the GV approach achieves a score close to 0.9. The main reason for the performance gap is that the vast majority of positive video pairs in FIVR-200K are partially similar, not in their entirety but in particular segments.

Additionally, FIVR-200K contains a wide variety of user-generated videos about news events of similar nature, resulting in many challenging distractors.

Comprehensive experiments

Table 3.8 presents the mAP performance of all possible feature-aggregation combinations. To begin with the DSVR task, similar to the previous section, the LBoW aggregation scheme combined with the VGG CNN features achieves the best result (mAP=0.710) at a considerable margin from the second. Notably, VGG performs consistently better than the other two CNN architectures for all aggregation schemes. Additionally, LBoW clearly outperforms the regular BoW aggregation irrespective of CNN or 3D CNN architecture. The same conclusions apply in the case of 3D CNN features. The intermediate I3D features achieve the best results for all methods, with performance close to or better than the performance of VGG features. For example, in the case of DML, the I3D_{int} achieves 0.425 mAP, while VGG 0.398. Among the handcrafted features, VLAD-SURF provides the best results (mAP=0.323); however, the performance gap with deep features is considerable.

Similar conclusions apply in the case of the CSVR task, with the LBoW-VGG combination achieving the best results (mAP=0.675). The performance for all runs decreases slightly compared to the DSVR task, indicating that it presents a more challenging problem.

The performance is notably worse in the case of the ISVR task for every approach-feature combination, with the decrease ranging from 0.03 to 0.13 in mAP. This reveals that ISVR is a much more challenging problem, and new systems need to be devised to address it effectively. Overall, deep network features (either CNN or 3D CNN) outperform the handcrafted features by a significant margin. Moreover, DML boosts the performance of deep features compared to the GV runs. However, this is not the case for handcrafted features where the performance drops. Moreover, for 3D CNN archi-

Table 3.8: mAP of the benchmarked approaches and the different visual features for three retrieval tasks. N/A stands for Not Applicable and means that the aggregation scheme can not be applied to the corresponding feature descriptors.

DSVR					
Run	**GV**	**BoW**	**LBoW**	**DML**	**HC**
HSV	0.165	0.202	N/A	0.163	0.360
LBP	0.112	0.158	N/A	0.097	
ACC	0.196	0.240	N/A	0.182	
VLAD	0.294	0.323	N/A	0.285	
VGG	0.366	0.575	**0.710**	0.398	0.470
RES	0.350	0.523	0.596	0.374	
INC	0.333	0.500	0.608	0.367	
C3D$_{fc}$	0.244	0.341	N/A	0.266	0.434
C3D$_{int}$	0.355	0.541	0.658	0.387	
I3D$_{fc}$	0.321	0.464	N/A	0.336	
I3D$_{int}$	0.366	0.574	0.665	0.425	

CSVR					
Run	**GV**	**BoW**	**LBoW**	**DML**	**HC**
HSV	0.153	0.189	N/A	0.150	0.339
LBP	0.106	0.146	N/A	0.091	
ACC	0.183	0.220	N/A	0.169	
VLAD	0.275	0.311	N/A	0.265	
VGG	0.347	0.543	**0.675**	0.378	0.454
RES	0.333	0.499	0.572	0.358	
INC	0.313	0.473	0.571	0.348	
C3D$_{fc}$	0.231	0.314	N/A	0.252	0.415
C3D$_{int}$	0.336	0.502	0.628	0.374	
I3D$_{fc}$	0.312	0.444	N/A	0.325	
I3D$_{int}$	0.345	0.544	0.634	0.405	

ISVR					
Run	**GV**	**BoW**	**LBoW**	**DML**	**HC**
HSV	0.118	0.143	N/A	0.116	0.262
LBP	0.087	0.113	N/A	0.074	
ACC	0.142	0.171	N/A	0.128	
VLAD	0.214	0.236	N/A	0.206	
VGG	0.281	0.450	**0.572**	0.309	0.382
RES	0.274	0.414	0.488	0.296	
INC	0.257	0.406	0.488	0.290	
C3D$_{fc}$	0.176	0.242	N/A	0.194	0.334
C3D$_{int}$	0.261	0.398	0.510	0.295	
I3D$_{fc}$	0.253	0.364	N/A	0.265	
I3D$_{int}$	0.280	0.450	0.527	0.332	

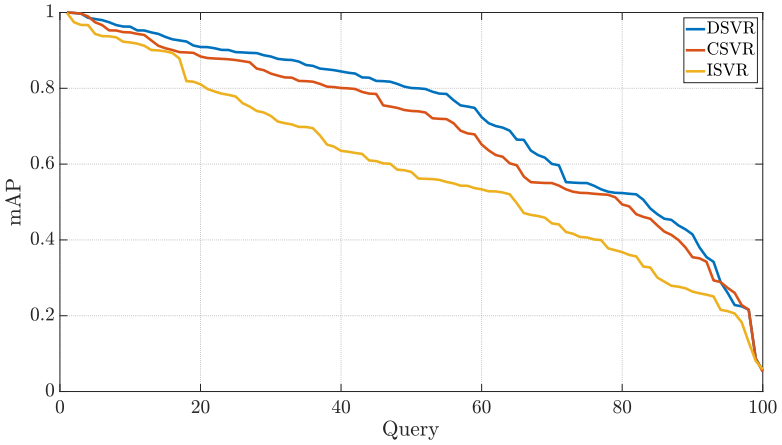
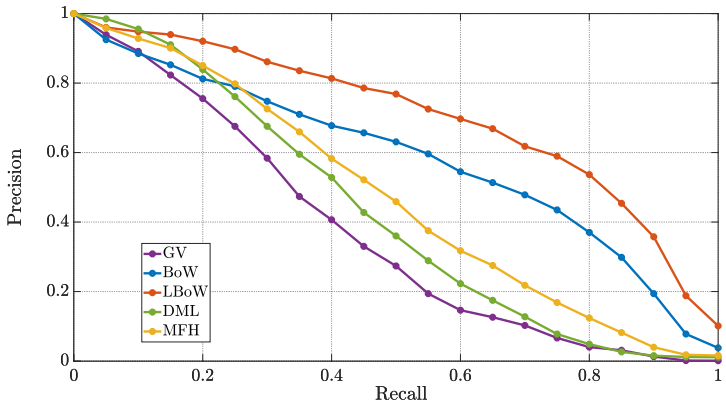


Figure 3.9: mAP of the queries in the dataset based on LBoW with VGG features run for the three retrieval tasks. The queries are ranked in descending order.

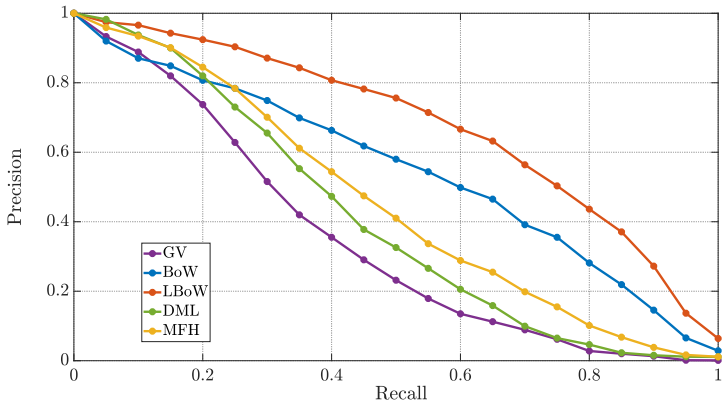
tectures, the runs with intermediate features consistently report better performance compared to the runs with features from the fully connected layers. HC combined with CNN features achieves the best performance compared to the other feature bundles, for all evaluation tasks. Additionally, GV performs poorly for all features compared to the other three schemes. For the rest of this chapter, we are going to refer to each method in relation to its combination with the best-performing features, i.e., VGG features for GV, BoW, and LBoW, I3D$_{int}$ features for DML, and the CNN features for HC.

Figure 3.9 illustrates the mAP per query of the best-performing run (LBoW with VGG features) for the three different tasks. The queries are ranked in descending order based on their mAP. For the DSVR task, 50% of the queries achieve higher than a 0.8 mAP, while the performance is significantly lower for the remaining queries. There is a notable drop in performance in the CSVR task, with 80% of the queries having higher than 0.5 mAP. Finally, it is evident that ISVR is a much harder task than the other two, with the majority of queries having lower than 0.6 mAP.

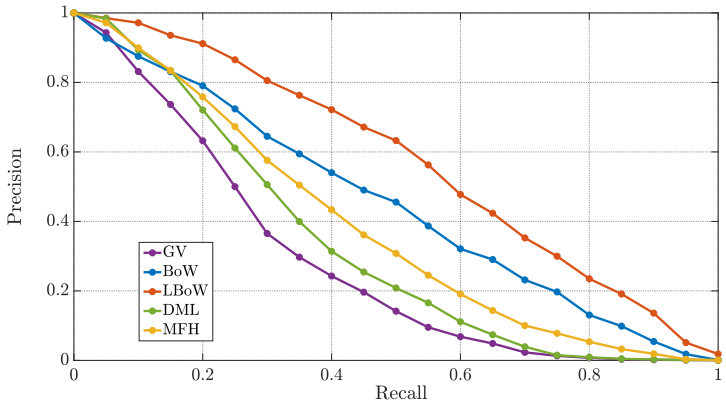
Figure 3.10 illustrates the interpolated PR-curves of the best-performing runs for



(a) DSVR



(b) CSVR



(c) ISVR

Figure 3.10: Interpolated PR-curves of the best-performing features for each approach in the three retrieval tasks.

Table 3.9: Storage and computation requirements per video for the best-performing run for each approach. The storage requirements are measured in bytes (B) and the retrieval time in milliseconds (ms).

Method	**GV**	**BoW**	**LBoW**	**DML**	**HC**
Storage space	16,384	209	3,050	2,048	512
Retrieval time	499	152	1,155	333	51

each method and for each evaluation task. Similar conclusions apply as in the case of their mAP comparison. LBoW outperforms other runs consistently for all three tasks by a significant margin. However, there is a large gap between the BoW and the other three runs. A reasonable explanation is that the BoW representation retains local information from the video frames, in contrast to the other aggregation methods that average frame descriptors in a global video representation. This is of critical importance for all three tasks since only a minority of similar videos share their entire content to the queries. Similar to the mAP evaluation, GV performs poorly for all retrieval tasks compared to the other schemes.

Retrieval time and memory requirements

Table 3.9 presents the requirements in terms of storage space and computation time for the best-performing run of each method. The results of all methods have been measured using the open-source library Scikit-learn [126] in Python on a Linux PC with a 4-core i7-4770K and 32GB of RAM. It is noteworthy that LBoW's superior performance comes at a high computational and storage cost. In particular, it needs approximately 1.2s per query to perform retrieval (being the slowest among the five approaches) and 3KB per video to store the video representations. The fastest method is the HC with 51ms per query, followed by BoW with three times slower retrieval time. The method that requires the least memory space in RAM is BoW reserving only 209B per video. DML is in the middle of the rank for both measures. The most demanding method in terms of storage space is GV requiring approximately 16KB for each video descriptor. Note that these figures are derived from computing video similarities for

Table 3.10: mAP of the benchmarked approaches built based on the FIVR-200K training set and evaluated on the FIVR-200K test set for the three retrieval tasks.

Run	DSVR	CSVR	ISVR
GV	0.389	0.370	0.301
BoW	0.302	0.287	0.237
LBoW	0.362	0.344	0.280
DML	0.465	0.443	0.381
HC	**0.468**	**0.444**	**0.382**

one query at a time, without vectorizing all query descriptors in a single matrix. This practice would significantly decrease retrieval time for all methods.

Within-dataset retrieval

Our initial goal for the construction of the FIVR-200K is to be used for evaluation purposes in its entirety. However, it is not always possible to have access to a separate dataset that simulates the same or a similar retrieval problem. To overcome this issue, we have also devised a within-dataset experimental setup, where we split the dataset into two separate video sets, one for the development/training of the methods and one for evaluation[5]. To do so, we order the videos based on their publication time and then split them in half, resulting in two sets of videos from different time periods. We select the early period video set for training and the late period video set for testing. The total number of queries in the training and test set are 31 and 69, respectively.

Table 3.10 presents the performance of the benchmarked approaches in the three evaluation tasks. There is a considerable decrease in terms of mAP for BoW and LBoW runs, reaching approximately half their performance compared to the previous runs for all three tasks (≈46-51%). We observed similar decreases in performance when using VCDB for development (i.e., generation of visual codebooks) and the whole FIVR-200K for testing. This makes clear that BoW-based schemes are quite sensitive to the dataset that is used for generating the underlying visual codebooks. There is also a negligible drop in performance for the HC scheme (less than 0.01 in

[5]The dataset split is only applied in this subsection.

terms of mAP); hence, in this setting, HC achieves the best results among all methods for all three tasks. As expected, DML is boosted when using part of the FIVR-200K for training. The improvement for DML in all tasks ranges between 0.03 and 0.05. Finally, the GV approach also sees a small improvement in all evaluation tasks compared to the initial results.

Dataset availability

Finally, we have noticed that there is a significant amount of videos that are no longer available on YouTube since the dataset collection [92]. Unfortunately, this is an ongoing trend, and it will only get worse in the following years. As of May 2020, from the 225,960 YouTube videos of the FIVR-200K dataset, only 187,311 are still available on YouTube, meaning that almost 40,000 videos of the dataset have been removed or restricted. This corresponds to a reduction of 17,1% in the total amount of videos. Various reasons account for video unavailability, e.g., deletions by the uploader, violation of their terms of service, copyright infringement, geographical restrictions. Video unavailability has a considerable impact on the reproducibility of the experiments and the fair comparisons between proposed approaches. Thus, we provide the extracted features publicly available[6] to facilitate future research on the FIVR problem.

3.4 Conclusion

In this chapter, we introduced the problem of Fine-grained Incident Video Retrieval (FIVR). First, we provided definitions for the various types of video associations arising in the more general problem setting of FIVR. Next, we built a large-scale dataset, FIVR-200K, with the aim of addressing the benchmarking needs of the problem. The dataset comprises of 225,960 YouTube videos, collected based on approximately 5,000 global news events crawled from Wikipedia over five years (2013-2017). Then, we selected 100 queries based on a principled approach that automatically assessed the

[6] `http://ndd.iti.gr/features/`

suitability of a query video for performing evaluations for the current problem. We also devised a protocol for annotating the dataset according to four labels for video pairs. Finally, we conducted a thorough experimental study on the dataset comparing five state-of-the-art methods, six feature extraction methods, five CNN/3D CNN architectures, and four video descriptor aggregation schemes. For the benchmark, we considered three retrieval tasks that represented different instances of the problem and accepted different labels as relevant, i.e., DSVR, CSVR, and ISVR. The best-performing methods achieved mAP scores of 0.710, 0.675, and 0.572, respectively. In general, retrieval performance across all experiments was not very high compared to performance values that have been reported for related datasets, such as CC_WEB_VIDEO. This demonstrates that the proposed problem and associated dataset offer a challenging setting with considerable room for improvement, especially in the case of the ISVR task.

Chapter 4

Video similarity calculation on video-level representations

Contents

In this chapter, we propose two video-level methods based on deep learning features. They have been initially designed for the problem of NDVR, which is closely related to our FIVR problem. Nevertheless, they can be directly applied for FIVR with slight modifications as presented in the previous chapter (Chapter 3). The first is an unsupervised scheme that relies on a modified Bag-of-Word (BoW) video representation. The second is a supervised method based on Deep Metric Learning (DML). For the development of both methods, features are extracted from the intermediate layers of Convolutional Neural Networks (CNN) and leveraged as frame descriptors since they offer a compact and informative image representation and lead to increased system performance. Extensive evaluation has been conducted on publicly available bench-

mark datasets, and the presented methods are compared with state-of-art approaches, achieving the best results in all evaluation setups. The implementations of the feature extraction and the DML scheme are publicly available[1,2].

Motivated by the excellent performance of deep learning in a wide variety of multimedia problems, we have developed two video-level approaches that incorporate deep learning and can be used in different application scenarios. The main contributions of this chapter are:

- A feature extraction process based on the activations of intermediate convolutional layers [105, 146] of pre-trained Convolutional Neural Networks (CNNs). Given an input frame to the CNN network, we apply the Maximum Activations of Convolutions (MAC) function on the activations of each convolutional layer. This process generates compact frame representations that are translation invariant and encodes information from several granularity levels.

- An unsupervised approach that is a variation of the traditional Bag-of-Words scheme. We propose a layer aggregation technique, with *tf-idf* weighting and organisation in an inverted file structure for fast retrieval. This method does not need labelled data, and as a result it can be applied on any video corpus.

- A supervised solution leveraging Deep Metric Learning (DML) that overcomes several limitations of the BoW approach (i.e., volatile performance on unseen data, computationally expensive retraining). We set up a DML framework based on a triplet-wise scheme to learn a compact and efficient embedding function. A significant benefit of the learning scheme is that it allows being trained in various scenarios; thus, it provides flexibility with respect to the FIVR definition.

The remainder of the chapter is organized as follows: In Section 4.1, we describe the

[1] https://github.com/MKLab-ITI/intermediate-cnn-features
[2] https://github.com/MKLab-ITI/ndvr-dml

feature extraction process employed to generate frame-level descriptors. In Sections 4.2 and 4.3, we present the two proposed approaches, i.e., the BoW and DML approach, respectively. In Section 4.4, we report on the results of a comprehensive experimental study, including a comparison with several state-of-the-art methods. In Section 4.5, we summarize the findings of our work.

4.1 CNN-based feature extraction

In recent research [142, 146, 124], pre-trained CNN models are used to extract visual features from intermediate convolutional layers. These features are computed through the forward propagation of an image over the CNN network and the use of an aggregation function (e.g. VLAD encoding [46, 8, 87], max/average pooling [9, 124, 97]) on the convolutional layer.

We adopt a compact representation for frame descriptors, derived from activations of all intermediate convolutional layers of a pre-trained CNN by applying the function called Maximum Activation of Convolutions (MAC) [124, 146, 97]. A pre-trained CNN network Θ is considered, with a total number of L convolutional layers, denoted as $\mathcal{L}^1, \mathcal{L}^2, ..., \mathcal{L}^L$. Forward propagating a frame through network Θ generates a total of L feature maps, denoted as $\mathcal{M}^l \in \mathbb{R}^{w^l \times h^l \times c^l} (l = 1, ..., L)$, where $w^l \times h^l$ is the spatial dimensions of every channel for convolutional layer $\mathcal{L}^l$ (which depends on the size of the input frame) and c^l is the total number of channels. An aggregation function is applied on the above feature maps to extract a single descriptor vector from every layer. In particular, we apply max pooling on every channel of feature map $\mathcal{M}^l$ to extract a single value. The extraction process is formulated in:

$$v^l(i) = \max \ \mathcal{M}^l(\cdot, \cdot, i), \quad i = \{1, 2, ..., c^l\} \tag{4.1}$$

where layer vector v^l is a c^l-dimensional vector derived from max pooling on every channel of feature map $\mathcal{M}^l$. The layer vectors are then ℓ_2-normalized. This process encodes the maximum activation of each of the convolutional filters; hence the ex-

tracted features are translation invariant. Also, since multiple convolutional layers are employed that process frames at different scales, the generated descriptors capture information at various granularity levels.

We extract and concatenate frame descriptors only from activations in intermediate layers since we aim to construct a visual representation that preserves local structure at different scales. Activations from fully-connected layers are not used since they are considered to offer global representation of the input. A positive side-effect of this decision is that the resulting descriptor is compact, reducing the total processing time and storage requirements. For very deep architectures (e.g., VGGNet, GoogLeNet), we do not extract features from the initial layer activations, since those layers are expected to capture very primitive frame features (e.g., edges and corners) that could lead to false matches.

Uniform sampling is applied to select one frame per second for every video and extract the respective features. Hence, given an arbitrary video with a total duration of N seconds and an equal number of selected frames $\{F_1, F_2, ..., F_N\}$, the video representation is a set that contains all feature vectors of the video frames $v = \{v_{F_1}, v_{F_2}, ..., v_{F_N}\}$, where v_{F_i} contains all layer vectors of frame F_i. Although v_{F_i} stands for a set of vectors, we opted to use this notation for convenience.

4.2 Video representation based on Bag-of-Words

In this section, an unsupervised NDVR approach is presented that relies on the Bag-of-Word (BoW) scheme. In particular, two aggregation variations are proposed: a vector aggregation where a single codebook of visual words is used, and a layer aggregation where multiple codebooks of visual words are used. The video representations are organised in an inverted file structure for fast indexing and retrieval. The video similarity is computed based on the cosine similarity of the *tf-idf* weighted vectors of the extracted BoW representations.

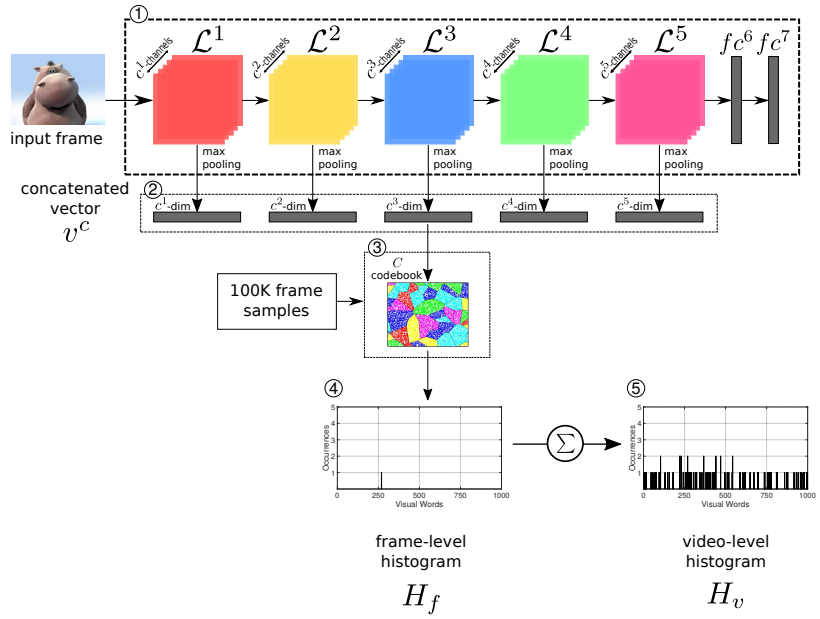
Feature aggregation

We follow two alternative feature aggregation schemes (i.e., ways of aggregating features from layers into a single descriptor for the whole frame): a) *vector aggregation* and b) *layer aggregation*. The outcome of both schemes is a frame-level histogram, H_F, which is considered the frame representation. Next, a video-level histogram H_V is derived from the frame representations by aggregating frame-level histograms to a single video representation. Figure 4.1 illustrates the two schemes.

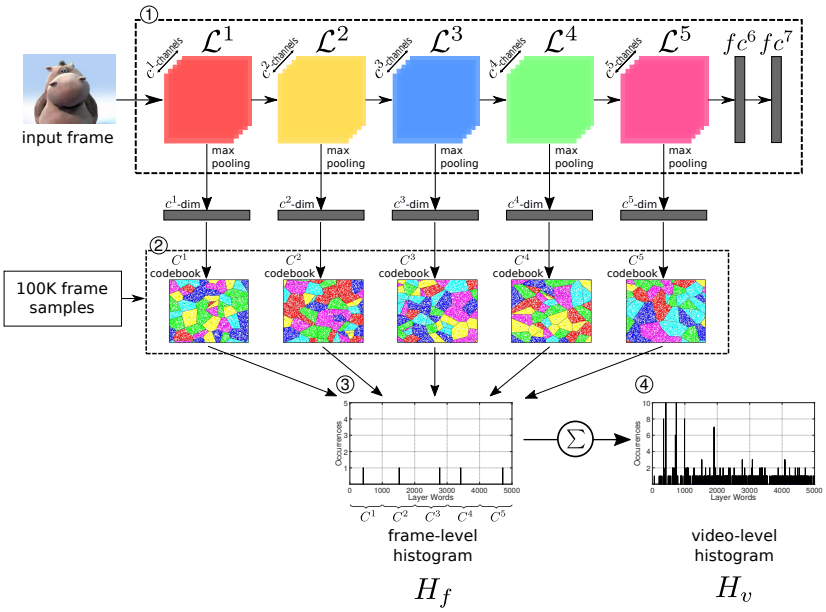
Vector aggregation: A bag-of-words scheme is applied on the vector v^c resulting from the concatenation of individual layer features to generate a single codebook of K visual words, denoted as $C_K = \{t_1, t_2, ..., t_K\}$. The selection of K, a system parameter, has a critical impact on the performance of the approach, further explored in section 4.4.2. Having generated the visual codebook, every video frame is assigned to the nearest visual word. Accordingly, every frame F_i with feature descriptor $v^c_{F_i}$ is aggregated to the nearest visual word $t_{F_i} = NN_{C_K}(v^c_{F_i})$, hence its H_{F_i} contains only a single visual word.

Layer aggregation: To preserve the structural information captured by intermediate layers L of the CNN network Θ, we generate L layer-specific codebooks of K words (denoted as $C^l_K = \{t^l_1, t^l_2, ..., t^l_K\}, l = 1, ..., L$), which we then use to extract separate bag-of-words representations (one per layer). The layer vectors $v^l_{F_i}$ of frame F_i are mapped to the nearest layer words $t^l_{F_i} = NN_{C^l_K}(v^l_{F_i})$, $(l = 1, 2, ..., L)$. In contrast to the previous scheme, every frame F_i is represented by a frame-level histogram H_{F_i} that results from the concatenation of the individual layer-specific histograms, thus comprising L words instead of a single one.

The final video representation is generated based on the BoW representations of its frames. In particular, given an arbitrary video with N frames $\{F_1, F_2, ..., F_N\}$, its video-level histogram H_V is derived by summing the histogram vectors corresponding



(a) Vector Aggregation



(b) Layer Aggregation

Figure 4.1: Overview of the two proposed aggregation schemes and the final video representation. Vector aggregation (top): the layer vectors extracted from the intermediate layers are concatenated to a single frame-level representation, then mapped to a visual word and aggregated to a video representation. Layer aggregation (bottom): the layer vectors are mapped to multiple visual words independently, and then are aggregated to a video representation

to its frames, i.e. $H_V = \sum_{i \in [1,N]} H_{F_i}$. Note that for the two aggregation schemes, histograms of different sizes are generated. In the first case, the total number of visual words equals K, whereas in the second case, it equals $K \cdot L$.

Video Indexing and Querying

In the proposed approach, we use *tf-idf* weighting to calculate the similarity between two video histograms. The *tf-idf* weights are computed for every visual word in every video in a video collection C_b:

$$w_{td} = n_{td} \cdot \log |C_b| / n_t \tag{4.2}$$

where w_{td} is the weight of word t in video d, n_{td} and n_t are the number of occurrences of word t in video d and the entire collection respectively, while $|C_b|$ is the number of videos in the collection. The former factor of the equation is called *term frequency* (tf) and the latter is called *inverted document frequency* (idf).

Video querying is the online part of the approach. Let q denote a query video. Once the final histogram H_v^q is extracted from the query video, an inverted file indexing scheme [111] is employed for fast and efficient retrieval of videos that have at least a common visual word with the query video. For all these videos (i.e. videos with non-zero similarity), the cosine similarity between the respective *tf-idf* representations is computed:

$$\mathrm{S}_{bow}(q,p) = \frac{\mathbf{w}_q \cdot \mathbf{w}_p}{\|\mathbf{w}_q\| \, \|\mathbf{w}_p\|} = \frac{\sum_{i=0}^{K} w_{iq} w_{ip}}{\sqrt{\sum_{i=0}^{K} w_{iq}^2} \sqrt{\sum_{i=0}^{K} w_{ip}^2}} \tag{4.3}$$

where $\mathrm{S}_{bow}(\cdot,\cdot)$ is the similarity function based on the BoW scheme which calculates the similarity between two given videos, $\mathbf{w}_q$ and $\mathbf{w}_p$ are the weight vectors of videos q and p, respectively, and $\|\mathbf{w}\|$ is the norm of vector $\mathbf{w}$. The database videos are ranked in descending order based on their similarity to the query.

In the inverted file structure, each entry corresponds to a visual word and contains its ID, the *idf* value, and all the video IDs in which the visual word occurs. The video IDs map to videos in the collection C_b, where the occurrences (tf) of the visual words are stored. With this inverted file structure, all the needed values for the calculation of the similarity between a query and a dataset video can be retrieved.

4.3 Learn video embeddings with Deep Metric Learning

The unsupervised approach has several limitations. The most important is that it offers a dataset-specific solution, i.e., the extracted knowledge is not transferable, and re-building the model is computationally expensive. A sufficiently large and diverse dataset to create vocabularies is required to observe no performance loss, which needs significant effort to be collected or sometimes is not even possible. We have experimentally validated that even external large-scale datasets (such as ImageNet [24]) are not adequate to build robust models. Also, the retraining of the BoW method with new samples from previously unseen data is inefficient due to the significant amount of time needed for codebook learning and video indexing. For codebook training and video indexing of large-scale datasets, several processing days are required, e.g., for two hundred thousand videos. Hence, we have also developed a Deep Metric Learning (DML) approach to overcome these limitations. This involves training a Deep Neural Network (DNN) to approximate an embedding function for the accurate computation of similarity between two candidate videos. For training, we devised a novel triplet generation process.

For feature extraction, we build upon the same process as the one presented in Section 4.1. Hence, given an arbitrary video with N frames $\{F_1, F_2, ..., F_N\}$, we extract one feature descriptor for each video frame by concatenating the layer vector to a single vector. Global video representations v are then derived by averaging and normalizing (zero-mean and ℓ_2-normalization) these frame descriptors.

Moreover, we should also note that feature extraction is not part of the training (deep metric learning) process, i.e., the training of the network is not end-to-end, and as a result, the weights of the pre-trained network used for feature extraction are not updated. We have empirically validated such settings, and the network's performance significantly drops when trained end-to-end. A possible explanation for this could be attributed to the different domains represented by the training and evaluation dataset, considering that each dataset represents a domain. The network is trained on VCDB; hence, it learns the limited domain represented by this dataset. As a result, the feature extraction CNN fails to transfer knowledge and generalize to the domains of the evaluation dataset, and therefore the performance drops. However, when the feature extraction network remains fixed, with weights trained on ImageNet [24], the extracted representations encode video content in a much broader and more diverse domain, leading to better generalization across different datasets and ultimately to better retrieval performance.

4.3.1 Problem Setting

We address the problem of learning a pairwise similarity function for NDVR from the relative information of pairwise/triplet-wise video relations. For a given query video and a set of candidate videos, the goal is to quantify the similarity between the query and every candidate video and use it for the ranking of the entire set of candidates in the hope that the NDVs are retrieved at the top ranks. To formulate this process, we define the similarity between two arbitrary videos q and p as the squared Euclidean distance in the video embedding space (Equation 4.4).

$$\mathrm{D}(f_\theta(q), f_\theta(p)) = \|f_\theta(q) - f_\theta(p)\|_2^2 \tag{4.4}$$

where $f_\theta(\cdot)$ is the embedding function that maps a video to a point in the Euclidean space, θ are the system parameters and $\mathrm{D}(\cdot,\cdot)$ is the squared Euclidean distance in this space. Additionally, we define a pairwise indicator function $\mathrm{I}(\cdot,\cdot)$ that specifies

whether a pair of videos are near-duplicate.

$$\mathrm{I}(q,p) = \begin{cases} 1 & \text{if } q,p \text{ are NDVs} \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

Our objective is to learn an embedding function $f_\theta(\cdot)$ that assigns smaller distances to NDV pairs than others. Given a video v, a NDV v^+ and a dissimilar video v^-, the embedding function $f_\theta(\cdot)$ should map videos to a common space $\mathbb{R}^d$, where d is the dimension of the feature embedding, in which the distance between query v and positive v^+ is always smaller than the distance between query v and negative v^- (Equation 4.6).

$$\begin{aligned} &\mathrm{D}(f_\theta(v), f_\theta(v^+)) < \mathrm{D}(f_\theta(v), f_\theta(v^-)), \\ &\forall v, v^+, v^- \text{ such that } \mathrm{I}(v,v^+) = 1, \mathrm{I}(v,v^-) = 0 \end{aligned} \tag{4.6}$$

4.3.2 Triplet loss

To implement the learning process, we create a collection of N training instances organized in the forms of triplets $\mathcal{T} = \{(v_i, v_i^+, v_i^-), i = 1, ..., N\}$, where v_i, v_i^+, v_i^- are the feature vectors of the query, positive (NDV), and negative (dissimilar) videos. A triplet expresses a relative similarity order among three videos, i.e., v_i is more similar to v_i^+ in contrast to v_i^-. We define the following hinge loss function for a given triplet called 'triplet loss' (Equation 4.7).

$$\mathcal{L}_\theta(v_i, v_i^+, v_i^-) = \max\{0, \mathrm{D}(f_\theta(v_i), f_\theta(v_i^+)) - \mathrm{D}(f_\theta(v_i), f_\theta(v_i^-)) + \gamma\} \tag{4.7}$$

where γ is a margin parameter to ensure a sufficiently large difference between the positive-query distance and negative-query distance. If the video distances are calculated correctly within margin γ, then this triplet will not be penalised. Otherwise the loss is a convex approximation of the loss that measures the degree of violation of the desired distance between the video pairs specified by the triplet. To this end, we use

batch gradient descent to optimize the objective function described in Equation 4.8.

$$\min_{\theta} \sum_{i=1}^{m} \mathcal{L}_{\theta}(v_i, v_i^+, v_i^-) + \lambda \left\|\theta\right\|_2^2 \tag{4.8}$$

where λ is a regularization parameter to prevent overfitting of the model, and m is the total size of a triplet mini-batch. Minimising this loss will narrow the query-positive distance while widening the query-negative distance, and thus lead to a representation satisfying the desirable ranking order. With an appropriate triplet generation strategy in place, the model will eventually learn a video representation that improves the effectiveness of the NDVR solution.

Although the current method has been proposed for NDVR, its adaptation for the FIVR problem is simple and straightforward. To train the network for FIVR, we form triplets using any video pair labeled as DSV, CSV, or ISV as the positive pairs. In that way, we generate triplets for all video associations related to FIVR.

4.3.3 DML network architecture

For training the DML model, a triplet-based network architecture is proposed (Figure 4.2) that optimizes the triplet loss function of Equation 4.7. The network is provided with a set of triplets $\mathcal{T}$ created by the triplet generation process of section 4.3.5. Each triplet contains a query, a positive and a negative video with v_i, v_i^+, and v_i^- feature vectors, respectively, which are fed independently into three siamese DNNs with identical architecture and parameters. The DNNs compute the embeddings of $v : f_\theta(v) \in \mathbb{R}^d$. The architecture of the deployed DNNs is based on three dense *fully-connected layers* and a *normalization layer* at the end leading to vectors that lie on a d-dimensional unit length hypersphere, i.e. $\left\|f_\theta(v)\right\|_2 = 1$. The size of each hidden layer (number of neurons) and the d-dimension of the output vector $f_\theta(v)$ depends on the dimensionality of input vectors, which is in turn dictated by the employed CNN architecture. The video embeddings computed from a batch of triplets are then given to a triplet loss layer to calculate the accumulated cost based on Equation 4.7.

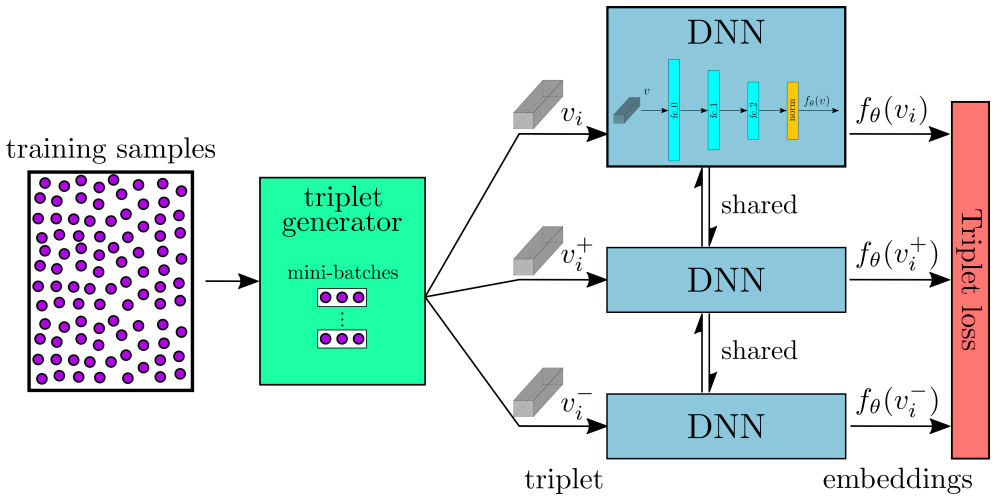


Figure 4.2: Overview of the DML network architecture for the training of the DNN network. A triplet generator organizes the training samples in triplets of a query, a positive (NDV), and a negative video. The video vectors of the triplets are fed to the DNN to generate the video embeddings. The network is trained by minimizing the triplet loss function.

4.3.4 Video-level similarity computation

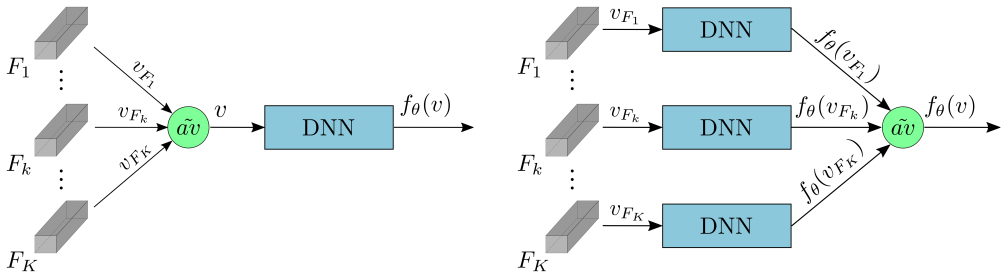
The learned embedding function $f_\theta(\cdot)$ is used for computing similarities between videos in a target video corpus. Given an arbitrary video with $v = \{v_{F_1}, v_{F_2}, ..., v_{F_N}\}$, two variants are proposed for fusing similarity computation across video frames: early and late fusion (Figure 4.3).

Early fusion: Frame descriptors are averaged and normalized into a global video descriptor before they are forward propagated to the network. The global video signature is the output of the embedding function $f_\theta(\cdot)$:

$$f_\theta(v) = f_\theta(\frac{1}{N}\sum_{i=1}^{N} v_{F_i}) \tag{4.9}$$

Late fusion: Each extracted frame descriptor of the input video is fed to the network, and the set of their embedding transformations is averaged and normalized:

$$f_\theta(v) = \frac{1}{N}\sum_{i=1}^{N} f_\theta(v_{F_i}) \tag{4.10}$$



(a) early fusion (b) late fusion

Figure 4.3: Illustration of early and late fusion schemes.

Even though the network has been trained using the early fusion scheme, both schemes are directly applicable to extract video embedding. Their main difference is that the early fusion operates on video-level, mapping the video as a whole into the embedding space. Whereas the late fusion encodes the information in frame-level, mapping each video into the embedding space independently and then generating the video representations.

There are several pros and cons to each scheme. The former is computationally lighter and more intuitive; however, it is slightly less effective. Late fusion leads to better performance and is amenable to possible extensions of the base approach (i.e., frame-level approaches). Nonetheless, it is slower since the features extracted from all selected video frames are fed to the DNN.

Finally, the similarity between two videos derives from the distance of their representations. For a given query q and a set of M candidate videos $\{p_i\}_{i=1}^{M} \in P$, the similarity within each candidate pair is determined by normalizing the distance with respect to the maximum value and then subtracting the result from the unit to map the similarity scores to the range [0, 1]. This process is formulated in:

$$\mathrm{S}_{dml}(q,p) = 1 - \frac{\mathrm{D}(f_\theta(q), f_\theta(p))}{\max\limits_{p_i \in P}(\mathrm{D}(f_\theta(q), f_\theta(p_i)))} \tag{4.11}$$

where $\mathrm{S}_{dml}(\cdot,\cdot)$ is the similarity function based on the DML scheme which calculates the similarity between two given videos, and $\max(\cdot)$ is the maximum function.

4.3.5 Triplet Generation

Generation process

A crucial part of the proposed approach is the generation of video triplets. It is important to provide a considerable amount of videos for constructing a representative triplet training set. However, the total number of triplets that can be generated equals the total number of 3-combinations over the size of N of the video corpus, i.e.:

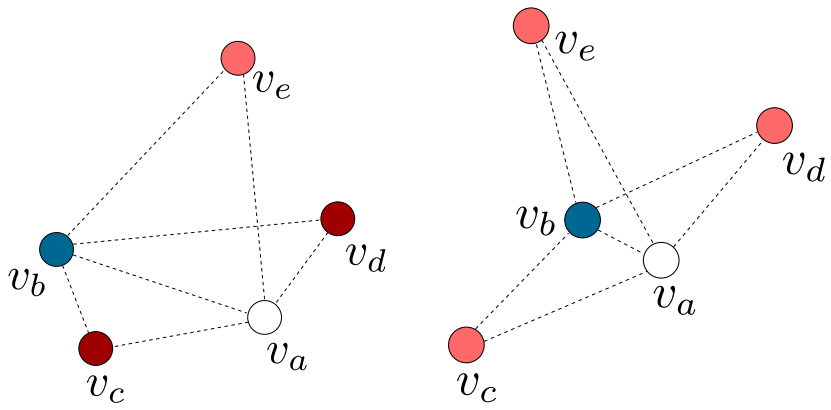
$$\binom{N}{3} = \frac{N \cdot (N-1) \cdot (N-2)}{6} \tag{4.12}$$

We have empirically determined that only a tiny portion of videos in a corpus could be considered near-duplicates for a given video query. Thus, it would be inefficient to randomly select video triplets from this vast set (for example, for $N = 1,000$, the total number of triplets would exceed 160M). Instead, a sampling strategy is employed as a key element of the triplet generation process, which is focused on selecting hard candidates to create triplets, i.e., triplets that will generate non-zero loss during training.

The proposed sampling strategy is applied on a development dataset. Such a dataset needs to contain two sets of videos: $\mathcal{P}$, a set of near-duplicate video pairs that are used as query-positive pairs, and $\mathcal{N}$, a set of dissimilar videos that are used as negatives. We aim at generating *hard triplets*, i.e., negative videos (*hard negatives*) with distance to the query that is smaller than the distance between the query and positive videos (*hard positives*). The aforementioned condition is expressed in Equation 4.13.

$$\mathcal{T} = \{(q,p,n) | (q,p) \in \mathcal{P}, n \in \mathcal{N}, \mathrm{D}(q,p) > \mathrm{D}(q,n)\} \tag{4.13}$$

where $\mathcal{T}$ is the resulting set of triplets. The global video features are first extracted following the process of section 4.1. Then, the distance between every query in $\mathcal{P}$ and every dissimilar video in $\mathcal{N}$ is calculated. If the query-positive distance is greater than a query-negative distance, then a hard triplet is formed composed of the three videos. The distance is calculated based on the Euclidean distance of the initial global video descriptors.



(a) Before training (b) After training

Figure 4.4: Examples of video representations in feature space before and after training. Colours: (white) query video (blue) NDV (red) distractor videos.

Figure 4.4 visualizes the training and triplet generation process. Figure 4.4(a) depicts the videos in feature space before training. The white and blue colour circles represent the query and near-duplicate videos, respectively, whereas the dissimilar videos are painted in red colour. In particular, v_a is the query, and v_b is a NDV. However, before training, it is clear that their distance D_{ab} is greater than distances D_{ac} and D_{ad}; therefore, v_c and v_d (deep red) are *hard negatives* and two triplets will be created $\{v_a, v_b, v_c\}$ and $\{v_a, v_b, v_d\}$. The video v_e (light red) does not generate any triplet because its distance from the two NDVs is greater than the distance between them. After training, the distance between the query and the NDV must be smaller than their distance to any other dissimilar video, as illustrated in Figure 4.4(b).

Development dataset

For generating triplets to train the supervised DML approach, we leverage the VCDB dataset [51]. This dataset is composed of videos from popular video platforms (YouTube and Metacafe) and has been compiled and annotated as a benchmark for the partial copy detection task, which is highly related to the NDVR problem setting. VCDB contains two subsets, the core $\mathcal{C}_c$ and the distractor subset $\mathcal{C}_d$. Subset $\mathcal{C}_c$ contains discrete sets of videos composed of 528 query videos and over 9,000 pairs of

partial copies. Each video set has been annotated, and the chunks of the video copies extracted. Subset $\mathcal{C}_d$ is a corpus of approximately 100,000 distractor videos, which is used to make the video copy detection problem more challenging.

For the triplet generation, we retrieve all video pairs annotated as partial copies. We define an overlap criterion to decide whether to use a pair for the triplet generation: if the duration of the overlapping content is greater than a certain threshold t compared to the total duration of each video, then the pair is retained, otherwise discarded. Each video of a given pair can be used once as query and once as positive video. Therefore, the set of query-positive pairs $\mathcal{P}$ is generated based on:

$$\mathcal{P} = \{(q,p) \cup (p,q) | q,p \in \mathcal{C}_c, \mathrm{o}(q,p) > t\} \tag{4.14}$$

where $\mathrm{o}(\cdot,\cdot)$ determines the video overlap. Subset $\mathcal{C}_d$ is used as the set $\mathcal{N}$ of negatives. To generate hard triplets, the negative videos are selected based on Equation 4.13.

4.4 Experimental study

In this section, the two developed approaches are evaluated. The experimental setup is described in Section 4.4.1, where we present the evaluation datasets used, several implementation details, and a number of competing approaches from the state-of-the-art. Extensive experimental evaluation is conducted and reported under various evaluation settings for BoW and DML approaches in Section 4.4.2 and Section 4.4.3, respectively. We use mAP and PR-curve as evaluation metrics for all the experiments, as defined in 3.3.1.

4.4.1 Experimental setup

Evaluation datasets

Experiments were performed on the CC_WEB_VIDEO dataset [134], which is publicly available. The collection consists of a sample of videos retrieved by submitting 24

popular text queries to popular video sharing websites (i.e., YouTube, Google Video, and Yahoo! Video). For every query, a set of video clips were collected and annotated based on their near-duplicate relation to the query video. The dataset contains a total of 12,790 videos and 24 queries, one for each set. Table 4.1 depicts the types of near-duplicate types and their annotation. In the present work, all videos annotated with any symbol but X are considered near-duplicates.

In addition, we use the FIVR-200K [63] dataset (Chapter 3) for validating the results on the FIVR problem. It consists of 225,960 videos collected based on the 4,687 events and contains 100 video queries. Table 4.1 depicts the annotation labels used in the dataset and their definitions. FIVR-200K includes three different tasks: a) the Duplicate Scene Video Retrieval (DSVR) task where only videos annotated with ND and DS are considered relevant, b) the Complementary Scene Video Retrieval (CSVR) task which accepts only the videos annotated with ND, DS or CS as relevant, and c) Incident Scene Video Retrieval (ISVR) task where all labels (with the exception of DI) are considered relevant.

Table 4.1: Annotation labels of CC_WEB_VIDEO and FIVR-200K datasets.

(a) CC_WEB_VIDEO

Label	**Transformation**
E	Exactly duplicate
S	Similar video
V	Different version
M	Major change
L	Long version
X	Dissimilar video

(b) FIVR-200K

Label	**Definition**
ND	Near-duplicate
DS	Duplicate scene
CS	Complementary scene
IS	Incident scene
DI	Distractor

Implementation details

We experiment with three deep network architectures: AlexNet [68], VGGNet [110] and GoogLeNet [118]. The AlexNet is an 8-layer network that consists of five convolutional/pooling layers, two fully-connected layers and one softmax layer. VGGNet has the same number of fully-connected layers, although the number of convolutional

Table 4.2: Deep CNN architectures and total number of channels per layer used in the proposed approach.

(a) AlexNet

Layer $\mathcal{L}^l$	c^l-dim
conv1	96
conv2	256
conv3	384
conv4	384
conv5	256
total	1376

(b) VGGNet

Layer $\mathcal{L}^l$	c^l-dim
conv2_1	128
conv2_2	128
conv3_1	256
conv3_2	256
conv3_3	256
conv4_1	512
conv4_2	512
conv4_3	512
conv5_1	512
conv5_2	512
conv5_3	512
total	4096

(c) GoogLeNet

Layer $\mathcal{L}^l$	c^l-dim
inception_3a	256
inception_3b	480
inception_4a	512
inception_4b	512
inception_4c	512
inception_4d	528
inception_4e	832
inception_5a	832
inception_5b	1024
total	5488

layers may vary. In this paper, the version with 16-layers is employed as it gives similar performance to the 19-layer version. Finally, GoogLeNet is composed of 22 layers in total. In this architecture, multiple convolutions are combined in an intersection module called "inception". There are nine inception modules in total that are sequentially connected, followed by an average pooling and a softmax layer at the end. All three architectures receive as input images of size 224×224. For all the experiments, the input frames are resized to fit these dimensions, even though this step is not mandatory. Table 4.2 depicts the employed CNN architectures and the number of channels in the respective convolutional layers.

For feature extraction, we use the Caffe framework [47], which provides pre-trained models on ImageNet [24] for all employed CNN networks[3]. Regarding the unsupervised BoW approach, the visual codebooks are generated based on scalable K-Means++ [10] – the Apache Spark[4] implementation of the algorithm is used for efficiency and scalability – in both aggregation schemes, a sample of 100K randomly selected video frames are used for training.

[3] https://github.com/BVLC/caffe/wiki/Model-Zoo
[4] http://spark.apache.org

The implementation of the supervised DML model is built on Theano [7]. We use [800, 400, 250], [2000, 1000, 500] and [2500, 1000, 500] neurons for the three hidden layers for AlexNet, VGGNet and GoogLeNet respectively. Adam optimization [60] is employed with learning rate 10^{-5}. For the triplet generation, we set $t = 0.8$, which generates approximately 2k pairs in $\mathcal{P}$ and 7M, 4M, and 5M triplets in $\mathcal{T}$, for AlexNet, VGGNet, and GoogLeNet, respectively. Other parameters are set to $\gamma = 1$, $\lambda = 10^{-5}$ and $m = 1000$.

State-of-the-art approaches

We compare the proposed approach with five widely used content-based NDVR approaches. Three of those were developed based on frames of videos sampled from the evaluation set. These are the following:

Auto Color Correlograms (ACC) - Cai et al. [18] use uniform sampling to extract one frame per second for the input video. The auto-color correlograms [41] of each frame are computed and aggregated based on a visual codebook generated from a training set of video frames. The retrieval of related videos is performed using tf-idf weighted cosine similarity over the visual word histograms of a query and a dataset video.

Pattern-based approach (PPT) - Chou et al. [22] build a pattern-based indexing tree (PI-tree) based on a sequence of symbols encoded from keyframes, which facilitates the fast filtering of candidate videos. They use m-pattern-based dynamic programming (mPDP) and time-shift m-pattern similarity (TPS) to determine video similarity.

Stochastic Multi-view Hashing (SMVH) - Hao et al. [35] combine multiple keyframe features to learn a group of mapping functions that project video keyframes into the Hamming space using Kullback-Leibler (KL) divergence. The combination of keyframe hash codes generates a video signature that constitutes the final video representation. The Hamming distance is used to rank videos.

The remaining two, which are based on the work of Wu et al. [134], are not built based on any development dataset:

Color Histograms (CH) - This is a global video representation based on the color histograms of keyframes. The color histogram is a concatenation of 18 bins for Hue, 3 bins for Saturation, and 3 bins for Value, resulting in a 24-dimensional vector representation for every keyframe. The global video signature is the normalized color histogram over all keyframes in the video.

Local Structure (LS) - Global signatures and local features are combined using a hierarchical approach. Color signatures are employed to detect relevant videos with high confidence and to filter out very dissimilar videos. For the reduced set of candidate videos, a local feature based method is employed, which compares the keyframes in a sliding window using their local features (PCA-SIFT [57]).

4.4.2 Evaluation of BoW approach

Comparison of global feature descriptors

In this section, we benchmark the proposed intermediate CNN features with a number of global frame descriptors used in the literature. The compared descriptors are divided in two groups: handcrafted and learned features[5]. The handcrafted features include RGB histograms, HSV histograms, Local Binary Patterns (LBP), Auto Colour Correlograms (ACC) and Histogram of Oriented Gradients (HOG). For the learned features, we extract the intermediate CNN features, as described in Section 4.1, and concatenate the layer vectors to generate a single descriptor. Additionally, we experiment with the global features derived from the activations of the first fully connected layer after the convolutional layers, for each architecture. To compare the retrieval performance, a standard bag-of-word scheme with vector aggregation (Section 4.2) is

[5]The features have been learned on the ImageNet [24] dataset, since pre-trained networks are utilized. However, ImageNet is a comprehensive dataset, so the learned features can be used in other computer vision tasks (e.g., image/video retrieval) without the need of retraining.

Table 4.3: mAP and dimensionality of eleven global frame descriptors.

Descriptor/ Network	**layers**	**dimensions**	**K**	
			1000	10,000
RGB	-	64	0.857	0.813
HSV	-	162	0.902	0.792
LBP	-	256	0.803	0.683
ACC	-	256	0.936	0.826
HOG	-	1764	**0.940**	**0.831**
AlexNet	int	1376	0.951	0.879
	fc	4096	0.953	0.875
VGGNet	int	4096	0.937	**0.886**
	fc	4096	0.936	0.854
GoogLeNet	int	5488	**0.958**	0.857
	fc	1000	0.941	0.849

built based on each global feature descriptor. Table 4.3 presents the mAP of each model built on a different global descriptor for two different values of K. The intermediate features of GoogLeNet and VGGNet achieved the best results with 0.958 and 0.886 for $K = 1,000$ and $K = 10,000$, respectively. In general, learned features lead to considerably better performance than handcrafted ones in both setups. Furthermore, intermediate CNN features outperformed the ones derived from the fully connected layers in almost all cases. One may notice that there is a correlation between the dimensions of the descriptors and the performance of the model. Hence, due to the considerable performance difference, we focused our research on the exploration of the potential of intermediate CNN features.

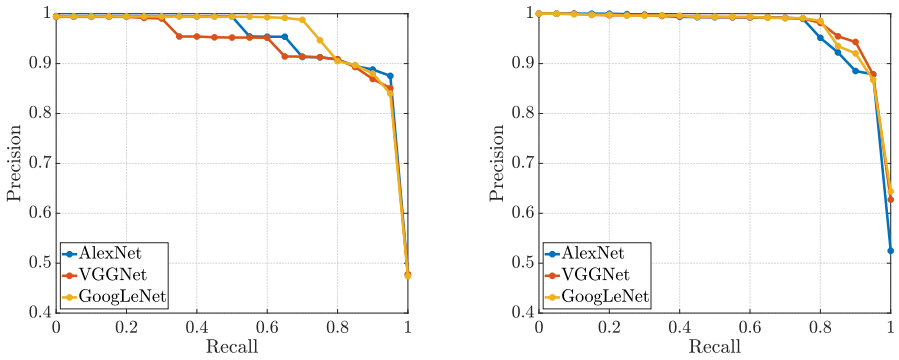
Impact of feature aggregation scheme

We study the performance of the proposed approach in the CC_WEB_VIDEO dataset in relation to the underlying CNN architecture and the size of the visual vocabulary. Regarding the first aspect, three CNN architectures are tested: AlexNet, VGGNet, and GoogLeNet, with both aggregation schemes implemented using $K = 1000$ words.

Figure 4.5 illustrates the PR curves of the different CNN architectures with the two aggregation schemes. Layer-based aggregation runs outperform vector-based ones for

Table 4.4: mAP per CNN architecture and aggregation scheme.

Method	K	AlexNet	VGGNet	GoogLeNet
Vector Aggregation	1000	0.951	0.937	**0.958**
	10,000	0.879	0.886	0.857
Layer Aggregation	1000	0.969	**0.976**	0.974
	10,000	0.948	0.959	0.958

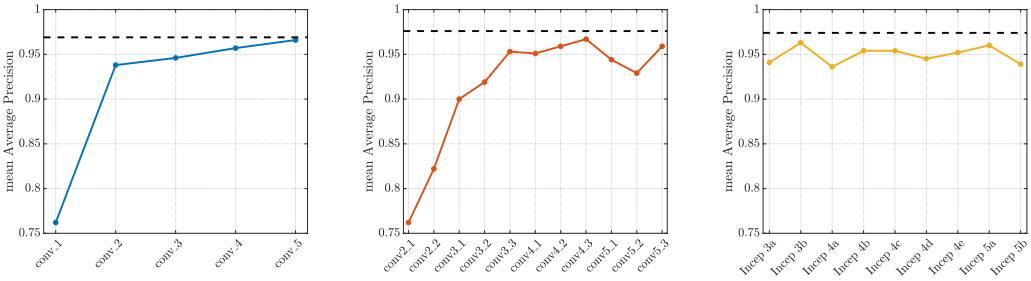


(a) Vector Aggregation (b) Layer Aggregation

Figure 4.5: PR curve of the proposed approach based on three CNN architectures and for the two aggregation schemes.

every architecture. GoogLeNet achieves the best results for the vector-based aggregation experiments with a precision close to 1.0 up to a 0.7 recall. For recall values in the range 0.8-1.0, all three architectures have similar results. For the layer-based aggregation scheme, all three architectures exhibit near-perfect performance up to 0.75 recall.

Similar conclusions are obtained from the analysis of mAP achieved using different CNN architectures, as depicted in Table 4.4. For the vector-based aggregation experiments, GoogLeNet achieved the best performance with a mAP of 0.958, and VGGNet the worst (mAP=0.937). On the other hand, when using the layer-based aggregation scheme, the best mAP score (0.976) was based on VGGNet. The lowest, yet competitive, results in the case of layer-based aggregation, are obtained for AlexNet (mAP=0.969).



(a) AlexNet (b) VGGNet (c) GoogLeNet

Figure 4.6: mAP of every layer for the three architectures.

To study the impact of vocabulary size, we compare the two schemes when used with $K = 1000$ and $K = 10,000$ (Table 4.4). Results reveal that the performance of vector-based aggregation for $K = 10,000$ is significantly lower compared to the case when $K = 1000$ words are used. It appears that the vector-based aggregation suffers considerably more from the increase of K compared to the layer-based aggregation, which appears to be less sensitive to this parameter. Due to this, we did not consider using the same amount of visual words for the vector-based and the layer-based aggregation, since the performance gap between the two types of aggregation with the same number of visual words would be much more pronounced.

Performance using individual layers

We also assessed the retrieval capability of every layer for the three tested CNN architectures. Figure 4.6 depicts the mAP of the approach using only a selected layer vector. In the AlexNet and VGGNet architectures, the mAP of the first layers is quite low, and as we are moving to deeper layers, the retrieval performance improves. In both cases, several layers exceed the performance of the vector-based aggregation scheme. This indicates that it is better to extract the feature descriptors only from one layer than concatenating all layers in a single vector, when using the BoW solution. However, no single layer overpasses the performance of the layer-based aggregation scheme displayed with a dashed line. In GoogLeNet, the first layer (Inception 3a) is already

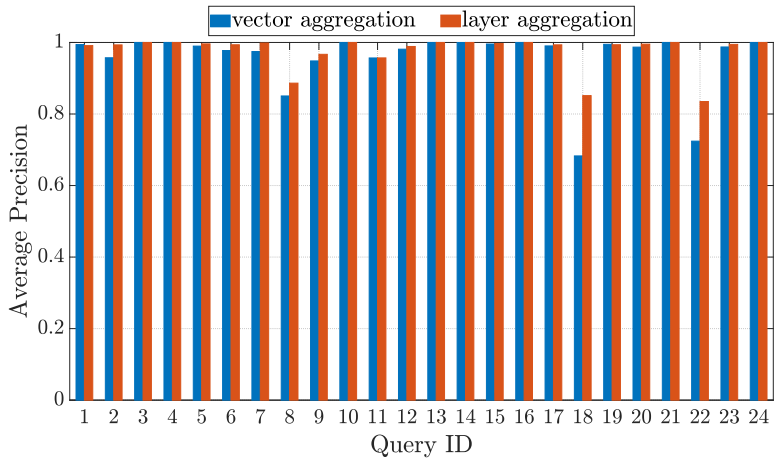


Figure 4.7: Average Precision per query for vector aggregation (GoogLeNet) and layer aggregation (VGGNet).

deep enough to achieve competitive performance. In this case, the performance for all layers fluctuates between 0.935 and 0.960.

Performance per query

Here, we analyse the performance of the best vector aggregation instance (GoogLeNet) with the best layer aggregation instance (VGGNet) on different queries. Figure 4.7 displays the Average Precision per query. Layer aggregation outperforms vector aggregation for every single query. However, both approaches fail in the difficult queries of the dataset, namely query 18 (`Bus uncle`) and query 22 (`Numa Gary`). The major factor leading to errors is that both videos have relatively low resolution/quality, and the candidate videos are heavily edited, which leads to a significant number of relevant videos not to be retrieved at all (i.e., many false negatives). Figure 4.8 illustrates some visual examples of the corresponding queries, their NDVs, and their rankings. Nevertheless, layer aggregation leads to considerably better results in both queries in comparison to vector aggregation.

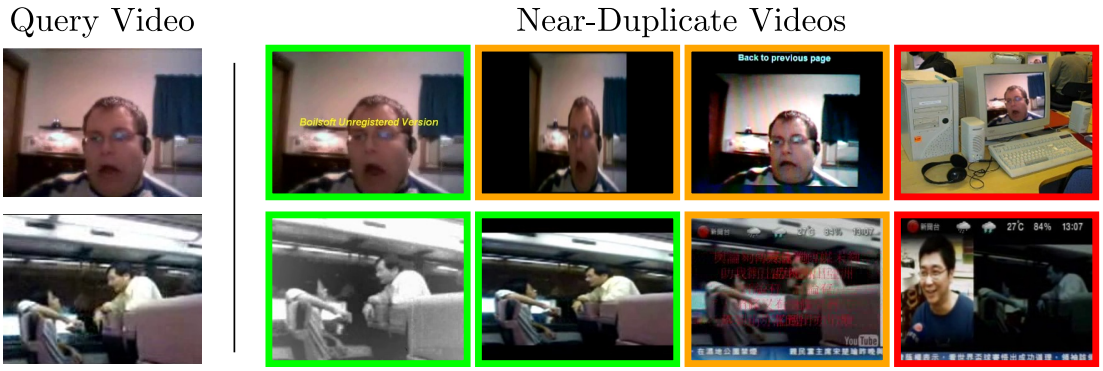


Figure 4.8: Visual examples of queries and their NDVs from CC_WEB_VIDEO. Colour indicates the rank of the NDVs based on LBoW: green corresponds to high ranks, orange corresponds to low ranks, and red indicates not retrieved at all.

Table 4.5: mAP of the baseline and two DML fusion schemes for the three benchmarked CNN architectures.

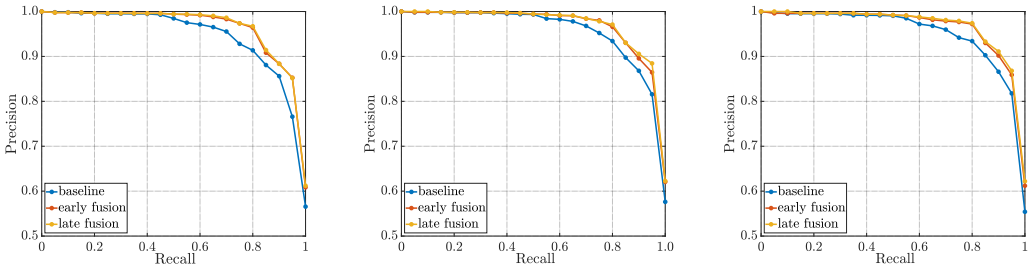
Architecture	**baseline**	**early fusion**	**late fusion**
AlexNet	0.948	0.964	0.964
VGGNet	0.956	0.970	**0.971**
GoogLeNet	0.952	0.968	0.969

4.4.3 Evaluation of DML approach

Impact of the different fusion schemes

In this section, we study the performance of the supervised DML approach in the evaluation dataset in relation to the underlying CNN architecture and the different fusion schemes. The three CNN architectures are benchmarked. For each of them, three configurations are tested: i) *baseline*: all frame descriptors are averaged to a single vector which is used for retrieval without any transformation, ii) *early fusion*: all frame descriptors are averaged to a single vector which is then transformed by applying the learned embedding function to generate the video descriptor, iii) *late fusion*: all frame descriptors are transformed by applying the learned embedding function and the generated embeddings are then averaged.

Figure 4.9 and Table 4.5 presents the PR curves and the mAP, respectively, of the three CNN architectures with the three fusion setups. Late fusion schemes consistently



(a) AlexNet (b) VGGNet (c) GoogLeNet

Figure 4.9: Precision-Recall curve of the proposed DML approach based on the two CNN architectures and for the three fusion setups.

Table 4.6: mAP of three feature extraction methods for the two CNN architectures based on the proposed DML apporach.

Architecture	**proposed**	**last conv**	**first fc**
AlexNet	0.964	0.957	0.962
VGGNet	**0.970**	0.965	0.964
GoogLeNet	0.968	0.960	0.961

outperform the other two fusion schemes for all CNN architectures. VGGNet achieves the best results for all three settings with a small margin compared to the GoogLeNet, with precision more than 0.97 up to 0.80 recall and mAP scores of 0.970 and 0.971 for early and late fusion respectively. Performance clearly increases in both trained fusion schemes compared to the baseline for all three architectures. The early and late fusion schemes achieve almost identical results, which is an indication that the choice of the fusion scheme is not critical.

Comparison of different features

To delve deeper into performance, we validate the performance of the DML framework with early fusion built on features extracted based on three different methods. The benchmarked methods are: i) **proposed**: apply max-pooling to all convolution layers and concatenate the vectors, ii) **last conv**: apply max-pooling to the activations of the last convolution layer, iii) **first fc**: the activations of the first fully-connected layer. We experiment with the three CNN architectures.

Table 4.7: mAP comparison between the two proposed approaches against five state-of-the-art methods. The approaches are divided based on the dataset used for development.

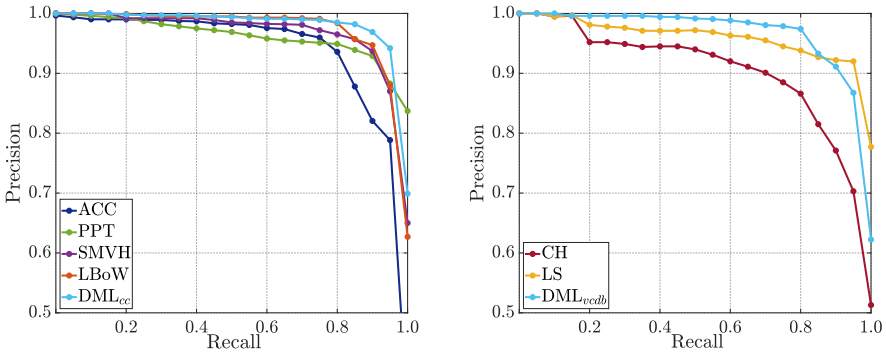
	Same Dataset					**No/Other Dataset**		
Method	**ACC**	**PPT**	**SMVH**	**LBoW**	**DML$_{cc}$**	**CH**	**LS**	**DML$_{vcdb}$**
mAP	0.944	0.958	0.971	0.976	**0.982**	0.892	0.954	**0.971**

Table 4.6 depicts the mAP of the three feature extraction methods for the three CNN architectures. The proposed feature extraction scheme outperforms the runs of the compared feature extraction methods, for all architectures. In case of AlexNet, the **proposed** method marginally outperforms the **first fc** method. However, our approach reports better performance compared to the others when VGGNet or GoogLeNet is used. Hence, we may draw the conclusion that the feature extraction using all convolution layers yields better results, when using the DML solution.

Comparison against state-of-the-art approaches

For comparing the performance of the two approaches with the five approaches from the literature, we select the setup using VGGNet features with layer aggregation for the BoW approach, denoted as LBoW, and the setup using VGGNet features with late fusion for the DML approach, denoted as DML$_{vcdb}$ since they achieved the best results in each case. We separate the compared approaches into two groups based on the developed dataset, i.e., whether the evaluation dataset is used for development or not. For the sake of comparison and completeness, the results of the DML method trained on a triplet set derived from both VCDB (similar to DML$_{vcdb}$) and a small sample of 1K triplets from CC_WEB_VIDEO are denoted as DML$_{cc}$. This simulates the situation where the DML-based approach has access to a portion of the evaluation corpus, similar to the setting used by the competing approaches.

In Table 4.7, the mAP scores of the competing methods are reported. The DML approach outperforms all methods in each group with a clear margin. A similar conclusion is reached by comparing the PR curves illustrated in Fig. 4.10, with the light



(a) Evaluation dataset (b) No/Other dataset

Figure 4.10: Precision-Recall curve comparison between the two proposed approaches against five state-of-the-art methods. The approaches are divided based on the dataset used for development.

blue line (DML approach) lying upon all others up to 0.90 recall in both cases. The DML approach trained on the VCDB dataset outperforms four out of five state-of-the-art methods. It achieves similar results to the SMVH, even though the latter has been developed with access to the evaluation dataset during training. The LBoW approach is in the second place consistently outperforming all five competing approaches by a considerable margin.

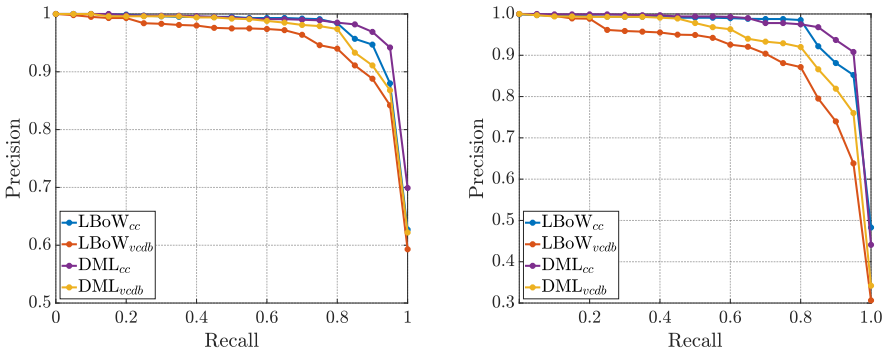
4.4.4 In-depth comparison of the two approaches

Experiments on CC_WEB_VIDEO

In this section, we compare the two implemented approaches in two evaluation settings. In addition to the existing experiments, we implement the BoW approach with VGGNet features and layer aggregation based on information derived from the VCDB dataset, i.e., we build the layer codebooks from a set of video frames sampled from the aforementioned dataset. We then test two variations, the LBoW$_{cc}$ that was developed on the CC_WEB_VIDEO dataset (same as Section 4.4.2) and the LBoW$_{vcdb}$ developed on the VCDB dataset. For each of the 24 queries of CC_WEB_VIDEO, only the videos contained in its subset (the dataset is organized in 24 subsets, one per

Table 4.8: mAP comparison of the two developed approaches on two different dataset setups.

Run	CC_WEB_VIDEO	CC_WEB_VIDEO*
LBoW$_{vcdb}$	0.957	0.906
DML$_{vcdb}$	**0.971**	**0.941**
LBoW$_{cc}$	0.976	0.960
DML$_{cc}$	**0.982**	**0.969**



(a) CC_WEB_VIDEO (b) CC_WEB_VIDEO*

Figure 4.11: Precision-Recall curve comparison of the two developed approaches on two dataset setups.

query) are considered as candidates and used for the calculation of retrieval performance. To emulate a more challenging setting, we created CC_WEB_VIDEO* in the following way: for every query in CC_WEB_VIDEO, the set of candidate videos is the entire dataset instead of only the query subset.

Figure 4.11 depicts the PR curves of the four runs and the two setups. There is a clear difference between the performance of the two variants of the LBoW approach, for both dataset setups. The DML approach outperforms the LBoW approach for all runs and setups at any recall point by a large margin. Similar conclusions can be drawn from the mAP scores of Table 4.8. The performance of LBoW drops by more than 0.02 and 0.062 when the codebook is learned on VCDB for each setup, respectively. Again, there is a considerable drop in performance in CC_WEB_VIDEO* setup for both approaches, with the DML being more resilient to the setup change.

Table 4.9: mAP of the two developed approaches on the FIVR-200K dataset.

Task	DSVR	CSVR	ISVR
LBoW	**0.710**	**0.675**	**0.572**
DML	0.398	0.378	0.309

As a result, DML has been demonstrated to be highly competitive and possible to transfer to different datasets with relatively lower performance loss.

Experiments on FIVR-200K

The developed approaches are also benchmarked on our FIVR-200K dataset. To compare the two methods, we implemented them with frame features derived from the VGGNet. The LBoW was built with samples from the FIVR-200K dataset and DML with triplets from the VCDB dataset. Table 4.9 present the mAP of the two developed approaches on the FIVR-200K dataset. The LBoW approach achieves noticeably better performance in comparison to the DML. This is expected as LBoW has been developed with samples from the evaluation dataset. Even though it is an unsupervised method, it greatly benefits from such settings so as to build more representative codebooks. On the other hand, DML has been trained with the VCDB dataset that does not adequately simulate the FIVR problem. More precisely, for the DSVR task, the two methods achieve 0.710 and 0.398 mAP for LBoW and DML, respectively. The performance of both approaches marginally drops on the CSVR task, compared to DSVR, with a reduction of more than 0.02 mAP. On the ISVR task, both runs also have a considerable drop in their performance, with 0.572 and 0.309 mAP for LBoW and DML, respectively. It is noteworthy that both methods' performance is significantly reduced compared to the CC_WEB_VIDEO dataset, revealing that the FIVR-200K dataset is much more challenging. The main reason is that the vast majority of positive video pairs are partially related, i.e., the videos are not related in their entirety but in particular segments. The competing approaches from the literature lead to even lower performance since they are based on schemes that employ

Table 4.10: mAP of the two developed approaches on the within-dataset split of FIVR-200K dataset.

Task	DSVR	CSVR	ISVR
LBoW	0.362	0.344	0.280
DML	**0.443**	**0.420**	**0.365**

handcrafted frame descriptors with limited representation capability.

Besides, to test the robustness of the proposed methods, we benchmark them in the scenario where they are developed and evaluated based on different video corpora derived from the same domain (i.e., FIVR). To do so, we evaluate the two approaches based on the within-dataset split of FIVR-200K, similar to the last experiment of Section 3.3.2. For this experiment, the dataset is split into two parts, i.e., training and test split. Table 4.10 depicts the results of the two approaches developed with the training split and evaluated on the test split. The results highlight that the DML approach achieves considerably better performance compared to the LBoW when they are both developed on a different dataset other than the evaluation. Comparing these results with the ones presented in Table 4.9, it is evident that the performance of LBoW drops by half when the codebooks are learned on a video corpus other than the one used for evaluation. We have also experimented with external resources for building the codebooks, i.e., VCDB or ImageNet, and with various vocabulary sizes. Nevertheless, no improvement in terms of retrieval performance was achieved. Furthermore, as expected, the DML significantly benefits from the training on a dataset from the same domain, i.e., its performance considerably improves when the network is trained with triplets from the within-dataset split. The mAP increases in all evaluation tasks by more than 0.04, with the case of ISVR being the most notable one with an 18% relative mAP increase. This highlights that DML provides the required flexibility with respect to the definitions of related videos, which is necessary for the FIVR. In conclusion, the DML method generalizes better on unseen data than the LBoW.

Both presented approaches are limited in similar ways, which leads to similar errors

in the retrieval process. The major issue of both approaches is that they do not function effectively when the related segment between the two videos is small relative to their total size. As revealed from the evaluation in the FIVR-200K dataset, video-level solutions suffer in such setups. Even the LBoW approach, where the video-level representation contains frame-level information, fails to retrieve relevant videos, especially when built on a different dataset than the evaluation. Another category of videos that the proposed schemes fail is when heavy transformations have been applied on the source video. Typically, the extracted frame descriptors are not close enough for these videos to be retrieved and ranked with a high similarity score. Even the DML scheme, which should learn to handle such cases, fails to assign high similarity scores to this kind of duplicate pairs, mainly when heavy edits or overlays have been applied. A solution to this issue is the use of frame descriptors that better capture local information within frames. This could perhaps be achieved with the use of another aggregation function (other than MAC) that better preserves local information or with the application of augmentation schemes that will result in more robust models.

Computational time

Finally, we compare the two approaches in terms of processing time on the FIVR-200K dataset. The results have been measured using the open-source library Scikit-learn [126] in Python on a Linux PC with a 4-core i7-4770K and 32GB of RAM. The DML approach is significantly faster than the LBoW approach with respect to retrieval time. It needs 333 ms to perform retrieval for one query on the FIVR-200K dataset, compared to 1,155 ms needed for the LBoW approach. However, both methods are significantly faster than common frame-level approaches, which usually need several minutes to process all videos in the dataset. Moreover, DML needs approximately four hours for the training of the DNN on VCDB and the extraction of the video embeddings. However, LBoW needs about two days for the codebook learning with samples from FIVR-200K and the generation of the inverted file structure. Therefore,

DML is much more practical, especially in scenarios where the video database is not static, i.e., new videos are constantly added, and the retraining of the video retrieval scheme is required.

4.5 Conclusions

In this chapter, we proposed two different video-level approaches (an unsupervised and a supervised) based on deep neural networks, that were initially introduced for the problem of NDVR, which is closely related to the FIVR problem. For both methods, we used CNN features extracted from the intermediate convolutional layers by applying Maximum Activations of Convolutions (MAC). We found that this setup led to the best results, among many other features, both handcrafted and learned. The first approach is an unsupervised scheme that relies on a Bag-of-Word (BoW) video representation. A layer-based aggregation scheme was introduced in order to generate the global video representation, and then store it in an inverted file index for fast indexing and retrieval. To quantify video similarity, we calculated the cosine similarity on *tf-idf* weighted versions of the extracted vectors and ranked the results in descending order. However, we found that the BoW approach has several limitations, with the most important being that it offers a dataset-specific solution, i.e., the extracted knowledge is not transferable, and re-building the model is computationally expensive. To address these issues, we developed a supervised approach based on DML. This method learns an embedding function that maps the input video descriptors to a feature space where related videos are closer than the irrelevant ones. The similarity between videos was assessed by their Euclidian distance in the embedding space. We conducted extensive evaluations with different experimental setups, testing the performance of the developed approaches under various settings. The developed approaches exceed the performance of existing state-of-the-art approaches. Finally, we empirically determined that the DML approach achieves significantly better performance than the BoW approach when they are both developed with no access to the evaluation dataset.

Chapter 5

Video similarity learning based on frame-level information

Contents

In this chapter, we introduce ViSiL, a Video Similarity Learning architecture that considers fine-grained spatio-temporal relations between pairs of videos – such relations are typically lost in previous video retrieval approaches that embed the whole frame or even the whole video into a vector descriptor before the similarity estimation. By contrast, our Convolutional Neural Network (CNN)-based approach is trained to calculate video-to-video similarity from refined frame-to-frame similarity matrices, so as to consider both intra- and inter-frame relations. In the proposed method, pairwise frame similarity is estimated by applying Tensor Dot (TD) followed by Chamfer Similarity (CS) on regional CNN frame features – this avoids feature aggregation before the similarity calculation between frames. Subsequently, the similarity matrix between all video frames is fed to a four-layer CNN, and then summarized using Chamfer Similarity (CS) into a video-to-video similarity score – this avoids feature aggregation

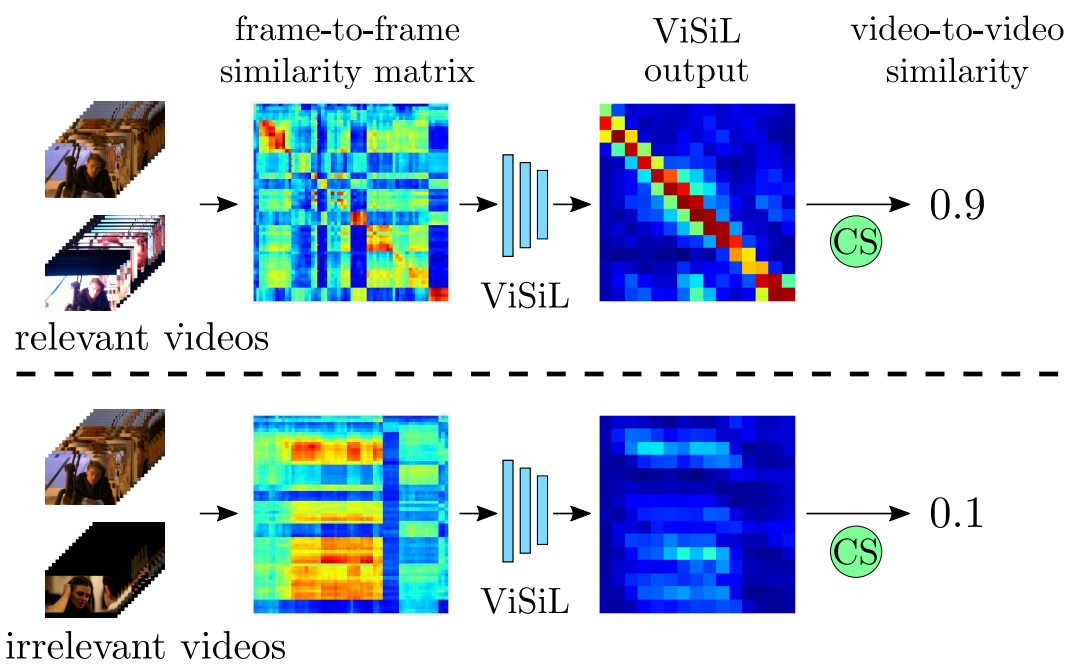


Figure 5.1: Depiction of the frame-to-frame similarity matrix and the CNN output of the ViSiL approach for two video pair examples: relevant videos that contain footage from the same incident (top), unrelated videos with spurious visual similarities (bottom). CS stands for Chamfer Similarity.

before the similarity calculation between videos and captures the temporal similarity patterns between matching frame sequences. We train the proposed network using a triplet loss scheme and evaluate it on six public benchmark datasets on four different video retrieval problems where we demonstrate large improvements in comparison to the state-of-the-art. The implementation of ViSiL is publicly available[1].

Base on our literature review in Section 2.2, a promising direction is exploiting better the spatial and temporal structure of videos in the similarity calculation [25, 51, 52]. However, recent approaches either focused on the spatial processing of frames and completely disregarded temporal information [30, 66], or considered global frame representations (essentially discarding spatial information) and then considered the temporal alignment among such frame representations [22, 11]. In this chapter, we propose ViSiL, a video similarity learning network that considers both the spatial

[1]https://github.com/MKLab-ITI/visil

(intra-frame) and temporal (inter-frame) structure of the visual similarity. The main contributions of this chapter are:

- We introduce a frame-to-frame similarity function that employs Tensor Dot (TD) product and Chamfer Similarity (CS) on *region-level* frame Convolutional Neural Network (CNN) features whitened with PCA and weighted with an attention mechanism. This leads to a frame-to-frame similarity function that takes into consideration region-to-region pairwise similarities, instead of calculating the similarity of frame-level embeddings where the regional details are lost.
- We propose a novel video similarity learning architecture for fine-grained video-to-video similarity calculation. We calculate the matrix with the similarity scores between each pair of frames between the two videos and use it as input to a four-layer CNN, which is followed by a Chamfer Similarity (i.e., a mean-max filter) at its final layer. By doing so, we learn the temporal structure of the frame-level similarity of relevant videos, such as the presence of diagonal structures in Figure 5.1, and suppress spurious pairwise frame similarities that might occur.
- We develop a pipeline to train the proposed network, which generates triplets of videos from two pools of selected and artificially-generated duplicate video pairs. Our goal is the network to assign higher similarity scores for relevant videos and lower for irrelevant ones; hence, it is trained to optimize the triplet loss scheme. In addition, we introduce a similarity regularization loss that penalizes the saturated values generated by the network, which demonstrates a significant performance boost.

We evaluate our method on several video retrieval problems using public benchmark datasets. We benchmark ViSiL for FIVR, the thesis's main problem, and two other content-based problems, i.e., NDVR, EVR. Besides, we test ViSiL's performance on Action Video Retrieval (AVR), whose objective is the retrieval of videos that depicts

the same action. Even though it belongs to a different line of research, our method can successfully tackle this problem with the proper modifications in the network's architecture. In all cases, the proposed method outperforms the state-of-the-art and often by a large margin.

The remainder of the chapter is organized as follows: Section 5.1 introduces the proposed ViSiL approach by presenting the features extraction process, the proposed frame-to-frame and video-to-video similarity calculation functions, and the pipeline for the training of the network. Section 5.2 reports our experiments, results, and comparisons, and finally, Section 5.3 summarizes our main conclusions.

5.1 Fine-grained spatio-temporal video similarity learning

In this section, we first provide a brief presentation of two underlying functions used for the similarity calculation of two compared videos (Section 5.1.1). Then, we describe in detail the ViSiL method and all of the individual components used to build the proposed method (Section 5.1.2). We conclude this section with the presentation of the training process followed to train the proposed network (Section 5.1.3).

5.1.1 Preliminaries

Tensor Dot (**TD**): Having two tensors $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times K}$ and $\mathcal{B} \in \mathbb{R}^{K \times M_1 \times M_2}$, their TD (also known as tensor contraction) is given by summing the two tensors over specific axes. Following the notation in [137], TD of two tensors is

$$\mathcal{C} = \mathcal{A} \cdot_{(i,j)} \mathcal{B} \tag{5.1}$$

where $\mathcal{C} \in \mathbb{R}^{N_1 \times N_2 \times M_1 \times M_2}$ is the TD of the tensors, and i and j indicate the axes over which the tensors are summed. In the given example i and j can only be 3 and 1 respectively, since they are the only ones of the same size (K).

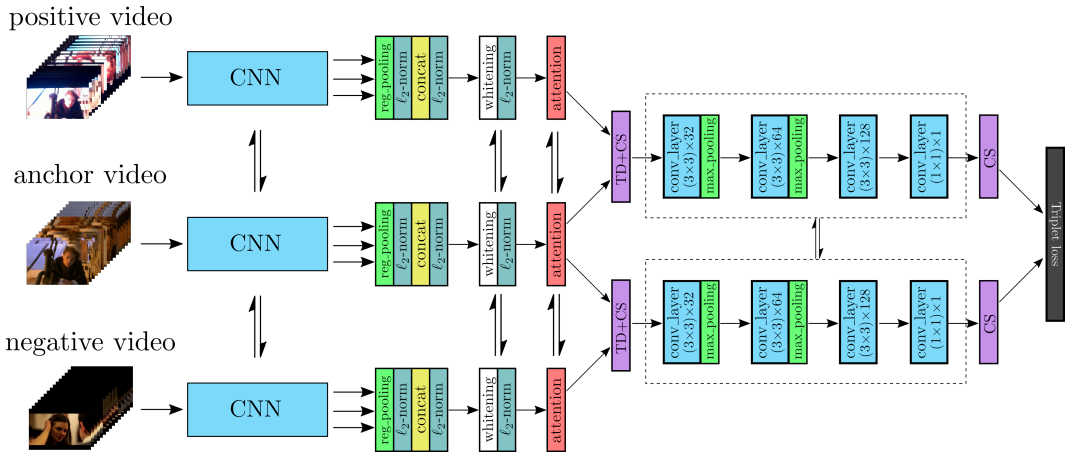


Figure 5.2: Overview of the training scheme of the proposed architecture. A triplet of an anchor, positive and negative videos, is provided to a CNN to extract regional features that are PCA-whitened and weighted based on an attention mechanism. Then the Tensor Dot product is calculated for the anchor-positive and anchor-negative pairs followed by Chamfer Similarity to generate frame-to-frame similarity matrices. The output matrices are passed to a CNN to capture temporal relations between videos and calculate video-to-video similarity by applying Chamfer Similarity on the output. The network is trained with the triplet loss function. The double arrows indicate shared weights.

Chamfer Similarity (**CS**): This is the similarity counterpart of Chamfer Distance [12]. Considering two sets of items x and y with total number of N and M items respectively and their similarity matrix $\mathcal{S} \in \mathbb{R}^{N \times M}$, CS is calculated as the average similarity of the most similar item in set y for each item in set x. This is formulated in Equation 5.2.

$$\mathrm{CS}(x, y) = \frac{1}{N} \sum_{i=1}^{N} \max_{j \in [1,M]} \mathcal{S}(i, j) \tag{5.2}$$

Note that CS is not symmetric, i.e. $\mathrm{CS}(x, y) \neq \mathrm{CS}(y, x)$, however, a symmetric variant Symmetric Chamfer Similarity (SCS) can be defined as,

$$\mathrm{SCS}(x, y) = (\mathrm{CS}(x, y) + \mathrm{CS}(y, x))/2 \tag{5.3}$$

5.1.2 ViSiL description

Figure 5.2 illustrates the proposed approach. We first extract features from the intermediate convolution layers of a CNN architecture by applying region pooling on

the feature maps. These are further PCA-whitened and weighted based on an attention mechanism. Additionally, a similarity function based on TD and CS is devised to accurately compute the similarity between frames. A similarity matrix comprising all pairwise frame similarities is then fed to a CNN to train a video-level similarity model. This is trained with a triplet loss scheme based on selected and automatically generated triplets from a training dataset.

Feature extraction

Given an input video frame, we apply Regional Maximum Activation of Convolution (R-MAC) [124] on the activations of the intermediate convolutional layers [65] given a specific granularity level $L_N, N \in \{1, 2, 3, ...\}$. Given a CNN architecture with a total number of K convolutional layers, this process generates K feature maps $\mathcal{M}^k \in \mathbb{R}^{N \times N \times C_k} (k = 1, ..., K)$, where C_k is the number of channels of the k^{th} convolution layer. All extracted feature maps have the same resolution ($N \times N$) and are concatenated into a frame representation $\mathcal{M} \in \mathbb{R}^{N \times N \times C}$, where $C = C_1 + ... + C_K$. We also apply ℓ^2-normalization on the channel axis of the feature maps, before and after concatenation. This feature extraction process is denoted as L_N-iMAC. The extracted frame features retain the spatial information of frames at different granularities. We then employ Principal Components Analysis (PCA) on the extracted frame descriptors to perform whitening and/or dimensionality reduction as in [44]. This process consists of a vector shifting and projection, which can be implemented with a fully-connected layer, namely whitening layer. By the end of this process, each video frame is represented by a tensor $\mathcal{M}$ with region vector $\mathbf{r}_{ij} : \mathcal{M}(i, j, \cdot) \in \mathbb{R}^C$, where $i \in [1, N], j \in [1, N]$.

ℓ^2-normalization on the extracted frame descriptors result in all region vectors being equally considered in the similarity calculation. For example, this would mean that a completely dark region would have the same impact on similarity with a region depicting a subject of interest. To avoid this, we weight the frame regions based on

Figure 5.3: Examples of the attention weighting on arbitrary video frames: sampled video frames from the same video (top), attention maps of the corresponding frames (bottom). Red colour indicates high attention weights, whereas blue indicates low ones.

their saliency via a visual attention mechanism over region vectors inspired by methods from different research fields, i.e. document classification [139]. To successfully adapt it to the needs of video retrieval, we build the following attention mechanism: given a frame with region vector $\mathbf{r}_{ij} \in \mathbb{R}^C$, we introduce a visual context unit vector $\mathbf{u}$ and use it to measure the importance of each region vector. We calculate the dot product between every region vector $\mathbf{r}_{ij}$ with the context vector $\mathbf{u}$ to derive the weight scores α_{ij}. Since all vectors are unit norm, α_{ij} will be in the range $[-1, 1]$. To retain region vectors' direction and change their norm, we divide the weight scores α_{ij} by 2 and add 0.5 in order to be in range $[0, 1]$. Equation 5.4 formulates the weighting process.

$$\begin{aligned} \alpha_{ij} &= \mathbf{u}^\top \mathbf{r}_{ij}, \quad s.t. \, \|\mathbf{u}\| = 1 \\ \mathbf{r}'_{ij} &= (\alpha_{ij}/2 + 0.5)\mathbf{r}_{ij} \end{aligned} \tag{5.4}$$

All functions in the weighting process are differentiable; therefore, $\mathbf{u}$ is learned through the training process. Unlike the common practice in the literature, we do not apply any normalization function on the calculated weights (e.g., softmax or division by sum) because we want to weight each vector independently. Also, we empirically found that, unlike other works, using a hidden layer in the attention module has a negative effect on the system's performance.

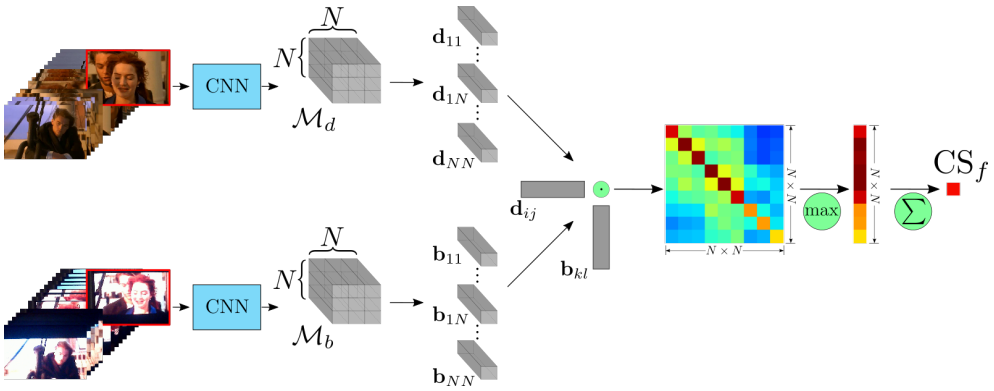


Figure 5.4: Illustration of frame-level similarity calculation between two video frames. Having extracted the region-level frame descriptor based on a CNN network, the regional feature maps are decomposed into their individual region vectors. Then, the dot product between every pair of region vectors is calculated to generate a region-to-region similarity matrix. To compute the frame-to-frame similarity, we apply the CS function on the generated similarity matrix. In this example, the frames are near duplicates.

Figure 5.3 illustrates three visual examples of video frames coloured based on the attention weights of their region vectors. Apparently, the proposed attention mechanism weights the frame regions independently based on their saliency, i.e., the amount of information depicted in corresponding areas of the video frames. It assigns high weight values on the information-rich regions (e.g., the concert stage, the Mandalay Bay building); whereas, it assigns low values on regions that contain no meaningful object (e.g., solid dark regions).

Frame-to-frame similarity

Figure 5.4 illustrates the similarity calculation process between two near-duplicate frames. Given two video frames d, b, we apply CS on their region feature maps to calculate their similarity. First, the regional feature maps $\mathcal{M}_d, \mathcal{M}_b \in \mathbb{R}^{N \times N \times C}$ are decomposed into their region vectors $\mathbf{d}_{ij}, \mathbf{b}_{kl} \in \mathbb{R}^C$. Then, the dot product between every pair of region vectors is calculated, creating the similarity matrix of the two frames, and CS is applied on the similarity matrix to compute the final frame-to-frame

Table 5.1: Architecture of the proposed network for video similarity learning. For the calculation of the output size, we assume that two videos with total number of X and Y frames are provided.

Type	**Kernel size / stride**	**Output size**	**Activ.**
Conv	3×3 / 1	$X \times Y \times 32$	ReLU
M-Pool	2×2 / 2	$X/2 \times Y/2 \times 32$	—
Conv	3×3 / 1	$X/2 \times Y/2 \times 64$	ReLU
M-Pool	2×2 / 2	$X/4 \times Y/4 \times 64$	—
Conv	3×3 / 1	$X/4 \times Y/4 \times 128$	ReLU
Conv	1×1 / 1	$X/4 \times Y/4 \times 1$	—

similarity of the frame pair.

$$\mathrm{CS}_f(d,b) = \frac{1}{N^2} \sum_{i,j=1}^{N} \max_{k,l \in [1,N]} \mathbf{d}_{ij}^{\top} \mathbf{b}_{kl} \tag{5.5}$$

This process leverages the geometric information captured by region vectors and provides some degree of spatial invariance. More specifically, the CNN extracts features that correspond to mid-level visual structures, such as object parts, and combined with CS, that by design disregards the global structure of the region-to-region matrix, constitutes a robust similarity calculation process against spatial transformations, e.g., spatial shift. This presents a trade-off between the preservation of the frame structure and invariance to spatial transformations.

Video-to-video similarity

To apply frame-to-frame similarity on two videos q, p with X and Y frames respectively, we apply TD combined with CS on the corresponding video tensors $\mathcal{Q}$ and $\mathcal{P}$ and derive the frame-to-frame similarity matrix $\mathcal{S}_f^{qp} \in \mathbb{R}^{X \times Y}$. This is formulated in Equation 5.6.

$$\mathcal{S}_f^{qp} = \frac{1}{N^2} \sum_{i=1}^{N^2} \max_{j \in [1,N^2]} \mathcal{Q} \cdot_{(3,1)} \mathcal{P}^{\top}(\cdot, i, j, \cdot) \tag{5.6}$$

where the TD axes indicate the channel dimension of the corresponding video tensors. In that way, we apply Equation 5.5 on every frame pair.

To calculate the similarity between two videos, the generated similarity matrix $\mathcal{S}_f^{qp}$ derived from the previous process is provided to a CNN network. The network is capable of learning robust patterns of within-video similarities at segment level. Table 5.1 displays the architecture of the CNN architecture of the proposed ViSiL framework. It consists of four convolutional layers and two max-pooling layers. The first three convolutional layers have the same kernel size (3×3), with a gradually incremented number of filters (32, 64, and 128, respectively). The ReLU activation function is applied to the output of the first three convolutional layers. After the first two convolutional layers, we apply max-pooling with kernel size (2×2) and stride 2; hence, the similarity matrix is analysed in coarser granularity. The final convolutional layer aggregates the activations of the third one to generate the output of the network.

To calculate the final video similarity, we apply the *hard tanh* activation function on the values of the network output, which clips values within range $[-1, 1]$. We use this activation function so as to bound the similarity in a specific range. Without the application of hard tanh, the network converges to a similarity space with arbitrary boundaries, which is impractical for ranking videos. Then, we apply CS to derive a single value as in Equation 5.7.

$$\mathrm{CS}_v(q,p) = \frac{1}{X'} \sum_{i=1}^{X'} \max_{j \in [1,Y']} \mathrm{Htanh}(\mathcal{S}_v^{qp}(i,j)) \tag{5.7}$$

where $\mathcal{S}_v^{qp} \in \mathbb{R}^{X' \times Y'}$ is the output of the CNN network, and Htanh indicates the element-wise hard tanh function. The output of the network has to be bounded in order to accordingly set the margin in Equation 5.8.

Similar to the frame-to-frame similarity calculation, this process is a trade-off between respecting video-level structure and being invariant to some temporal differences. As a result, different temporal similarity structures in the frame-to-frame similarity matrix can be captured, e.g., strong diagonals or diagonal parts (i.e., contained sequences). Also, the network learns to filter the noise introduced in the frame-to-frame similarity calculation, i.e., a pair of frames has high similarity value but no temporal pattern can

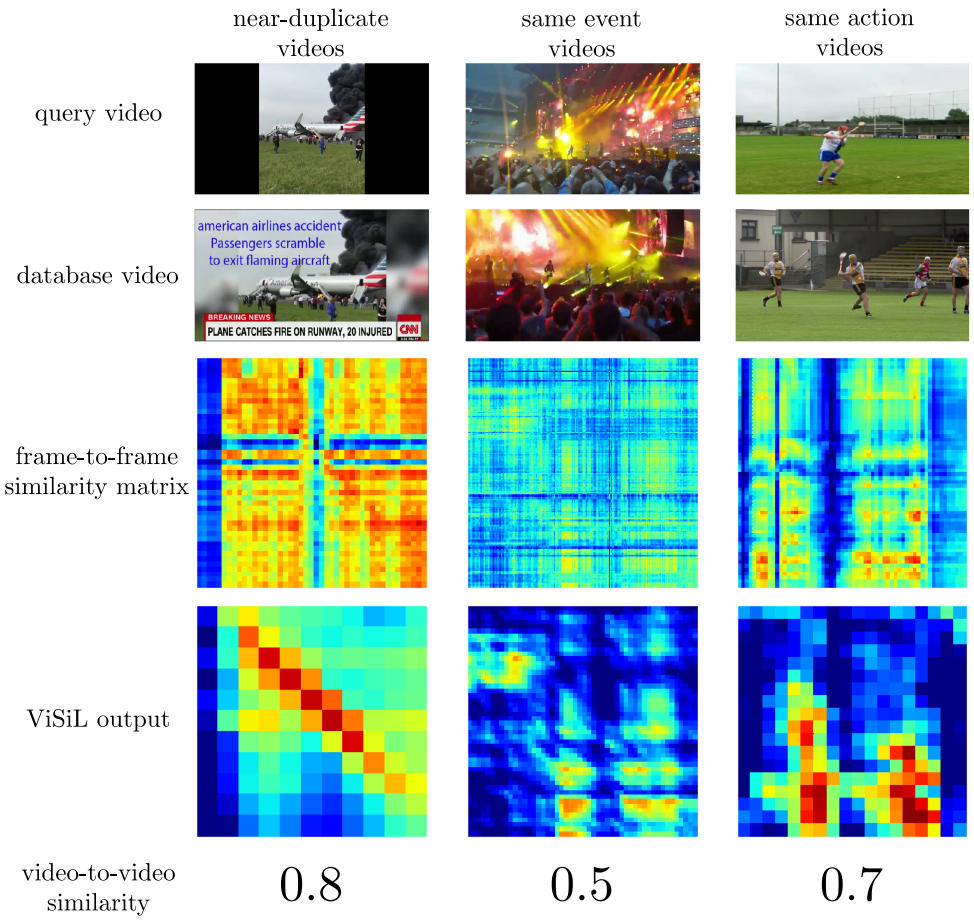


Figure 5.5: Visual examples of the input and output of ViSiL for three different video relation types. Two sampled frames of the compared videos are depicted on top, then the input frame-to-frame similarity matrix and the ViSiL output are displayed, and the final video-to-video similarity is reported. In the similarity matrices, red colour indicates a high similarity score, whereas blue indicates low similarity.

be captured. Hence, this process leads to the precise calculation of video similarity by considering the temporal alignment of the videos.

Figure 5.5 illustrates examples of the input frame-to-frame similarity matrix, the network output, and the calculated video similarity of two compared videos for three video categories. The network can extract temporal patterns from the input frame-to-frame similarity matrices (e.g., strong diagonals, consistent parts with high similarity) and suppress the noisy (i.e., small inconsistent parts with high similarity values) to calculate the final video-to-video similarity precisely. It also provides invariance to the

video starting points, assigning high similarity scores to videos that are not perfectly aligned (as in the near-duplicate example). Sampled frames from the compared videos are depicted for a better understanding of the different video relation types.

5.1.3 Training ViSiL

Loss function

The target video similarity score $\mathrm{CS}_v(q, p)$ should be higher for relevant videos and lower for irrelevant ones. To train our network we organize our video collection in video triplets (v, v^+, v^-), where v, v^+, v^- stand for an anchor, a positive (i.e. relevant), and a negative (i.e. irrelevant) video respectively. To force the network to assign higher similarity scores to positive video pairs and lower to negative ones, we use the 'triplet loss', that is

$$\mathcal{L}_{tr} = \max\{0, \mathrm{CS}_v(v, v^-) - \mathrm{CS}_v(v, v^+) + \gamma\} \tag{5.8}$$

where γ is a margin parameter.

In addition, we define a similarity regularization function that penalizes high values in the input of hard tanh that would lead to saturated outputs. This is an effective mechanism to drive the network to generate output matrices $\mathcal{S}_v$ with values in the range $[-1, 1]$, which is the clipping range of hard tanh. To calculate the regularization loss, we simply sum all values in the output similarity matrices that fall outside the clipping range (Equation 5.9).

$$\mathcal{L}_{reg} = \sum_{i=1}^{X'} \sum_{j=1}^{Y'} |\max\{0, \mathcal{S}_v^{qp}(i, j) - 1\}| + |\min\{0, \mathcal{S}_v^{qp}(i, j) + 1\}| \tag{5.9}$$

Finally, the total loss function is given in Equation 5.10.

$$\mathcal{L} = \mathcal{L}_{tr} + r * \mathcal{L}_{reg} \tag{5.10}$$

where r is a regularization hyperparameter that tunes the contribution of the similarity regularization to the total loss.

Training process

Training the ViSiL architecture requires a labeled training dataset. According to the ground truth annotations, we first extract video pairs with related visual content to serve as anchor-positive pairs during training. We also generate artificial anchor-positive video pairs by applying a number of transformations on arbitrary videos. We use the original video as the anchor and the generated as the positive. We consider three categories of transformation: (i) *colour*, including conversion to grayscale, brightness, contrast, hue, and saturation adjustment, (ii) *geometric*, including horizontal or vertical flip, crop, rotation, resize and rescale, and (iii) *temporal*, including slow motion, fast forward, frame insertion, video pause, random time shifts, or reversion. During training, one transformation from each category is randomly selected and applied on the selected video.

We construct two video pools that consist of anchor-positive pairs. For each pair, we aim to generate *hard triplets*, i.e., mine for negative videos (hard negatives) with similarity to the anchor that is greater than the one between the anchor and positive videos. In what follows, we use the LBoW approach presented in Section 4.2 to calculate similarities between videos.

The first pool derives from the annotated videos in the training dataset. Two videos with at least five-second overlap constitute an anchor-positive pair. Let s be the similarity of the corresponding videos. Videos with a similarity larger than s (measured with LBoW) with either of the videos in the anchor-positive pair are considered hard negatives. The second pool derives from arbitrary videos from the training dataset used to artificially generate positive pairs. Videos that are similar to the initial videos (similarity > 0.1) are considered hard negatives. To avoid potential near-duplicates, we exclude videos with similarity > 0.5 from the hard negative sets.

At each training epoch, we sample T triplets from each video pool. Due to GPU

memory limitations, we do not feed the entire videos to the network. Instead, we select a random video snippet with a total size of W frames from each video in the triplet. To ensure that there are at least five seconds overlap between the anchor and the positive videos, we use a training dataset that contains segment-level annotations. Also, it is noteworthy that we only train our networks once and then apply it on each content-based video retrieval problem, i.e., FIVR, NDVR, EVR.

5.2 Experimental study

In this section, we first present the evaluation setup by introducing the benchmark datasets and the implementation setting for four retrieval problems (Section 5.2.1). Then, we compare the proposed frame-to-frame similarity calculation scheme with several global features with dot product as similarity measure (Section 5.2.2). We also provide an ablation study to evaluate the proposed approach under different configurations (Section 5.2.3). Finally, we compare the "full" proposed approach (denoted as ViSiL$_v$) with the best-performing methods in the state-of-the-art (to the best of our knowledge) in each problem (Section 5.2.4).

5.2.1 Evaluation setup

The proposed approach is evaluated on four retrieval tasks, namely Near-Duplicate Video Retrieval (NDVR), Fine-grained Incident Video Retrieval (FIVR), Event-based Video Retrieval (EVR), and Action Video Retrieval (AVR). We assess its performance on six evaluation datasets and compare against several state-of-the-art methods. In all cases, we report the mean Average Precision (mAP).

Datasets

The **VCDB** [51] is used as the training dataset to generate triplets for training our models. The **CC_WEB_VIDEO** [134] and **SVD** [50] simulate the problem of NDVR. Regarding the former dataset, we found several quality issues with the annotations,

e.g., numerous positives mislabelled as negatives. Hence, we provide results on a 'cleaned' version of the annotations. We also use two evaluation settings, one measuring performance only on the query sets, and a second on the entire dataset. The latter dataset consists of 1,206 queries split into two sets, i.e., training and test set with 1,000 and 206 queries, respectively. We use the test set provided by the authors to benchmark the performance of the retrieval systems. Our **FIVR-200K** [63] is used for the simulation of the FIVR problem (Chapter 3). For quick comparisons of the different variants, we use **FIVR-5K**, a subset of FIVR-200K by selecting the 50 most difficult queries in the DSVR task (using [65] to measure difficulty), and for each one randomly picking the 30% of annotated videos per label category. To add distractors in the subset, we randomly select videos from the FIVR-200K dataset, until the population of FIVR-5K reaches 5K videos. The **EVVE** [99] was designed for the EVR problem. It consists of 620 queries and 2,375 videos collected based on 13 events; yet, we managed to download and process only 503 queries and 1,897 videos (≈80% of the initial dataset) due to the unavailability of the remaining ones. Finally, the **ActivityNet** [17], reorganised based on [28], is used for the AVR task. It consists of 3,791 training, 444 validation, and 494 test videos. The annotations contain the exact video segments that correspond to specific actions. For evaluation, we consider any pair of videos with at least one common label as related.

Implementation details

We extract one frame per second for each video. For all retrieval problems except for AVR, we are using the feature extraction scheme of Section 5.1.2 based on ResNet-50 [36], but for efficiency purposes only extract intermediate features from the output maps of the four residual blocks. Additionally, the PCA for the whitening layer is learned from 1M region vectors sampled from videos in VCDB. Since AVR is not directly related to content-based problems, we run a separate training session using the training set from ActivityNet dataset. We extract features from the last 3D con-

volutional layer of the I3D architecture [19] by applying max-pooling on the spatial dimensions. We also tested I3D features for the other retrieval problems, but without any significant improvements.

For training, we feed the network with only one video triplet at a time due to GPU memory limitations. We employ Adam optimization [60] with learning rate 10^{-5}. For each epoch, T=1000 triplets are selected per pool. The model is trained for 100 epochs, i.e., 200K iterations, and the best network is selected based on mean Average Precision (mAP) on a validation set. Other parameters are set to $\gamma = 0.5$, $r = 0.1$ and $W = 64$. The weights of the feature extraction CNN and whitening layer remain fixed. Training end-to-end results in a significant performance drop, which we attribute to the domain shift between the training and evaluation sets, as explained in Section 4.3.

5.2.2 Frame-to-frame similarity comparison

This section presents a comparison on FIVR-5K of the proposed feature extraction scheme against several global pooling schemes proposed in the literature. Dot product is used for frame-to-frame similarity calculation. Video-level similarity for all runs is calculated with the application of the raw CS on the generated similarity matrices. The benchmarked feature extraction methods include the Maximum Activations of Convolutions (MAC) [124], Sum-Pooled Convolutional features (SPoC) [9], Regional Maximum Activation of Convolutions (R-MAC) [124], Generalized Mean (GeM) pooling [96] (with initial $p = 3$ (cf. Table 1 in [96]) and intermediate Maximum Activation of Convolutions (iMAC) [65], which is equivalent to the proposed feature extraction for $N = 1$. Additionally, we evaluate the proposed scheme with region levels $L_N, N = 2, 3$, and with two different region vector sizes for each region level. We use PCA to reduce region vectors' size, without applying whitening.

Table 5.2 presents the results of the comparison on FIVR-5K. The proposed scheme with $N = 3$ (L_3-iMAC) achieves the best results on all evaluation tasks by a large

Table 5.2: mAP comparison of proposed feature extraction and similarity calculation against state-of-the-art feature descriptors with dot product for similarity calculation on FIVR-5K. Video similarity is computed based on CS on the derived similarity matrices.

Features	**Dims.**	**DSVR**	**CSVR**	**ISVR**
MAC [124]	2048	0.747	0.730	0.684
SPoC [9]	2048	0.735	0.722	0.669
R-MAC [124]	2048	0.777	0.764	0.707
GeM [96]	2048	0.776	0.768	0.711
iMAC [65]	3840	0.755	0.749	0.689
L_2-iMAC	4x3840	0.814	0.810	0.738
L_2-iMAC	4x512	0.804	0.802	0.727
L_3-iMAC	9x3840	**0.838**	**0.832**	**0.739**
L_3-iMAC	9x256	0.823	0.818	0.738

margin. Furthermore, it is noteworthy that the reduced features achieve competitive performance, especially compared with the global descriptors of similar dimensionality. Hence, in settings where there is insufficient storage space, the reduced ViSiL features offer an excellent trade-off between retrieval performance and storage cost. We also tried to combine the proposed scheme with other pooling schemes, e.g., GeM pooling, but this had no noteworthy impact on the system's performance. Next, we will consider the best performing scheme (L_3-iMAC without dimensionality reduction) as the baseline frame-to-frame similarity scheme **ViSiL**$_f$.

5.2.3 Ablation study

This section provides an in-depth analysis of the retrieval performance of the proposed method under different settings and hyperparameter values. We first assess the impact of each network component on the system's performance. Also, we evaluate ViSiL implemented with different similarity calculation functions for frame-to-frame and video-to-video calculation. We validate the influence of the γ, r, and W hyperparameters on the retrieval performance. Finally, we provide a discussion regarding the computational complexity of the proposed method. All comparisons are carried out on FIVR-5K.

Table 5.3: Ablation studies on FIVR-5K. **W** and **A** stand for whitening and attention mechanism respectively.

Task	DSVR	CSVR	ISVR
ViSiL$_f$	0.838	0.832	0.739
ViSiL$_f$**+W**	0.844	0.837	0.750
ViSiL$_f$**+W+A**	0.856	0.848	0.768
ViSiL$_{sym}$	0.830	0.823	0.731
ViSiL$_v$	**0.880**	**0.869**	**0.777**

Table 5.4: Impact of similarity regularization on the performance of the proposed method on FIVR-5K.

$\mathcal{L}_{reg}$	DSVR	CSVR	ISVR
✗	0.859	0.842	0.756
✓	**0.880**	**0.869**	**0.777**

Impact of network components

We first evaluate the impact of each individual module of the architecture on the retrieval performance of ViSiL. Table 5.3 presents the results of four runs with different configuration settings on FIVR-5K. The attention mechanism in the third run is trained using the main training process. The addition of each component offers additional boost to the performance of the system. The biggest improvement for the DSVR and CSVR tasks, 0.024 and 0.021 of mAP, respectively, is due to employing the CNN model for refined video-level similarity calculation in **ViSiL**$_v$. Also, considerable gains on the ISVR task (0.018 mAP) are due to the application of the attention mechanism. We also report results when the Symmetric Chamfer Distance (SCS) is used for both frame-to-frame and video-to-video similarity calculation (**ViSiL**$_{sym}$). Apparently, the non-symmetric version of the CS works significantly better in this problem. Additionally, we evaluate the impact of the similarity regularization loss $\mathcal{L}_{reg}$ of Equation 5.9 in Table 5.4. This appears to have a notable impact on the retrieval performance of the system. The mAP increases for all three tasks reaching an improvement of more than 0.021 mAP on DSVR and ISVR tasks.

Table 5.5: mAP comparison of four pooling combinations for frame-to-frame and video-to-video similarity calculation on FIVR-5K. **MP** stands for Max-Pooling and **AP** for Average-Pooling.

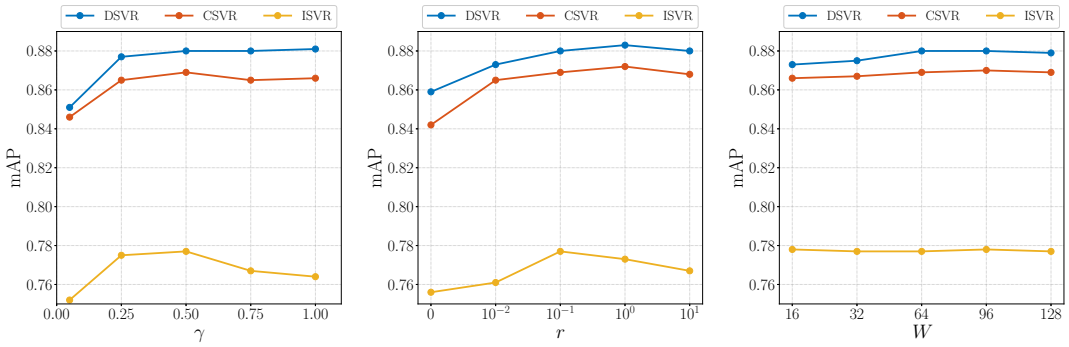
F2F	**V2V**	**DSVR**	**CSVR**	**ISVR**
MP-AP	MP-AP	**0.880**	**0.869**	**0.777**
AP-AP	MP-AP	0.769	0.748	0.682
MP-AP	AP-AP	0.640	0.652	0.623
AP-AP	AP-AP	0.439	0.436	0.341

Table 5.6: mAP comparison of four setups for frame-to-frame and video-to-video similarity calculation on FIVR-5K.

F2F	**V2V**	**DSVR**	**CSVR**	**ISVR**
CS	CS	**0.880**	**0.869**	**0.777**
SCS	CS	0.863	0.854	0.763
CS	SCS	0.836	0.831	0.740
SCS	SCS	0.830	0.823	0.731

Different similarity calculation functions

In this section, we compare the impact of different functions, other than CS, on the frame-to-frame (F2F) and video-to-video (V2V) similarity calculation. In general, CS can be considered equivalent to a Max-Pooling (MP) function followed by Average-Pooling (AP). A different combination could be, e.g., the application of two AP functions. Table 5.5 illustrates the results for different combinations of the core similarity functions of the proposed system on FIVR-5K. It is evident that the use of two AP functions for V2V does not work at all. The run with the two AP for F2F and CS for V2V achieves competitive mAP, but still lower than the run with CS in both functions as proposed. Also, we evaluate different combinations of the CS and SCS similarity functions, for F2F and V2V similarity calculation. Table 5.6 illustrates the results for four different combinations on FIVR-5K. Apparently, the use of two CS works the best on this dataset/problem, whereas the use of two SCS works the worst.



(a) (b) (c)

Figure 5.6: Impact of the margin hyperparameter γ, the regularization parameter r and video snippet size W on the performance of the proposed method on FIVR-5K.

Impact of hyperparameter values

In this section, we compare the impact of different values of hyperparameter γ, r, and W, on the performance of the proposed system. As default values, we use the values reported in implementation settings, i.e. $\gamma = 0.5$, $r = 0.1$ and $W = 64$, and change one at a time.

We first assess the impact of the margin parameter γ on the retrieval performance of the proposed approach. Figure 5.6(a) illustrates the performance of the method trained with different margins on the three tasks of FIVR-5K. Regarding the DSVR task, one may notice that that the performance of the model improves as the margin parameter increases. However, this is not the case for the ISVR task. The approach reports high performance (mAP greater than 0.775) for small values of γ, i.e., within the range [0.25, 0.5], but performance drops as γ increases.

Additionally, we assess the impact of the regularization parameter r on the retrieval performance of the proposed approach. Figure 5.6(b) illustrates the performance of the method trained with different regularization parameters on the three tasks of FIVR-5K. On DSVR and CSVR tasks, the proposed approach achieves the best results for $r = 1.0$ with a considerable margin from the second best, approximately 0.003 mAP.

However, on the ISVR task, the performance significantly dropped in comparison to the default value ($r = 0.1$). For values lower than the default, the proposed approach does not report competitive results on any evaluation task.

Finally, we assess the impact of the size of video snippet W on the retrieval performance of the proposed approach. Figure 5.6(c) depicts the mAP of the method with different values of W on the three tasks of the FIVR-5K dataset. Regarding the DSVR and CSVR tasks, it is evident that the larger the size of video snippets W, the better the performance of the proposed methods. The run with $W = 96$ yields the best results on both tasks with 0.880 and 0.870 mAP, respectively. However, the system's performance on the ISVR task is independent of the size of video snippets used for training since all runs report approximately the same mAP, which is intuitively expected and is a validation that our holistic approach with a single training session can be applied for all content-based problems.

Computational complexity

In this section, we compare the computational complexity of different setups of the proposed approach. The proposed method can be split into two distinct processes, an offline and an online. The offline process comprises of the feature extraction from video frames, whereas the online one the similarity calculation between two videos.

In Table 5.7, we compare the MAC and iMAC runs (cf. Table 5.2) with the ViSiL$_f$ and ViSiL$_v$ in terms of execution time and performance. In that way, we assess the trade-off between the performance gain from the introduction of each component of the method and the associated computational cost. The average length of videos in FIVR-5K is 103 seconds. All the experiments were executed on a machine with an Intel i7-4770K CPU and a GTX1070 GPU.

For the offline process, all runs need approximately the same time to extract frame features. The use of intermediate convolutional layers does not slow down the feature

Table 5.7: mAP and execution time (ms) comparison of four versions of the proposed approach on FIVR-5K. The execution time of the offline process refers to the average feature extraction time per video. The execution time of the online process refers to the average time for the calculation of video similarity of video pairs.

Run	**Comp. Time** (ms)		**FIVR-5K**		
	Offline	**Online**	**DSVR**	**CSVR**	**ISVR**
MAC	950	2.0	0.747	0.730	0.684
iMAC	950	2.3	0.755	0.749	0.689
ViSiL$_f$	960	6.0	0.838	0.832	0.739
ViSiL$_v$	1,040	9.5	0.880	0.869	0.777
LBoW	920	$5 * 10^{-3}$	0.704	0.698	0.657
DML	900	10^{-3}	0.418	0.423	0.389

extraction process since both MAC and iMAC needs 950 ms for feature extraction. The extraction of regional vectors (ViSiL$_f$) has a minor impact on the speed, approximately 1% increase of the total extraction time. Also, the application of whitening and attention-based weighting does not significantly increase the extraction time. ViSiL$_v$ needs 80 ms more than ViSiL$_f$ per video.

Additionally, we compare the proposed method with the two video-level methods presented in Chapter 4 in terms of computation time and performance. The offline process of all methods fluctuates about one second per video. The required time for the online process of the video-level methods is many times lower than the one required by the proposed approach. Nevertheless, this comes with very significant compromises in terms of performance, which significantly drops, especially in the case of DML, and with all limitations discussed in the Chapter 4, i.e., dataset-specific solutions that do not generalize well. Yet, our primary research objective is the maximization of the retrieval performance instead of computational efficiency, and our proposed solution proves its robustness on several retrieval problems.

Regarding the online process, the complexity of calculating the frame-to-frame similarity matrix between videos of M frames each is $O(M^2N^2)$, where N is the number of regions per frame. This is to be compared to $O(M^2)$ of frame-to-frame methods such

Table 5.8: mAP comparison of three ViSiL setups and state-of-the-art methods on the three tasks of FIVR-200K.

Method	DSVR	CSVR	ISVR
LBoW	0.710	0.675	0.572
LAMV [11]	0.515	0.483	0.391
DP [22]	0.775	0.740	0.632
TN [119]	0.724	0.699	0.589
ViSiL$_f$	0.843	0.797	0.660
ViSiL$_{sym}$	0.833	0.792	0.654
ViSiL$_v$	**0.899**	**0.842**	**0.720**

as iMAC (where $N = 1$). Based on our experiments, the MAC and iMAC runs need less than 2.5 ms to calculate video similarity. The computation of the proposed frame-to-frame similarity matrix increases the execution time by 3.7 ms, which is more than a 150% increase (comparing iMAC and ViSiL$_f$). In ViSiL$_v$, the second-stage CNN on the frame-to-frame similarity matrix takes 40% of the execution time, and further increasing it approximately by 3.5 ms but for a significant performance gain.

5.2.4 Comparison against state-of-the-art

We have re-implemented two popular approaches that employ similarity calculation on frame-level representations, i.e., Dynamic Programming (DP) [22] and Temporal Networks (TN) [119]. However, both of them were originally proposed in combination with handcrafted features, which is an outdated practice. Hence, we combine them with the proposed feature extraction scheme and our frame-to-frame similarity calculation. We also implemented a naive adaptation of the publicly available Video re-localization (VReL) method [28] to a retrieval setting, where we rank videos based on the probability of the predicted segment (Equation 12 in the original paper).

Fine-grained incident video retrieval

Here, we evaluate the performance of ViSiL against the state-of-the-art approaches on on our FIVR-200K. We compare with the best performing run reported in Section 3.3, i.e., our LBoW [66] approach implemented with features from VGG [110], the

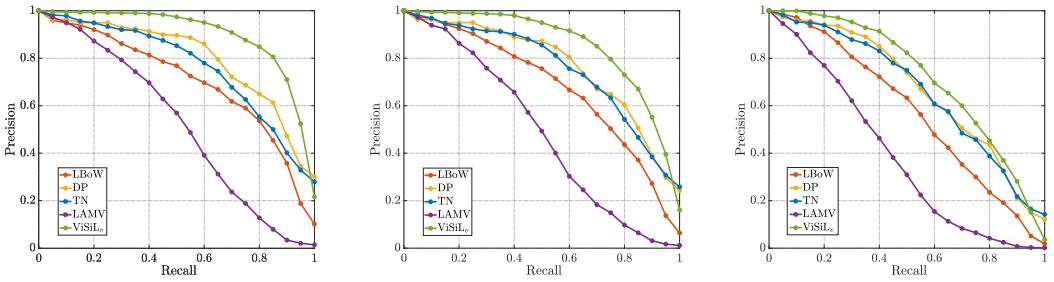


Figure 5.7: PR-curves of the proposed ViSiL approach and state-of-the-art methods on the three tasks of FIVR-200K.

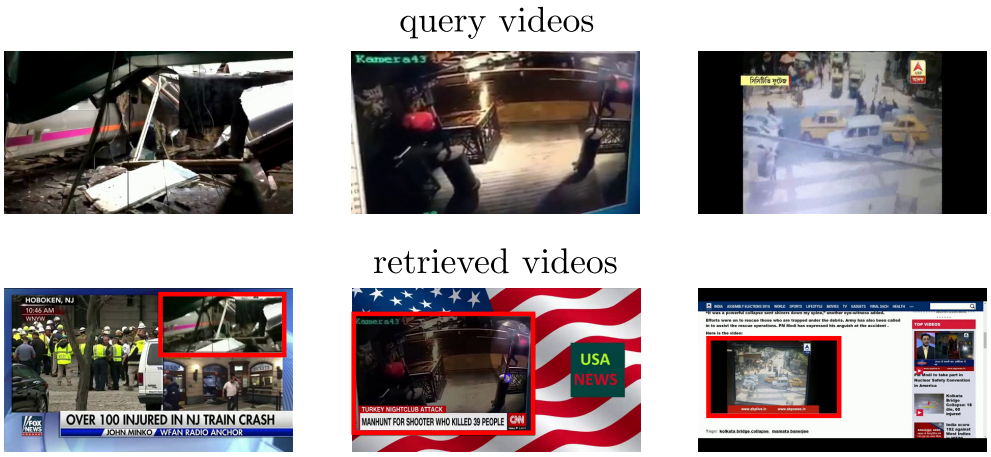


Figure 5.8: Examples of challenging cases of related videos that were mistakenly not labelled as positives in FIVR-200K.

publicly available LAMV [11], and our two re-implementations of DP [22] and TN [119]. Furthermore, we tested our adaptation of VReL [28], but with no success (neither when training on VCDB nor on ActivityNet). As shown in Table 5.8, ViSiL$_v$ outperforms all competing systems, including DP and TN. Its performance is considerably higher on the DSVR task achieving almost 0.9 mAP. Similar conclusions apply for the PR-curves of Figure 5.7. The proposed approach remains on top of all others with a significant margin for almost all precision levels. When conducting a manual inspection of the erroneous results, we came across some interesting cases (among the top-ranked irrelevant videos), which should actually be considered as positive results but were not labelled as such. Figure 5.8 illustrates three such cases, where the query videos are

Table 5.9: mAP of three ViSiL setups and state-of-the-art methods on four different versions of CC_WEB_VIDEO and the SVD dataset. (*) denotes evaluation on the entire dataset, and subscript c that the cleaned version of the annotations was used.

Method	CC_WEB	CC_WEB*	CC_WEB$_c$	CC_WEB$^*_c$	SVD
LBoW	0.976	0.960	0.984	0.975	0.756
DML	0.971	0.941	0.979	0.959	0.785
LAMV [11]	0.971	0.950	0.982	0.969	0.781
DP [22]	0.975	0.958	0.990	0.982	0.861
TN [119]	0.978	0.965	0.991	0.987	0.873
ViSiL$_f$	0.984	0.969	0.993	0.987	0.869
ViSiL$_{sym}$	0.982	0.969	0.991	0.988	0.882
ViSiL$_v$	**0.985**	**0.971**	**0.996**	**0.993**	**0.887**

displayed within the retrieved videos. We should note that the annotators did not miss these cases during the annotation, but they were not presented with them whatsoever, meaning that the calculated visual and textual similarity (Section 3.2.3) was too low. This demonstrates the effectiveness of ViSiL to assign high similarity scores, even when heavy spatial transformations have been applied on the query videos.

Near-duplicate video retrieval

For NDVR, we evaluate the performance of ViSiL against the state-of-the-art approaches on several versions of CC_WEB_VIDEO [134] and on the SVD [50] dataset. The proposed approach is compared with our two video-level methods (Chapter 4), i.e., LBoW [65] and DML [66] implemented with VGG [110] features, the publicly available implementation of Learning to Align and Match Videos (LAMV) [11], and our two re-implementations based on DP [22] and TN [119]. As shown in Table 5.9, The ViSiL$_v$ approach achieves the best performance compared to all competing systems in all cases and on both datasets. When tested on the 'cleaned' version of the CC_WEB_VIDEO, ViSiL achieves almost perfect results in both evaluation settings with 0.996 and 0.993 mAP. It is noteworthy that our re-implementations of the state-of-the-art methods lead to considerably better results on this dataset than the ones reported in the original papers, meaning that direct comparison with the originally reported results would be much more favorable for ViSiL. This is the case because the proposed frame-to-frame

Table 5.10: mAP comparison of three ViSiL setups with the LAMV [11] on EVVE. The ordering of events is the same as in [99]. Our results are reported on a subset of the videos (≈80% of the original dataset) due to unavailability of the full original dataset.

Event	**LAMV**†[11]	**LAMV**$^{\dagger}_{qe}$[11]	**LAMV**[11]	**ViSiL**$_f$	**ViSiL**$_{sym}$	**ViSiL**$_v$
#1	0.715	0.837	0.605	0.889	0.864	**0.918**
#2	0.383	0.500	0.364	0.570	0.704	**0.724**
#3	0.158	0.126	0.197	0.169	0.357	**0.227**
#4	0.461	**0.588**	0.273	0.432	0.440	0.446
#5	0.387	**0.455**	0.361	0.345	0.363	0.390
#6	0.277	0.343	0.297	0.393	0.295	**0.405**
#7	0.247	0.267	0.257	0.297	**0.370**	0.308
#8	0.138	0.142	0.153	0.181	0.214	**0.223**
#9	0.222	0.230	0.381	0.479	0.577	**0.604**
#10	0.273	0.293	0.271	0.564	0.389	**0.578**
#11	0.273	0.216	0.188	0.369	0.266	**0.399**
#12	0.908	**0.950**	0.877	0.885	0.943	0.916
#13	0.691	0.776	0.675	0.799	0.702	**0.855**
mAP	0.536	0.587	0.533	0.589	0.610	**0.631**

similarity calculation scheme is used in combinations with the re-implemented solutions, which also validates the superiority of the proposed video-to-video calculation scheme. Similar conclusions can be drawn from the comparison in the SVD dataset. The proposed approach also achieves state-of-the-art performance with 0.887 mAP, followed by the symmetric variant ViSiL$_{sym}$ with a margin of about 0.005 mAP. Also, the baseline ViSiL$_f$ outperforms the majority of the competing methods.

Event video retrieval

For EVR, we compare ViSiL with the state-of-the-art approach Learning to Align and Match Videos (LAMV) [11]. However, due to the fact that some of the videos are no longer available, we report the results reported in the original paper of LAMV [11], but also the results on the currently available ones that account for ≈80% of the original EVVE dataset for the proposed and competing method. ViSiL performs well on the EVR problem, even without applying any query expansion technique, i.e., Average Query Expansion [26]. As shown in Table 5.10, ViSiL$_v$ achieves the best results on the majority of the events in the dataset. It is outperformed only in four out of the

Table 5.11: mAP comparison of three ViSiL setups and state-of-the-art methods on ActivityNet based on the reorganization from [28].

Method	mAP
DML	0.705
VReL [28]	0.209
LAMV [11]	0.516
DP [22]	0.621
TN [119]	0.648

Method	mAP
ViSiL$_f$	0.652
ViSiL$_{sym}$	**0.745**
ViSiL$_v$	0.710

thirteen events, i.e., in three events from the competing approach with query expansion and one event from the symmetric variant ViSiL$_{sym}$. Additionally, the performance of LAMV in the original and the subset of EVVE is close in the majority of events, indicating that the available subset is suitable for the simulation of the EVR problem.

Action video retrieval

We also assess the performance of the proposed approach on ActivityNet [17] reorganized based on [28]. We compare with our DML [66] approach, the publicly available LAMV [11] approaches, our re-implementations of DP [22] and TN [119], and the adapted version of VReL [28]. For all runs, we extracted features from the I3D network [19]. As shown in Table 5.11, the proposed approach with the symmetric similarity calculation ViSiL$_{sym}$ outperforms all other approaches by a considerable margin (0.035 mAP) to the second best. Moreover, it is noteworthy that the application of the proposed video-to-video similarity calculation scheme significantly improves the system performance, since both ViSiL$_{sym}$ and ViSiL$_v$ outperform the baseline ViSiL$_f$ with a significant margin. Additionally, the baseline of the proposed method outperforms five out of six of the competing methods.

5.3 Conclusions

In this chapter, we proposed a network that learns to compute the similarity between pairs of videos by considering their inter- and intra-frame relations. The key contri-

butions of ViSiL are a) a frame-to-frame similarity computation scheme that captures similarities at the regional level and b) a supervised video-to-video similarity computation scheme that analyses the frame-to-frame similarity matrix through a CNN network which robustly establishes high similarities between video segments of the compared videos. Combined, they lead to a video similarity computation method that is accounting for both the fine-grained spatial and temporal aspects of video similarity. For the training of the proposed approach, a triplet-based pipeline was established for the optimization of the triplet loss function. We conducted extensive evaluations with different experimental setups, testing the performance of the developed approaches under various settings. The proposed method has been applied to a number of content-based video retrieval problems, where it improved the state-of-art consistently and, in several cases, by a large margin.

Chapter 6

Conclusions and Future Work

Contents

6.1 Discussion and conclusions

In this work, we studied the problem of Fine-grained Incident Video Retrieval (FIVR); our overall aim was the proper formulation of the FIVR problem, and the development and evaluation of video retrieval approaches that solve it. While significant progress has been made during the last years in the field of video retrieval, we found that FIVR is a challenging task that can not be solved with the methods existing in the literature. Several definitions related to our problem had been proposed in the literature. Yet, they were either too narrow, considering only the close to identical videos, or too broad, considering videos that depict the same event or concept, and none of the existing ones addresses the case of the same incident videos. Additionally, many approaches had been proposed that tackle similar retrieval problems. Nevertheless, we noticed several limitations of the proposed systems considering solutions to the FIVR problem. Most approaches usually disregarded the spatio-temporal structure of the similarity that can significantly improve performance. Also, supervised training had not been

sufficiently explored for video retrieval problems. Hence, most of the proposed methods do not provide flexibility with respect to the definition of which videos are related, a property that is necessary for FIVR. In this thesis, we introduced the FIVR problem by providing formal definitions for the association types that determine the relations between video pairs, and also with the composition of the FIVR-200K, a large-scale video dataset that simulates the problem at hand. Moreover, we developed approaches that: i) provide flexibility with respect to FIVR definitions, ii) consider the fine-grained sptatio-temporal relation between videos for similarity calculation, and iii) achieve competitive retrieval performance on FIVR-200K and other large-scale datasets.

We started with the introduction of the FIVR problem. First, we provided definitions for the various types of video associations arising in the more general problem setting of FIVR. We focused on two fundamental associations between similar videos, i.e., duplicate videos and videos of the same incident. Duplicate videos were considered the videos that have been captured by the same camera and depict exactly the same scene regardless of any visual transformations applied. In the second category, we considered videos that capture the same incident. We further slit them to complementary viewpoint videos, i.e., videos that capture the same spatio-temporal span but from a different viewpoint, and same incident video, i.e., videos that capture the same incident at different time intervals.

To address the benchmarking needs of FIVR, we built a large-scale dataset, which we call FIVR-200K. Our goal was to gather a challenging large-scale dataset that ultimately consists of numerous pairs of videos that are associated with each other through the relations of interest. We started by crawling the major global news events that occurred over five years (2013-2017) from Wikipedia and were related to armed conflicts and natural disasters. To collect videos based on the news events, we queried the YouTube API with their headlines. Then, we developed a principled approach based on a video clustering scheme that automatically assessed the suitability of a query video

for performing evaluations for the current problem. For the last step of the dataset composition, we devised a protocol for annotating the dataset according to four labels for video pairs. This pipeline resulted in the collection of 225,960 videos associated with 4,687 Wikipedia news events and 100 selected video queries based on the largest video clusters. The FIVR-200K consisted of 7,100 hours of video with 113s average video length, making it a large-scale dataset. The selected queries have, on average, 123 related videos with multiple types of associations, fulfilling our requirements for the queries to be associated with numerous related videos.

Next, we conducted a thorough experimental study on the dataset comparing five state-of-the-art methods, six feature extraction methods based on deep and handcrafted features, and four video aggregation schemes. For the benchmark, we considered three retrieval tasks that represented different instances of the problem and accepted different labels as relevant, i.e., DSVR, CSVR, and ISVR. The best-performing methods achieved mAP scores of 0.710, 0.675, and 0.572, respectively. In general, we found that the benchmarked approaches exhibited low retrieval performance, even though their results in other related datasets are close to perfect. The main reason for the performance gap is that the vast majority of positive video pairs in FIVR-200K are partially similar, not in their entirety but in small segments. Additionally, FIVR-200K contains a wide variety of user-generated videos about news events of similar nature, resulting in many challenging distractors. This highlights the challenging aspect of the FIVR problem, especially in the case of retrieval of the same incident videos.

Additionally, we proposed two video-level approaches that have been initially designed to tackle the problem of NDVR, which is closely related to the FIVR problem. Such approaches offer high-speed retrieval and scalability; hence, they can be applied on massive datasets. Due to the lack of approaches in the literature that employ deep learning and motivated by its outstanding performance in a wide variety of multimedia problems, we use CNN features extracted from the intermediate convolutional layers.

Global frame descriptors were generated by applying the Maximum Activations of Convolutions (MAC) function on the activations of each convolutional layer. Then, for the first approach, we built an unsupervised scheme that relies on a Bag-of-Word (BoW) representation. Two aggregation schemes were introduced for the generation of video representations: i) a vector aggregation that uses a single codebook to map each frame descriptor to a single visual word, and ii) a layer aggregation that uses multiple codebooks that map the frame descriptors to several visual words. Then, the video representations were stored in an inverted file index for the fast indexing and retrieval, while video similarity was carried out based on the cosine similarity of the tf-idf weighted video representations. Since this method was unsupervised in principle, it can be developed with any video corpus without the need for annotated data. However, its main limitation was that it does not generalize well on new unseen data, and it was inefficient to be retrained from scratch with the new video corpus. To tackle these issues, we developed a second supervised approach based on Deep Metric Learning (DML), which learns an embedding function for video representations. The method was built based on a triplet-wise DML scheme that learned a compact and efficient embedding function that maps videos in a feature space where related videos are closer than the irrelevant ones. The similarity between videos was assessed by their Euclidian distance in the embedding space. Also, we experimented with two fusion schemes: i) an early fusion, where the frame descriptors were first averaged to a global vector and then mapped in the embedding space, and ii) a late fusion, where the frame descriptors were independently mapped in the embedding space and then average to a global video representation.

We conducted extensive evaluations with different experimental setups, testing the performance of the developed approaches under various settings. First, we compared the employed CNN features against several others deep and handcrafted features, and we validated its robust performance. Regarding the BoW approach, we experimented with various sizes for the two proposed aggregations, where the layer aggregation

achieved better retrieval performance. Regarding the DML approach, the late fusion achieved marginally better results compared to the early. Three CNN architectures were benchmarked AlexNet, VGGNet, and GoogleNet, with the second one reporting the best performance in almost all comparisons. Moreover, the evaluation process made it evident that the developed approaches exceed the performance of several state-of-the-art video retrieval approaches. Also, we empirically determined that the DML approach overcomes the limitations imposed by the BoW approach, i.e., it achieves competitive performance even without being trained on part of the evaluation dataset (even though further improvements are possible if such access is possible). Finally, DML performance was improved when trained with a video corpus that simulates the same retrieval scenario. This provides flexibility with respect to the definition of related videos required by the FIVR problem.

Finally, we presented a frame-level approach based on video similarity learning. From our review of the related work, we noticed that a promising direction is exploiting better the spatial and temporal structure of videos in the similarity calculation. However, recent approaches either focused on the spatial processing of frames and completely disregarded temporal information, or considered global frame representations (essentially discarding spatial information) and then considered the temporal alignment among such frame representations. To overcome this limitation, we proposed ViSiL, a video similarity learning method that considers fine-grained spatio-temporal relations between videos to assess their similarity. The main novelties of the proposed approach were: i) A frame-to-frame similarity computation scheme that captured similarities at the regional level. We devised a process based on Tensor Dot product, and Chamfer Similarity applied on PCA-whitened and attention weighted CNN features. With this function, we model the spatial structure in the similarity calculation. ii) A supervised video-to-video similarity computation scheme that analyzed the temporal structure of frame-to-frame similarity in order to calculate video similarity. We built a four-layer CNN that was fed with the similarity matrices generated by the previous

process, and robustly establishes high similarities between video segments of the compared videos. With this function, we model the temporal structure in the similarity calculation. Combined, they lead to a video similarity computation method that is accounting for both the fine-grained spatial and temporal aspects of video similarity. Finally, a triplet-based pipeline was established for the training of the proposed approach. We drew triplets of an anchor, a positive and a negative video from two pools of selected and artificially-generated duplicate video pairs. We optimized our network based on the triplet loss and a proposed similarity regularization loss that penalizes the saturated values in the similarity matrix generated by the CNN network.

We evaluated ViSiL on several video retrieval problems, namely our FIVR problem, Near-Duplicate Video Retrieval (NDVR), Event-based Video Retrieval (EVR), and Action Video Retrieval (AVR) using six public benchmark datasets. In all cases, the proposed method outperformed the state-of-the-art and often by a large margin. We also tested the proposed approach implemented with the symmetric equivalent of Chamfer Similarity, i.e., Symmetric Chamfer Similarity. The proposed version performed better on the problems of NDVR, FIVR, and EVR; whereas, the symmetric version achieved the best results on the AVR problem. In addition, we compared the proposed frame-to-frame similarity function against the common practice, which is the combination of global frame representation with dot product for similarity calculation. Our solutions outperformed the compared feature extraction schemes, even in settings where the dimensionality of the descriptors was almost the same. This highlights the value of modeling the spatial structure in the similarity calculation. Also, the contribution of each system component was validated. We found that the video-to-video similarity calculation component had a major impact on the performance of the systems, which confirms that the modeling of the temporal structure can improve the retrieval results. Finally, we validated the impact of the similarity regularization function, which demonstrated considerable performance improvement.

6.2 Future extensions

Although the research community has invested considerable effort in video retrieval problems, there is plenty of room for improvements, and no retrieval problem can be considered solved. An auspicious direction for future work is the exploitation of different aggregation methods that generate more comprehensive frame and video representations. Trainable pooling layers, such as NetVLAD [8], or Transformers [127] have been successfully employed for the generation of global video representations in other video problems, i.e., video classification [87], offering significant performance improvement. Hence, their applicability in the case of video retrieval worth to be validated. Another very promising direction is building training schemes that do not rely on supervised training and can be applied to any video corpus. Knowledge distillation via a Teacher-Student (TS) network setup [38] is an emerging topic with lots of applications in several computer vision problems. A TS framework can be established, where the student network is a compact video-level architecture that tries to mimic the similarity scores calculated from a teacher frame-level architecture so as to improve its retrieval performance.

Furthermore, regarding our ViSiL approach, a direction of future work could be the investigation of ways to reduce the computational complexity of the approach without significant compromises in retrieval accuracy. This could be achieved with a network component that binarizes the region-level features and is trained through the deployed learning scheme. Also, the proposed scheme could be exploited for the corresponding detection problems (e.g., video copy detection, re-localization). A possible solution could be the use of a Region of Interest (RoI) pooling layer on the network's output, which can localize the particular video segments with high similarity scores. Beyond the visual analysis aspect of the problems, the proposed method can be extended for cross-domain retrieval, i.e., by utilizing audio information that can be processed in the same manner.

Additionally, a possible direction for future work could be the combination of the proposed methods in a filter-and-refine scheme. Reinforcement Learning (RL) has been successfully applied on many decision-making problems; hence, a possible solution could be the build of methods based on popular Temporal Difference algorithms, such as Q-learning [131] or its deep learning equivalent with Deep Q networks [89]. Given a video pair for comparison, the RL system receives as input the calculated similarity from a fast video-level approach (e.g., DML) along with other measures for the two compared videos (e.g., video duration, self-similarity, number of segments), and decides whether the calculation with a computationally expensive approach (e.g., ViSiL) have to be performed.

Finally, due to its large size and the wide variety of user-generated videos and news events, FIVR-200K could also facilitate many similar research problems, such as audio-based video retrieval, event reconstruction, and synchronization. In the future, the extension of the dataset annotation should be considered to cover the needs of such problems. In that way, a better understanding of the enclosed news events and the different relations between video pairs would be gained, which will help with the exploration of different use cases and opportunities for the dataset.

Bibliography

[1] "TREC Video Retrieval Evaluation: TRECVID," 2018. [Online]. Available: https://trecvid.nist.gov/ 32

[2] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/ 51

[3] R. Abobeah, A. Shoukry, and J. Katto, "Video alignment using bi-directional attention flow in a multi-stage learning model," *IEEE Access*, vol. 8, pp. 18 097–18 109, 2020. 25, 26, 34

[4] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "Youtube-8m: A large-scale video classification benchmark," *arXiv preprint arXiv:1609.08675*, 2016. 33

[5] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE transactions on Computers*, vol. 100, no. 1, pp. 90–93, 1974. 28

[6] X. Ai, Y. He, Y. Hu, and W. Tian, "Inter-frame relationship graph based near-duplicate video clip detection method," in *Chinese Conference on Image and Graphics Technologies*. Springer, 2019, pp. 70–79. 28

[7] R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov, A. Belopolsky *et al.*, "Theano: A Python

framework for fast computation of mathematical expressions," *arXiv preprint arXiv:1605.02688*, 2016. 89

[8] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5297–5307. 73, 138

[9] 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. 73, 119, 120

[10] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, and S. Vassilvitskii, "Scalable k-means++," *Proceedings of the VLDB Endowment*, vol. 5, no. 7, pp. 622–633, 2012. 88

[11] L. Baraldi, M. Douze, R. Cucchiara, and H. Jégou, "LAMV: Learning to align and match videos with kernelized temporal layers," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7804–7813. xv, 7, 21, 22, 25, 26, 34, 35, 105, 126, 127, 128, 129, 130

[12] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf, "Parametric correspondence and chamfer matching: two new techniques for image matching," in *Proceedings of the International Joint Conference on Artificial intelligence*, 1977, pp. 659–663. 108

[13] A. Basharat, Y. Zhai, and M. Shah, "Content based video matching using spatiotemporal volumes," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 360–377, 2008. 6, 17

[14] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proceedings of the European Conference on Computer Vision*. Springer, 2006, pp. 404–417. 28

[15] S. Bird and E. Loper, "NLTK: the natural language toolkit," in *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*. Association for Computational Linguistics, 2004, p. 31. 47, 55

[16] M. Brown, "Reporting on Las Vegas, Pixel by Pixel," 2017. [Online]. Available: https://www.nytimes.com/2017/10/23/insider/reporting-on-las-vegas-pixel-by-pixel.html 2, 3

[17] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2015, pp. 961–970. 33, 118, 130

[18] Y. Cai, L. Yang, W. Ping, F. Wang, T. Mei, X.-S. Hua, and S. Li, "Million-scale near-duplicate video retrieval system," in *Proceedings of the 19th ACM international conference on Multimedia*. ACM, 2011, pp. 837–838. 19, 22, 30, 59, 62, 89

[19] J. Carreira and A. Zisserman, "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. IEEE, 2017, pp. 4724–4733. 60, 61, 119, 130

[20] J. Chen and T. Huang, "A robust feature extraction algorithm for audio fingerprinting," in *Pacific-Rim Conference on Multimedia*. Springer, 2008, pp. 887–890. 28

[21] C.-Y. Chiu, C.-S. Chen, and L.-F. Chien, "A framework for handling spatiotemporal variations in video copy detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 3, pp. 412–417, 2008. 23

[22] C.-L. Chou, H.-T. Chen, and S.-Y. Lee, "Pattern-based near-duplicate video retrieval and localization on web-scale videos," *IEEE Transactions on Multimedia*,

vol. 17, no. 3, pp. 382–395, 2015. 8, 21, 24, 27, 28, 30, 34, 89, 105, 126, 127, 128, 130

[23] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the twentieth annual symposium on Computational geometry*, 2004, pp. 253–262. 28

[24] 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*. Ieee, 2009, pp. 248–255. 60, 78, 79, 88, 90

[25] M. Douze, H. Jégou, and C. Schmid, "An image-based approach to video copy detection with spatio-temporal post-filtering," *IEEE Transactions on Multimedia*, vol. 12, no. 4, pp. 257–266, 2010. 2, 7, 23, 26, 105

[26] M. Douze, J. Revaud, C. Schmid, and H. Jégou, "Stable hyper-pooling and query expansion for event detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1825–1832. 129

[27] D. Feng, W.-C. Siu, and H. J. Zhang, *Multimedia information retrieval and management: Technological fundamentals and applications*. Springer Science & Business Media, 2013. 3

[28] Y. Feng, L. Ma, W. Liu, T. Zhang, and J. Luo, "Video re-localization," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 51–66. xv, 7, 25, 26, 34, 118, 126, 127, 130

[29] L. Gao, P. Wang, J. Song, Z. Huang, J. Shao, and H. T. Shen, "Event video mashup: From hundreds of videos to minutes of skeleton." in *31st AAAI Conference on Artificial Intelligence, AAAI 2017*. AAAI press, 2017, pp. 1323–1330. 2, 3

[30] Z. Gao, G. Hua, D. Zhang, N. Jojic, L. Wang, J. Xue, and N. Zheng, "ER3: A unified framework for event retrieval, recognition and recounting," in *Proceedings*

of the IEEE conference on Computer Vision and Pattern Recognition, 2017, pp. 2253–2262. 7, 18, 19, 105

[31] Z. Gao, L. Wang, N. Jojic, Z. Niu, N. Zheng, and G. Hua, "Video imprint," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 12, pp. 3086–3099, 2018. 18, 19

[32] Z. J. Guzman-Zavaleta and C. Feregrino-Uribe, "Partial-copy detection of non-simulated videos using learning at decision level," *Multimedia Tools and Applications*, vol. 78, no. 2, pp. 2427–2446, 2019. 24

[33] A. Hanjalic, R. L. Lagendijk, and J. Biemond, "Automated high-level movie segmentation for advanced video-retrieval systems," *IEEE transactions on circuits and systems for video technology*, vol. 9, no. 4, pp. 580–588, 1999. 40

[34] Y. Hao, T. Mu, J. Y. Goulermas, J. Jiang, R. Hong, and M. Wang, "Unsupervised t-distributed video hashing and its deep hashing extension," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5531–5544, 2017. 20

[35] Y. Hao, T. Mu, R. Hong, M. Wang, N. An, and J. Y. Goulermas, "Stochastic multiview hashing for large-scale near-duplicate video retrieval," *IEEE Transactions on Multimedia*, vol. 19, no. 1, pp. 1–14, 2017. 20, 89

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778. 60, 118

[37] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern recognition*, vol. 42, no. 3, pp. 425–436, 2009. 23

[38] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015. 138

[39] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997. 26

[40] Y. Hu and X. Lu, "Learning spatial-temporal features for video copy detection by the combination of cnn and rnn," *Journal of Visual Communication and Image Representation*, vol. 55, pp. 21–29, 2018. 24

[41] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. IEEE, 1997, pp. 762–768. 19, 59, 60, 89

[42] Z. Huang, H. T. Shen, J. Shao, B. Cui, and X. Zhou, "Practical online near-duplicate subsequence detection for continuous video streams," *IEEE Transactions on Multimedia*, vol. 12, no. 5, pp. 386–398, 2010. 25

[43] Z. Huang, H. T. Shen, J. Shao, X. Zhou, and B. Cui, "Bounded coordinate system indexing for real-time video clip search," *ACM Transactions on Information Systems (TOIS)*, vol. 27, no. 3, p. 17, 2009. 18

[44] H. Jégou and O. Chum, "Negative evidences and co-occurences in image retrieval: The benefit of pca and whitening," in *Proceedings of the European Conference on Computer Vision*. Springer, 2012, pp. 774–787. 109

[45] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proceedings of the European Conference on Computer Vision*. Springer, 2008, pp. 304–317. 23

[46] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 3304–3311. 60, 73

[47] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embed-

ding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678. 88

[48] J. Jiang, Y. Tong, H. Lu, B. Cui, K. Lei, and L. Yu, "GVoS: a general system for near-duplicate video-related applications on storm," *ACM Transactions on Information Systems (TOIS)*, vol. 36, no. 1, pp. 1–36, 2017. 18, 19

[49] M. Jiang, Y. Tian, and T. Huang, "Video copy detection using a soft cascade of multimodal features," in *2012 IEEE International Conference on Multimedia and Expo*. IEEE, 2012, pp. 374–379. 27, 28

[50] Q.-Y. Jiang, Y. He, G. Li, J. Lin, L. Li, and W.-J. Li, "SVD: A large-scale short video dataset for near-duplicate video retrieval," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5281–5289. 29, 31, 117, 128

[51] Y.-G. Jiang, Y. Jiang, and J. Wang, "VCDB: A large-scale database for partial copy detection in videos," in *Proceedings of the European Conference on Computer Vision*. Springer, 2014, pp. 357–371. 6, 23, 26, 29, 32, 46, 60, 85, 105, 117

[52] Y.-G. Jiang and J. Wang, "Partial copy detection in videos: A benchmark and an evaluation of popular methods," *IEEE Transactions on Big Data*, vol. 2, no. 1, pp. 32–42, 2016. 23, 24, 26, 34, 105

[53] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang, "Exploiting feature and class relationships in video categorization with regularized deep neural networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 2, pp. 352–364, 2017. 33

[54] Z. Jin, J. Cao, Y. Zhang, J. Zhou, and Q. Tian, "Novel visual and statistical image features for microblogs news verification," *IEEE transactions on multimedia*, vol. 19, no. 3, pp. 598–608, 2017. 2

[55] W. Jing, X. Nie, C. Cui, X. Xi, G. Yang, and Y. Yin, "Global-view hashing: harnessing global relations in near-duplicate video retrieval," *World Wide Web*, pp. 1–19, 2018. 20

[56] jQuery, "jquery — new wave javascript," https://github.com/jquery/jquery, 2017. 51

[57] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2004, pp. II–II. 90

[58] K.-R. Kim, W.-D. Jang, and C.-S. Kim, "Frame-level matching of near duplicate videos based on ternary frame descriptor and iterative refinement," in *Image Processing (ICIP), 2015 IEEE International Conference on.* IEEE, 2015, pp. 31–35. 23

[59] S. Kim, S. H. Lee, and Y. M. Ro, "Rotation and flipping robust region binary patterns for video copy detection," *Journal of Visual Communication and Image Representation*, vol. 25, no. 2, pp. 373–383, 2014. 18, 19

[60] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 89, 119

[61] G. Kordopatis-Zilos, S. Papadopoulos, and I. Kompatsiaris, "Geotagging text content with language models and feature mining," *Proceedings of the IEEE*, vol. 105, no. 10, pp. 1971–1986, 2017. 55

[62] G. Kordopatis-Zilos, S. Papadopoulos, I. Patras, and I. Kompatsiaris, "Finding near-duplicate videos in large-scale collections," in *Video Verification in the Fake News Era.* Springer, 2019, pp. 91–126. 21

[63] G. Kordopatis-Zilos, S. Papadopoulos, I. Patras, and I. Kompatsiaris, "FIVR: Fine-grained incident video retrieval," *IEEE Transactions on Multimedia*, 2019. 87, 118

[64] G. Kordopatis-Zilos, S. Papadopoulos, I. Patras, and I. Kompatsiaris, "ViSiL: Fine-grained spatio-temporal video similarity learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 21, 22

[65] G. Kordopatis-Zilos, S. Papadopoulos, I. Patras, and Y. Kompatsiaris, "Near-duplicate video retrieval by aggregating intermediate cnn layers," in *Proceedings of the international conference on Multimedia Modeling*. Springer, 2017, pp. 251–263. 59, 60, 62, 109, 118, 119, 120, 128

[66] G. Kordopatis-Zilos, S. Papadopoulos, I. Patras, and Y. Kompatsiaris, "Near-duplicate video retrieval with deep metric learning," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*. IEEE, 2017, pp. 347–356. 46, 59, 62, 105, 126, 128, 130

[67] W. Kraaij and G. Awad, "Trecvid 2011 content-based copy detection: Task overview," *Online Proceedings of TRECVid 2010*, 2011. 2, 29, 32

[68] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105. 87

[69] J. Law-To, A. Joly, and N. Boujemaa, "Muscle-VCD-2007: a live benchmark for video copy detection," 2007. 16, 29, 31

[70] H. Lee, J. Lee, J. Y.-H. Ng, and P. Natsev, "Large scale video representation learning via relational graph clustering," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2020. 18, 19, 21, 22, 34

[71] J. Lee, S. Abu-El-Haija, B. Varadarajan, and A. Natsev, "Collaborative deep metric learning for video understanding," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 481–490. 18, 19, 21, 22

[72] J. Li, H. Zhang, W. Wan, and J. Sun, "Two-class 3D-CNN classifiers combination for video copy detection," *Multimedia Tools and Applications*, pp. 1–13, 2018. 25, 26

[73] S. Li, Z. Chen, J. Lu, X. Li, and J. Zhou, "Neighborhood preserving hashing for scalable video retrieval," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8212–8221. 20, 21, 34

[74] D. Liang, L. Lin, R. Wang, J. Shao, C. Wang, and Y.-W. Chen, "Unsupervised teacher-student model for large-scale video retrieval," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0. 23, 24, 26, 34, 35

[75] S. Liang and P. Wang, "An efficient hierarchical near-duplicate video detection algorithm based on deep semantic features," in *Proceedings of the international conference on Multimedia Modeling*. Springer, 2020, pp. 752–763. 28, 34

[76] K. Liao, H. Lei, Y. Zheng, G. Lin, C. Cao, M. Zhang, and J. Ding, "IR feature embedded bof indexing method for near-duplicate video retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018. 23, 26

[77] V. E. Liong, J. Lu, Y.-P. Tan, and J. Zhou, "Deep video hashing," *IEEE Transactions on Multimedia*, vol. 19, no. 6, pp. 1209–1219, 2017. 7, 20, 21, 22

[78] A. J. Lipton, J. Clark, P. Brewer, P. L. Venetianer, and A. J. Chosak, "Objectvideo forensics: Activity-based video indexing and retrieval for physical security applications," 2004. 3

[79] H. Liu, Q. Zhao, H. Wang, P. Lv, and Y. Chen, "An image-based near-duplicate video retrieval and localization using improved edit distance," *Multimedia Tools and Applications*, vol. 76, no. 22, pp. 24 435–24 456, 2017. 24, 27, 28

[80] H. Liu, H. Lu, and X. Xue, "A segmentation and graph-based video sequence matching method for video copy detection," *IEEE Transactions on knowledge and data engineering*, vol. 25, no. 8, pp. 1706–1718, 2013. 23

[81] J. Liu, Z. Huang, H. Cai, H. T. Shen, C. W. Ngo, and W. Wang, "Near-duplicate video retrieval: Current research and future trends," *ACM Computing Surveys (CSUR)*, vol. 45, no. 4, p. 44, 2013. 15

[82] L. Liu, W. Lai, X.-S. Hua, and S.-Q. Yang, "Video histogram: A novel video signature for efficient web video duplicate detection," in *Proceedings of the international conference on Multimedia Modeling*. Springer, 2007, pp. 94–103. 18

[83] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004. 23, 28

[84] F. Magliani, N. M. Bidgoli, and A. Prati, "A location-aware embedding technique for accurate landmark recognition," in *Proceedings of the 11th International Conference on Distributed Smart Cameras*, 2017, pp. 9–14. 26

[85] J. Mairs, "Forensic Architecture to create 3D video of Grenfell Tower fire," 2018. [Online]. Available: https://www.dezeen.com/2018/03/23/forensic-architecture-grenfell-tower-fire-3d-video-reconstruction-london-uk-news/ 2, 3

[86] L. Mengyang, L.-M. Po, Z. Chang, W. Y. Yuen, H.-K. Cheung, H. Peter, H.-T. Luk, and K.-W. Lau, "Content-based video copy detection using binary object fingerprints," in *2018 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*. IEEE, 2018, pp. 1–6. 23, 26

[87] A. Miech, I. Laptev, and J. Sivic, "Learnable pooling with context gating for video classification," *arXiv preprint arXiv:1706.06905*, 2017. 73, 138

[88] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International journal of computer vision*, vol. 60, no. 1, pp. 63–86, 2004. 23

[89] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013. 139

[90] X. Nie, W. Jing, C. Cui, J. Zhang, L. Zhu, and Y. Yin, "Joint multi-view hashing for large-scale near-duplicate video retrieval," *IEEE Transactions on Knowledge and Data Engineering*, 2019. 20

[91] X. Nie, X. Li, J. Sun, and Y. Yin, "Ufvh: unified feature video hashing for near-duplicate video retrieval," in *Proceedings of the Workshop on Visual Analysis in Smart and Connected Communities*, 2017, pp. 17–24. 20

[92] O. Papadopoulou and S. Papadopoulos, "On the Ephemerality of Web Media," 2020. [Online]. Available: https://mever.iti.gr/web/2020/03/10/on-the-ephemerality-of-web-media/ 69

[93] O. P. Popoola and K. Wang, "Video-based abnormal human behavior recognition—a review," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 865–878, 2012. 3

[94] S. Poullot, S. Tsukatani, A. Phuong Nguyen, H. Jégou, and S. Satoh, "Temporal matching kernel with explicit feature maps," in *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015, pp. 381–390. 25, 35

[95] F. Radenović, G. Tolias, and O. Chum, "CNN image retrieval learns from bow: Unsupervised fine-tuning with hard examples," in *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 3–20. 22

[96] F. Radenović, G. Tolias, and O. Chum, "Fine-tuning CNN image retrieval with no human annotation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 119, 120

[97] A. S. Razavian, J. Sullivan, S. Carlsson, and A. Maki, "Visual instance retrieval with deep convolutional networks," *ITE Transactions on Media Technology and Applications*, vol. 4, no. 3, pp. 251–258, 2016. 73

[98] J. Ren, F. Chang, T. Wood, and J. R. Zhang, "Efficient video copy detection via aligning video signature time series," in *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval.* ACM, 2012, p. 14. 25

[99] J. Revaud, M. Douze, C. Schmid, and H. Jégou, "Event retrieval in large video collections with circulant temporal encoding," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.* IEEE, 2013, pp. 2459–2466. xv, 6, 7, 16, 25, 29, 33, 35, 118, 129

[100] Z. Sabeur, N. Doulamis, L. Middleton, B. Arbab-Zavar, G. Correndo, and A. Amditis, "Multi-modal computer vision for the detection of multi-scale crowd physical motions and behavior in confined spaces," in *International Symposium on Visual Computing.* Springer, 2015, pp. 162–173. 3

[101] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823. 22

[102] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," *arXiv preprint arXiv:1611.01603*, 2016. 26

[103] L. Shang, L. Yang, F. Wang, K.-P. Chan, and X.-S. Hua, "Real-time large scale near-duplicate web video retrieval," in *Proceedings of the 18th ACM international conference on Multimedia.* ACM, 2010, pp. 531–540. 18, 19, 30

[104] J. Shao, X. Wen, B. Zhao, and X. Xue, "Temporal context aggregation for video retrieval with contrastive learning." 21, 22, 23, 24, 34, 35

[105] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 806–813. 72

[106] H. T. Shen, X. Zhou, Z. Huang, J. Shao, and X. Zhou, "UQLIPS: a real-time near-duplicate video clip detection system," in *Proceedings of the 33rd international conference on Very large data bases*. VLDB Endowment, 2007, pp. 1374–1377. 15, 16, 18, 35

[107] L. Shen, R. Hong, and Y. Hao, "Advance on large scale near-duplicate video retrieval," *Frontiers of Computer Science*, vol. 14, no. 5, p. 145702, 2020. 15

[108] N. Shuyo, "Language detection library for java," 2010. [Online]. Available: http://code.google.com/p/language-detection/ 55

[109] C. Silverman, "Verification handbook," 2013. 2

[110] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Proceedings of International Conference on Learning Representations*, 2015. 60, 87, 126, 128

[111] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proceedings of the IEEE International Conference Computer Vision*. IEEE, 2003, pp. 1470–1477. 19, 22, 23, 28, 35, 77

[112] J. Song, L. Gao, L. Liu, X. Zhu, and N. Sebe, "Quantization-based hashing: a general framework for scalable image and video retrieval," *Pattern Recognition*, vol. 75, pp. 175–187, 2018. 20

[113] J. Song, Y. Yang, Z. Huang, H. T. Shen, and R. Hong, "Multiple feature hashing for real-time large scale near-duplicate video retrieval," in *Proceedings of the 19th ACM international conference on Multimedia*. ACM, 2011, pp. 423–432. 7, 20, 29, 30

[114] J. Song, Y. Yang, Z. Huang, H. T. Shen, and J. Luo, "Effective multiple feature hashing for large-scale near-duplicate video retrieval," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 1997–2008, 2013. 20, 59, 62

[115] J. Song, H. Zhang, X. Li, L. Gao, M. Wang, and R. Hong, "Self-supervised video hashing with hierarchical binary auto-encoder," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3210–3221, 2018. 7, 20, 34

[116] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012. 61

[117] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *AAAI*, vol. 4, 2017, p. 12. 60

[118] 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. 87

[119] H.-K. Tan, C.-W. Ngo, R. Hong, and T.-S. Chua, "Scalable detection of partial near-duplicate videos by visual-temporal consistency," in *Proceedings of the 17th ACM international conference on Multimedia*. ACM, 2009, pp. 145–154. 7, 23, 126, 127, 128, 130

[120] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, "Yfcc100m: The new data in multimedia research," *Communications of the ACM*, vol. 59, no. 2, pp. 64–73, 2016. 33

[121] Y. Tian, T. Huang, M. Jiang, and W. Gao, "Video copy-detection and localization with a scalable cascading framework," *IEEE MultiMedia*, vol. 20, no. 3, pp. 72–86, 2013. 27

[122] Y. Tian, M. Qian, and T. Huang, "TASC: A transformation-aware soft cascading approach for multimodal video copy detection," *ACM Transactions on Information Systems (TOIS)*, vol. 33, no. 2, p. 7, 2015. 8, 27, 28

[123] Y. Tian, M. Jiang, L. Mou, X. Rang, and T. Huang, "A multimodal video copy detection approach with sequential pyramid matching," in *2011 18th IEEE International Conference on Image Processing*. IEEE, 2011, pp. 3629–3632. 27

[124] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of cnn activations," in *Proceedings of the International Conference on Learning Representations*, 2016. 22, 35, 61, 73, 109, 119, 120

[125] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4489–4497. 25, 60

[126] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, "scikit-image: image processing in python," *PeerJ*, vol. 2, p. e453, 2014. 67, 102

[127] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, pp. 5998–6008, 2017. 24, 138

[128] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *Proceed-*

ings of the IEEE conference on Computer Vision and Pattern Recognition, 2014, pp. 1386–1393. 22

[129] L. Wang, Y. Bao, H. Li, X. Fan, and Z. Luo, "Compact cnn based video representation for efficient video copy detection," in *Proceedings of the international conference on Multimedia Modeling*. Springer, 2017, pp. 576–587. 23, 24

[130] A. Wary and A. Neelima, "A review on robust video copy detection," *International Journal of Multimedia Information Retrieval*, vol. 8, no. 2, pp. 61–78, 2019. 16

[131] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992. 24, 139

[132] S. Wei, Y. Zhao, C. Zhu, C. Xu, and Z. Zhu, "Frame fusion for video copy detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 1, pp. 15–28, 2011. 23

[133] I. H. Witten and E. Frank, "Data mining: practical machine learning tools and techniques with java implementations," *Acm Sigmod Record*, vol. 31, no. 1, pp. 76–77, 2002. 18

[134] X. Wu, A. G. Hauptmann, and C.-W. Ngo, "Practical elimination of near-duplicates from web video search," in *Proceedings of the 15th ACM international conference on Multimedia*. ACM, 2007, pp. 218–227. 6, 7, 8, 15, 16, 18, 21, 27, 29, 30, 35, 58, 62, 86, 90, 117, 128

[135] Z. Wu and K. Aizawa, "Self-similarity-based partial near-duplicate video retrieval and alignment," *International Journal of Multimedia Information Retrieval*, vol. 3, no. 1, pp. 1–14, 2014. 25

[136] F. Yang and S. Satoh, "Burst-survive temporal matching kernel with fibonacci periods," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 2062–2066. 25

[137] Y. Yang and T. Hospedales, "Deep multi-task representation learning: A tensor factorisation approach," in *International Conference on Learning Representations*, 2017. 107

[138] Y. Yang, Y. Tian, and T. Huang, "Multiscale video sequence matching for near-duplicate detection and retrieval," *Multimedia Tools and Applications*, pp. 1–26, 2018. 8, 28

[139] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1480–1489. 110

[140] M.-C. Yeh and K.-T. Cheng, "Video copy detection by fast sequence matching," in *Proceedings of the ACM International Conference on Image and Video Retrieval*. ACM, 2009, p. 45. 23

[141] L. Yuan, T. Wang, X. Zhang, F. E. Tay, Z. Jie, W. Liu, and J. Feng, "Central similarity quantization for efficient image and video retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3083–3092. 20, 21, 22, 34

[142] 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. 73

[143] C. Zhang, Y. Lin, L. Zhu, A. Liu, Z. Zhang, and F. Huang, "CNN-VWII: An efficient approach for large-scale video retrieval by image queries," *Pattern Recognition Letters*, vol. 123, pp. 82–88, 2019. 18

[144] J. R. Zhang, J. Y. Ren, F. Chang, T. L. Wood, and J. R. Kender, "Fast near-duplicate video retrieval via motion time series matching," in *2012 IEEE In-*

ternational Conference on Multimedia and Expo. IEEE, 2012, pp. 842–847. 25

[145] G. Zhao and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 6, pp. 915–928, 2007. 19, 59, 60

[146] L. Zheng, Y. Zhao, S. Wang, J. Wang, and Q. Tian, "Good practice in cnn feature transfer," *arXiv preprint arXiv:1604.00133*, 2016. 22, 35, 72, 73

[147] X. Zhou, L. Chen, and X. Zhou, "Structure tensor series-based large scale near-duplicate video retrieval," *IEEE Transactions on multimedia*, vol. 14, no. 4, pp. 1220–1233, 2012. 27

[148] Z. Zhou, J. Chen, C.-N. Yang, and X. Sun, "Video copy detection using spatio-temporal cnn features," *IEEE Access*, vol. 7, pp. 100 658–100 665, 2019. 28, 34

[149] Y. Zhu, X. Huang, Q. Huang, and Q. Tian, "Large-scale video copy retrieval with temporal-concentration sift," *Neurocomputing*, vol. 187, pp. 83–91, 2016. 25