BSSE FINAL PROJECT

Phase 1

Video Similarity Detection

Project Advisor

Mr. Waqas Ali

Presented by:

Group ID: F23SE074

Student Reg#	**Student Name**
L1F20BSSE0618	Ali Raza
L1F20BSSE0365	M. Hassan Saeed
L1F20BSSE0567	Bilal Ashoob
L1F20BSSE0598	Muhammad Ali

Faculty of Information Technology

University of Central Punjab

Software Requirements Specification

Version <Version # 1>

Video Similarity Detection

Advisor: Mr. Waqas Ali

Group ID: F23SE074

Member Name	Primary Responsibility
Muhammad Ali	Requirement Specification, Implementation and Documentation
M. Hassan Saeed	Requirement Specification, Implementation and Documentation
Ali Raza	Requirement Specification, Implementation and Documentation
Bilal Ashoob	Requirement Specification, Implementation and Documentation

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

Abstract

With the rise of internet socialization went to a next level, people share a lot of text, images and videos to socialize with others, this limits less sharing of visual content bring lot of problems like copy right protection, duplication, misinformation, sensitive and harmful content. The solution to this problem is very costly, time taking, and effort seeking, because daily tens of thousands of videos are uploaded on a daily basis on different social media platforms. In these circumstances performing these tasks manually is not possible and if possible, to an extent will be so costly, to harness this problem there should be a model which will be capable to accurately detecting the derived video content from the reference videos data set with minimum resource consumption. Paying attention especially to video copy detection, there are two main challenges. The first is to detect the manipulated and edited video derived from the dataset and utilization of minimum time and resource consumption. Video copy detection challenge encompasses two tasks, first is descriptor track in which the model should predict that the query video is derived video from the dataset or not by confidence score. Secondly matching track, the model will provide the temporal segment of the copied part in query video. The model will be trained on specially designed dataset for video similarity detection model by Meta named as YFCC100M containing 0.8 million videos with noise removal approximately 15 to 25 seconds which are being dividing into three-part training, validation and testing and the dataset will be shrink due to resource limitation.

The method of the first placed group in the competition will be discussed in this paper. The query video pattern is categorized in three groups: first which are not edited, second which are edited and the third in which frames are containing more than one scene. Dual level method with video editing detection (VED) will tell whether the query video is edited or not. For unedited video the random vectors with norms and bias will be replaced with negative value in descriptor. For editing video for the multiple scenes, the traditional image processing method will be used. By using dual level detection, the computation cost will be reduced. Frame Scene Detection (FSD) is used to deal with the major problems of video copy detection.

For the matching track there are two problems how to extract the features, with embedding or similarity matrix and matching of copy segments in video. Both tracks required the features, by using embedding changing descriptor model the embedding also have to change as compared to embedding similarity matrix doesn’t change so the similarity matrix is used. And for video segment copy we used similarity alignment model (SAM) which takes similarity matrix as an input and gives results in the score matrix.

1. Introduction

1.1 Review of Related Literature

Recently Meta Ai has conducted a challenge on video similarity detection and some of the most prominent position holder methods are being discussed. The first group used dual level detection method with video copy detection and Frame Scene Detection for Descriptor track [1] and Segment Alignment Model for matching track [2]. The second group [3] used Feature Compatible Progressive Learning (FCPL) along with auxiliary pretrained models to achieve the required results on both tracks. The third group [4] used Test Time Augmentation (TTA) and Temporal Networks (TN) for both tracks respectively.

1.2 Problem Statement

The primary challenge at hand is the scalability of content moderation, particularly on platforms like Instagram and Facebook, where tens of thousands of hours of video content are uploaded daily. The requirement for accurate and high-performance algorithms is crucial for effectively dropping and removing inappropriate content. Furthermore, this model can be in various problems like nature wise Advertisement Recommendation, Deep fake detection and nature wise content retrieval.

1.3 Proposed Solution

The project's primary goal is to generate vector representations of videos using advanced deep learning algorithms by classifying the content in three classifications edited, unedited and multi frame videos using with dual level detection method with video editing detection. In the Matching Track, specific emphasis is placed on creating a model capable of directly detecting clips within a query video, corresponding to segments within a larger amount of reference videos by Frame level video decomposition and using Similarity Alignment Model.

1.4 Problem Scope

The project focuses on modifying issues related to content manipulation, misinformation, and the presence of inappropriate content on social media platforms.

1.5 Challenges

Implementing vision transformers, self-supervised copy detection models, image similarity detection models, and convolutional neural networks poses significant challenges that the project aims to overcome.

1.6 Knowledge Areas Required

Successful completion of the project demands expertise in artificial intelligence, deep learning, and a profound understanding of computer vision. Familiarity with key algorithms like YOLO, CNN, and Transformers is essential.

1.7 Completeness Criteria

The project's success will be measured against various criteria, the model should provide pairs of similar videos with temporal segments in the pairs and user will interact with simple webGUI, the efficiency of the model architecture, algorithm development, overall accuracy, and effective communication of results.

1.8 Research Outcomes/Nature of End Product

The tangible outcome of the project will be a user-friendly web-based graphical interface. This interface will facilitate the detection of manipulated clips within query videos, providing users with confidence scores and temporal localization information. Our priority is to gain optimum accuracy This contributes to making social media platforms more trustworthy and secure.

1.9 Learning Outcomes

Participants in the project will gain advanced programming skills in Python, with a specific focus on popular AI libraries such as TensorFlow, NumPy, Pandas, Keras, PyTorch, and transformers. The project will provide a detailed understanding and hands-on experience in implementing algorithms like CNN and YOLO.

2. Background Study and Literature Survey

Image and Video copy detection has been one of the hottest working domains for the past ten years. In 2007 on Muscle-VCD dataset video copy detection work done as a series of TRACvid [8], in 2007 another work found like current web video search (CWeb) [9] was based on keywords or tags and the near duplicate video use to come in the result the model this by matching the signatures derived from color histograms even if this did not work then feature based near-duplicate detection but was comparatively expensive. In 2008 The content-based copy detection (CCD) [10] benchmark worked with a large collection of synthetic queries, which is atypical for TRECVID, as was the use of a normalized detection cost framework. In 2011 Near Duplicate Video Retrieval (NDVR) [11] became popular when it solves the problem of accuracy as previous models was extracting single feature to retrieve duplicate video, but single feature was not enough to do so, here the Multiple Feature Hashing (MFH) technique begets the comparatively good accuracy. In 2013 Event retrieval through Circulant Temporal Encoding (CTE) [12] was very effective in this domain this model works by embedding the frame descriptors to jointly represent the appearance and temporal order and product quantization to complex vectors to compress the descriptors on the new datasets of EVVE. Art Vision base article by Lukas Klic which represents computer vision APIs with the Research Space platform, allowing for the matching of similar artworks and photographs across cultural heritage image collections [5]. TransVLC: an attention-enhanced video copy localization network which is optimized directly from initial frame-level features and trained end-to-end with three main components: customized Transformer, correlation and SoftMax layer for similarity matrix generation, and a temporal alignment module for copied segments localization [6]. George Awad and Keith Curtis evaluated video retrieval tasks at TRECVID 2022 in which TRECVID, a national institute, was researched on in content-based video retrieval [7]. In the context of image similarity 2008 the GIST [12] descriptor had recently received increasing attention in the context of scene recognition. In this paper we evaluate the search accuracy and complexity of the global GIST descriptor for two applications, for which a local description is usually preferred: same location/object recognition and copy detection. We identify the cases in which a global description can reasonably be used.

The comparison is performed against a state-of-the-art bag-of-features representation. To evaluate the impact of GIST's spatial grid, we compare GIST with a bag-of-features restricted to the same spatial grid as in GIST.

3. Overall Description

3.1 Proposed Solution

The project is organized into two primary tracks: The Descriptor Track and the Matching Track. The Descriptor Track focuses on generating vector representations of videos, while the Matching Track involves creating a model for detecting specific clips within query videos. Here we have the following demonstration.

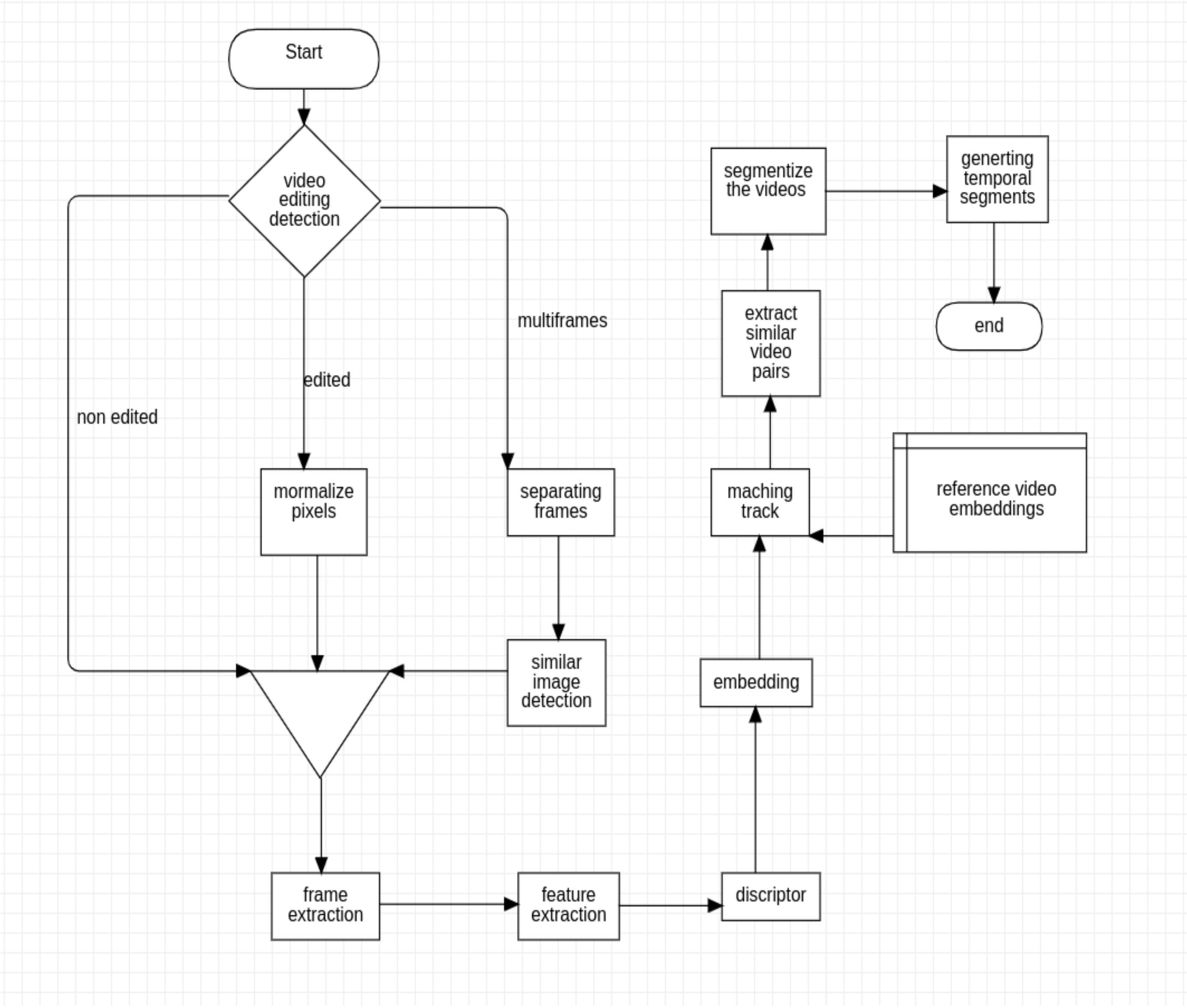


figure 1: System Demonstration

3.2 User Classes and Characteristics

User classes are identified based on their interest in content tracing, copyright enforcement, and ensuring the safety of social media platforms. Characteristics of each user class are considered, including frequency of use, technical expertise, and security levels.

3.3 Operating Environment

The system is designed to operate on GPUs, with specific hardware requirements outlined. The dataset is limited and further shrinks according to the available resources in the institute based on available resources, ensuring optimal utilization of hardware.

3.4 Design and Implementation Constraints

Various constraints, including corporate policies, hardware limitations, interfaces with other applications, and design conventions, are taken into consideration during the development phase.

3.5 Assumptions and Dependencies

Assumptions include the prerequisite knowledge of AI and deep learning, while dependencies involve external datasets, such as YFCC100M and hardware resources like GPUs.

4. Functional Requirements

4.1 Use-Case 1

Identifier	Drag videos.
Purpose	Uploading videos.
Priority	High.
Pre-conditions	Query video must be contained by reference video set.
Post-conditions	Model must provide a pair of similar videos with temporal segments.
Typical Course of Action	

S#	Actor Action	System Response
1	Drag a query video in the video section.	Perform the description and embedding.
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	Corrupt or unseen query video dragged.	Suspicious mark.

Table 1: UC-1

4.2 Use-Case 2

Identifier	Feature Extraction.		
Purpose	Create Embeddings		
Priority	High.		
Pre-conditions	Query video must be contained by reference video set.		
Post-conditions	Model must provide a pair of similar videos with temporal segments.		
Typical Course of Action			
S#	**Actor Action**	**System Response**	
2	Press “Retrieve Replica” button.	Shows Possible pairs of similar videos.	
…			
Alternate Course of Action			
S#	**Actor Action**	**System Response**	
2	Press “Retrieve Replica” button.	Replica not found.	
…			

Table 2: UC-2

4.3 Use-Case 3

Identifier	Temporal segments.

Purpose	Time Stamps Segments.	
Priority	High.	
Pre-conditions	Query video must be contained by reference video set.	
Post-conditions	Model must provide a pair of similar videos with temporal segments.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
3	Press “Run Temporal Segments” button.	Shows the query and referenced matched video segments parallelly.
…		
Alternate Course of Action		
S#	**Actor Action**	**System Response**
3	Press “discard drag” button	Video discarded.
…		

Table 3: UC-3

4.4 Analysis and Modeling of Requirements

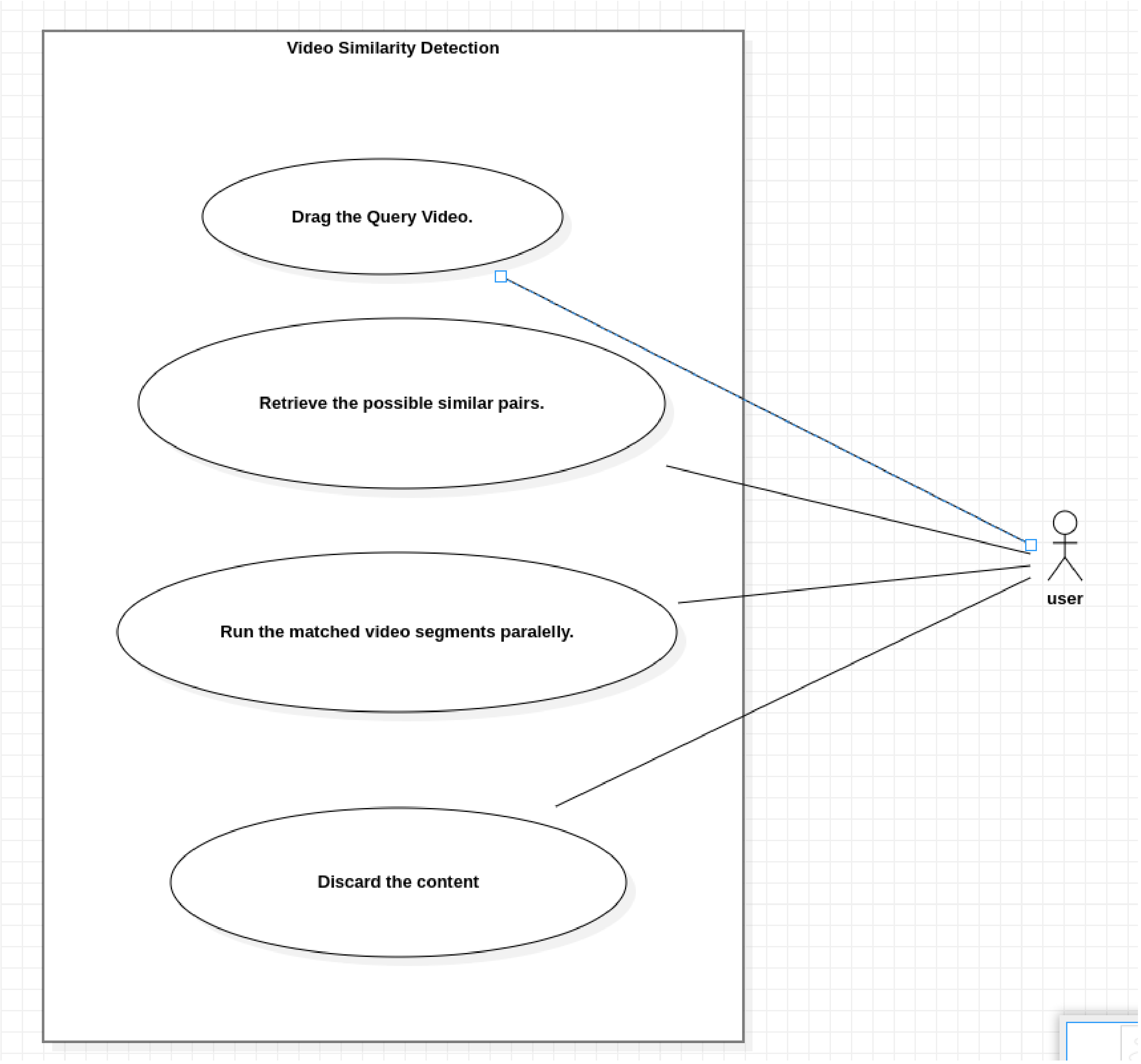

Video Similarity Detection
Drag the Query Video.
Retrieve the possible similar pairs.
Run the matched video segments paralelly.
Discard the content
user

Figure 2: System Use Cases

5. Nonfunctional Requirements

5.1 Target Performance

Accuracy:

The system aims to achieve a high level of accuracy in identifying similarities between query and reference videos yet the query video must by contained by reference set.

Efficiency:

To enhance user experience, video analysis should be completed within a reasonable time frame. The target is an average processing time of a five to fifteen minutes per video to retrieve the similar temporal pairs, ensuring timely results.

5.2 Safety Requirements

Shrink dataset:

As we are not having enough hardware resources to run the complete dataset. We shrink the dataset according to the related available resources to avoid fatality of hardware resources.

Notification Popup:

In cases where the dragged video is corrupted or not present in the reference dataset, during the description and embedding the message will print “the suspicious mark” and by clicking the retrieve replica the message will show “replica not founded”.

Limited edited functionality:

The model only works for the limited editing function for detection editing before description and embedding for instance rotation, resize, blend, and certain filter.

5.3 Additional Software Quality Attributes

Maintainability:

The system's codebase should be well-documented and modular to facilitate easy maintenance and updates. This ensures that future modifications or enhancements can be implemented efficiently.

Scalability:

Considering potential growth in the volume of uploaded videos and the reference set, the system should be designed to scale seamlessly. Scalability is important for accommodating increased user activity and expanding datasets and in the context of editing functions.

Integrated model:

Currently the model can only be interacted by simple GUI web interface. But for future integration the complete pipeline and API structure in needed for effective use.

6. Other Requirements

Hardware Availability:

The system can only be used efficiently on A100 GPU. The production environment must have the hardware processing units mentioned to run the model.

External Interface Requirements:

Compatibility with major web browsers is essential for the graphical user interface. Additionally, the system should provide APIs for potential integration with other systems, enhancing its adaptability.

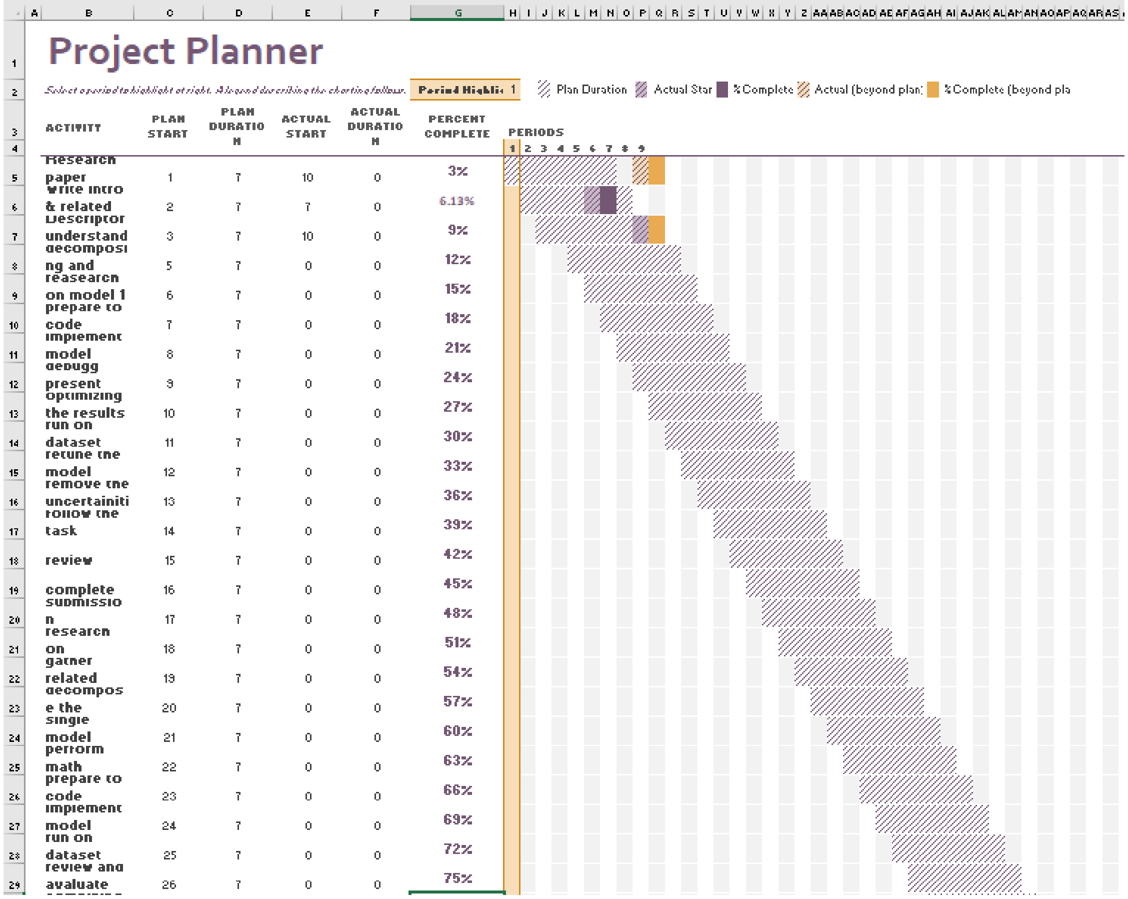
Legal Requirements:

To comply with copyright laws and regulations, the system must ensure that content analysis and removal processes align with legal standards. This includes respecting intellectual property rights and following to relevant legal frameworks.

7. Initial Results

The initial results obtained during the project development phase showcase successful implementation of fundamental functionalities the descriptor and matching track respectively, firstly the descriptor and then matching track. The accuracy of the similarity detection algorithm can be very low, the successful model run will be the first achievement. Moreover, the average processing time per video falls within the acceptable range, meeting the project's performance expectations afterwards.

8. Revised Project Plan



Project Planner

Select a period to highlight at right. A legend describing the charting follows. Period Highlig 1

▨ Plan Duration ▨ Actual Star ■ % Complete ▨ Actual (beyond plan) ■ % Complete (beyond pla

	ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
5	Research paper	1	7	10	0	3%
6	write intro & related	2	7	7	0	6.13%
7	Descriptor understand	3	7	10	0	9%
8	decomposi ng and	5	7	0	0	12%
9	reasearch on model 1	6	7	0	0	15%
10	prepare to code	7	7	0	0	18%
11	implement model	8	7	0	0	21%
12	debugg present	9	7	0	0	24%
13	optimizing the results	10	7	0	0	27%
14	run on dataset	11	7	0	0	30%
15	retune the model	12	7	0	0	33%
16	remove the uncertainiti	13	7	0	0	36%
17	follow the task	14	7	0	0	39%
18	review	15	7	0	0	42%
19	complete	16	7	0	0	45%
20	submissio n	17	7	0	0	48%
21	research on	18	7	0	0	51%
22	gather related	19	7	0	0	54%
23	decompos e the	20	7	0	0	57%
24	single model	21	7	0	0	60%
25	perform math	22	7	0	0	63%
26	prepare to code	23	7	0	0	66%
27	implement model	24	7	0	0	69%
28	run on dataset	25	7	0	0	72%
29	review and avaluate	26	7	0	0	75%

Figure 3: Activity Gannt Chart

9. References

[1]:https://github.com/FeipengMa6/VSC22-Submission/blob/main/VSC22-Descriptor-Track-1st/documents/VSC22-Descriptor-Track-Solutions.pdf.

[2]:https://github.com/FeipengMa6/VSC22-Submission/blob/main/VSC22-Matching-Track-1st/documents/VSC22-Matching-Track-Solutions.pdf.

[3]:https://arxiv.org/pdf/2304.10305.pdf.

[4]:https://arxiv.org/pdf/2304.11964.pdf.

[5]:Lukas Klic I Tatti, The Harvard University Center for Italian Renaissance Studies, Harvard University, Florence,Italy(chromeextension://efaidnbmnnnibpcajpcglclefindmkaj/https://content.iospress.com/download/semantic-web/sw212893?id=semantic-web%2Fsw212893) (Doi: 10.3233/SW-212893).

[6]: https://ojs.aaai.org/index.php/AAAI/article/view/25158.

[7]: https://trecvid.nist.gov/past.data.table.html

[8]:J Law-To, A Joly, and N Boujemaa. Muscle-vcd-2007: alive benchmark for video copy detection, 2007.

[9]:Xiao Wu, Alexander G Hauptmann, and Chong-Wah Ngo. Practical elimination of near-duplicates from web video search. In Proc. ACM MM, pages 218–227, 2007.

[10]:George Awad, Paul Over, and Wessel Kraaij. Content-based video copy detection benchmarking at trecvid. ACM Transactions on Information Systems, 32(3):1–40, 2014.

[11]:Jingkuan Song, Yi Yang, Zi Huang, Heng Tao Shen, and Richang Hong. Multiple feature hashing for real-time large scale near-duplicate video retrieval. In Proc. ACM MM, pages 423–432, 2011.

[12]:Jérôme Revaud, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Event retrieval in large video collections with circulant temporal encoding. In Proc. CVPR, pages 2459–2466, 2013.

[13]: Matthijs Douze, Hervé Jégou, Harsimrat Sandhawalia, Laurent Amsaleg, and Cordelia Schmid. Evaluation of gist descriptors for web-scale image search. In Proc. CIVR, 2009.

[14]: Matthijs Douze, Giorgos Tolias, Ed Pizzi, Zoë Papakipos, Lowik Chanussot, Filip Radenovic, Tomas Jenicek, Maxim Maximov, Laura Leal-Taixé, Ismail Elezi, et al. The 2021 image similarity dataset and challenge. ArXiv preprint arXiv:2106.09672, 2021.

Appendix A: Glossary

CNN: Convolutional Neural Network.

YOLO: You Only Look Once.

FSD: Frame Scene Detection.

VED: Video Editing Detection.

SAM: Similarity Alignment Model.

TTA: Test Time Augmentation.

TN: Temporal Network.

FCPL: Features Compatible Progressive Learning.

CCD: Content Based Copy Detection.

CWEB: Current Web Video Search.

NDVR: Near Duplicate Video Retrieval.

Appendix B: IV & V Report

(Independent verification & validation)

IV & V Resource

Name Signature

S#	**Defect Description**	**Origin Stage**	**Status**	**Fix Time**	
				Hours	**Minutes**
1					
2					
3					
…					

Table 1: List of non-trivial defects