BSSE FINAL PROJECT

Video Similarity Detection

Project Advisor

Mr. Waqas Ali

Presented by:
Group ID: F23SE074

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

Faculty of Information Technology

University of Central Punjab

Complete System
SDP Phase IV

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

Table of Contents i
Revision History iii
Abstract iv
1. Introduction 1
1.1 Product 1
1.2 Background 1
1.3 Objective(s)/Aim(s)/Target(s) 2
1.4 Scope 2
1.5 Business Goals 4
1.6 Document Conventions 5
1.7 Miscellaneous 5
2. Technical Architecture 7
2.1 Application and Data Architecture 7
2.1.1 Application Architecture 7
2.1.2 Data Architecture 8
2.2 Component Interactions and Collaborations 8
2.3 Design Reuse 9
2.3.1 Design Reuse 9
2.4 Technology Architecture 9
2.5 Architecture Evaluation 10
2.5.1 Selection of Infrastructure/Technology 10
2.5.2 Technology Architecture 12
3. Detailed/Component Design 12
3.1 Descriptor Module 14
3.2 Matching Module 14
3.3 Dataset 14
4. Screenshots/Prototype 15
4.1 Workflow 15
4.2 Screens 17
4.2.1 Screen 1 17
4.2.2 Screen 2 17
4.2.3 Screen 3 18
4.2.4 Screen 4 19
5. Test Specification and Results 20
5.1 Test Case Specification 20
5.1.1 TC: 1 20
5.1.2 TC: 2 21
5.1.3 TC: 3 22
5.1.4 TC: 4 23
5.1.5 TC: 5 24
5.1.6 TC: 6 25
5.2 Summary of Test Results 26
7. Project Completion Status 26
8. Deployment/Installation Guide 28
9. User Manual 28
10. References 29

11. Project Summary Form....30
Appendix A: Glossary....33
Appendix B: IV & V Report....34

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 tens of thousands of videos are uploaded daily 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, Video copy detection challenge encompasses two tasks, first is descriptor track in which the model should extract the frames of the video and create its embedding and maintain the descriptors database of the that query video. Secondly matching track, the model will compare the descriptors from the database with descriptors of the query video and provide the temporal segment of the copied part in query video. The model is supposed to 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 but the dataset is to 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 by extraction the frames of the video and saved them for further processing, now the video frames are undergo the process of creating the embedding to reduce computation power and increase efficiency and these embedding are further processed and save in descriptors database 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, embedding also must change, as compared to embedding similarity matrix doesn’t change, so similarity matrix is used. And for video segment copy we used similarity alignment model (SAM).

1. Introduction

1.1 Product

The product of this project is a simple user-friendly web-based graphical interface designed to enhance content moderation on social media platforms. This interface serves as a powerful tool for detecting manipulated clips within query videos. Users would be able to drag a video or give link of the video containing subpart of the referenced video on which the model has been trained, press retrieve replicas button to get same pairs and run temporal segment button to play the same section. Interface would display the possible matched pairs of query and reference video from the dataset according to the confidence score with their potential temporal location providing users with confidence scores and temporal localization information. The primary goal is to offer a solution to the challenges associated with content manipulation, misinformation, and the presence of inappropriate content in the realm of social media.

This graphical interface represents the culmination of advanced deep learning algorithms, particularly focusing on video copy detection. Leveraging the capabilities of artificial intelligence, computer vision, and deep learning techniques, the product aims to deliver a reliable and efficient solution for moderating and ensuring the integrity of visual content shared on various social media platforms.

1.2 Background

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 framelevel 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.

1.3 Objective(s)/Aim(s)/Target(s)

The project's primary target 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 Scope

Ads recommendation system:

Video similarity detection can analyze the content of videos that users have engaged with or viewed. By identifying patterns and similarities in users' video consumption behavior.

Video content classification:

Video similarity detection can be used to classify videos into different genres based on visual and thematic similarities. For example, it can group together action movies, comedies, or dramas by identifying common visual elements or themes within the content.

Crime events investigation:

In the context of crime events investigation, video similarity detection can serve as a valuable tool for law enforcement agencies and investigative teams.

Video surveillance:

Video similarity detection can analyze surveillance footage to identify similarities between different incidents. This helps investigators link events, recognize patterns, and track the movements of individuals involved in criminal activities.

Street crime detection:

Video similarity detection enables real-time monitoring of street surveillance cameras. The system can continuously analyze video feeds to identify patterns or anomalies associated with criminal behavior, such as theft, vandalism, or assault.

Plagiarism and Copyright Infringement:

Video similarity detection models can be used to identify instances of video content that has been copied or plagiarized from other sources, helping content creators and copyright holders to protect their intellectual property.

Content Duplication:

Video platforms and content aggregators can use video similarity detection to identify and remove duplicated or near duplicated content, ensuring that their platforms offer original and diverse content to users.

Content Moderation:

Video platforms can employ similarity detection to identify and flag videos that violate community guidelines or contain inappropriate or offensive content, improving content moderation efforts.

Video Verification:

In cases where the authenticity of a video is questioned, similarity detection models can be used to compare the video in question with other videos to determine if it has been manipulated or altered.

Video Analytics:

Similarity detection can help content creators and marketers analyze their video performance by identifying patterns of similarity among different videos, providing insights into what types of content resonate with their audience.

Educational Content:

Video similarity detection can help educators and students discover educational videos that cover similar topics, enabling more comprehensive learning experiences.

1.5 Business Goals

The business goals in this project are like big targets we want to hit.

Content Moderation in Social Media:

Social media platforms employing video similarity detection aim to moderate and filter content effectively. This ensures compliance with regulations, protects users from inappropriate content, and maintains a positive user experience.

Data-Driven Decision-Making:

Businesses leveraging video similarity detection aim to make informed, data-driven decisions. This includes using insights from video analytics to adapt strategies, improve security protocols, and enhance operational efficiency.

Plagiarism and Copyright Infringement:

Video similarity detection models can be used to identify instances of video content that has been copied or plagiarized from other sources, helping content creators and copyright holders to protect their intellectual property.

Content Duplication:
Video platforms and content aggregators can use video similarity detection to identify and remove duplicated or near duplicated content, ensuring that their platforms offer original and diverse content to users.

Video Verification:
In cases where the authenticity of a video is questioned, similarity detection models can be used to compare the video in question with other videos to determine if it has been manipulated or altered.

Video Analytics:
Similarity detection can help content creators and marketers analyze their video performance by identifying patterns of similarity among different videos, providing insights into what types of content resonate with their audience.

Educational Content:
Video similarity detection can help educators and students discover educational videos that cover similar topics, enabling more comprehensive learning experiences.

1.6 Document Conventions

Heading 1: 16px, Bold Times New Roman
Heading 2: 14px, Bold, Times New Roman
Heading 3:12px, Bold, Times New Roman
Normal: 12px, Bold, Times New Roman

1.7 Miscellaneous

CNN: Convolution 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.

2. Technical Architecture

2.1 Application and Data Architecture

2.1.1 Application Architecture

Video editing detection:

The system employs a dual level detection method to categorize queried videos based on their editing status. This method helps in distinguishing between unedited videos, edited videos, and videos containing multiple scenes.

Descriptors:

The descriptor in the proposed video similarity detection system extracts meaningful features from videos, transforms them into compact embeddings, calculates similarity between videos, and aids in detecting manipulations within queried videos.

Matching module:

The matching module in the video similarity detection system aligns temporal segments between a query video and reference videos, measures similarity, and provides localization information for matched segments.

User-Friendly Web Page:

This is your portal to our system. It's a simple webpage where you ask our system to check videos. You must upload video and it gives you which part of video is copied or similar, making the whole process user-friendly.

2.1.2 Data Architecture

Reference Video Repository:

The data architecture includes a repository for storing reference videos used for comparison and similarity detection. These videos may be organized based on categories, tags, or other metadata to facilitate efficient retrieval and analysis.

Smart Descriptions:

These are like smart summaries of each video. They help our system quickly grasp and compare videos, making the analysis faster and more accurate.

High-Level Architecture Diagram:

The Smart Video Checker and Matching Finder are like the main cities, and the Video Library and Smart Descriptions are like the big databases connecting everything. Your interaction through the web page is like the road that connects you to these smart cities.

2.2 Component Interactions and Collaborations

Descriptor and Matching Module Interaction:

The descriptor module extracts feature from videos, which are then used by the matching module to compare and find similarities between the query video and reference videos.

Matching Module and reference videos descriptions:

The matching module collaborates with the reference videos descriptions to identify any discrepancies between the queried video and reference videos, helping to flag potential manipulations.

2.3 Design Reuse

2.3.1 Design Reuse

Utilizing existing libraries and frameworks:

The project can leverage established libraries and frameworks for tasks such as feature extraction, similarity calculation, and manipulation detection. Libraries like TensorFlow or PyTorch can be used for deep learning-based feature extraction, while OpenCV can assist in video processing tasks.

Reusing pre-trained models:

Pre-trained deep learning models for tasks like object detection or image classification can be reused and fine-tuned for specific aspects of the video similarity detection system, saving time and computational resources.

2.4 Technology Architecture

Backend Development:

Machine learning and deep learning algorithms for video similarity detection. Python libraries like scikit-learn, Keras, or TensorFlow for implementing machine learning models.

Descriptor module:

Descriptor Track, we propose feature extractor from vision transformer and Frame Scenes Detection (FSD) to tackle the core challenges on Video Copy Detection.

Matching module:

We propose a Similarity Alignment Model (SAM) for video copy segment matching.

Frontend Development:

This architecture leverages Gradio capabilities to create a seamless and responsive user interface for the video similarity detection system, all within the Python environment.

2.5 Architecture Evaluation

2.5.1 Selection of Infrastructure/Technology

Machine Learning Frameworks

TensorFlow and PyTorch:

Reason for Selection: TensorFlow and PyTorch are selected for their robust support in deep learning and extensive community support. These frameworks provide powerful tools for building, training, and deploying machine learning models, especially for video processing tasks.

Pros:

- Wide adoption and strong community support.
- Comprehensive documentation and tutorials.
- Efficient for handling large-scale deep learning tasks.
- Support for hardware acceleration (GPU and TPU).

Cons:

- Steeper learning curve for beginners.
- Can be overkill for simpler tasks.

Alternative:

scikit-learn:

Pros: Easier for quick prototyping and simpler machine learning models. Well-suited for classical machine learning tasks.

Cons: Limited in handling deep learning tasks and large datasets compared to TensorFlow and PyTorch.

Video Processing Libraries

OpenCV:

Reason for Selection: OpenCV is selected for its extensive functionalities in computer vision and image processing. It supports a wide range of operations needed for video frame extraction and manipulation.

Pros:

- Comprehensive set of image and video processing tools.
- High performance and real-time capabilities.
- Cross-platform support.

Cons:

- Can be complex to use for advanced tasks.
- Limited in handling end-to-end machine learning pipelines.

Alternative:

FFmpeg:

Pros: Powerful tool for video and audio processing, excellent for format conversion and streaming.
Cons: Less suitable for complex computer vision tasks, requires command-line proficiency.

Frontend Frameworks

Gradio:

Reason for Selection: Gradio is chosen for its simplicity in creating interactive web applications directly from Python scripts. It is particularly suitable for rapid prototyping and developing user-friendly interfaces for data-driven applications.

Pros:

- Quick to set up and deploy.
- Seamless integration with Python data science libraries.

- User-friendly for non-web developers.

Cons:

- Limited customization compared to traditional web frameworks.

- Not ideal for highly complex web applications.

Alternative:

Django/Flask:

Pros: Full-fledged web frameworks with extensive features and customization options.

Cons: More setup time and complexity, particularly for rapid prototyping.

2.5.2 Technology Architecture

Backend Development:

Reason for Selection: Utilizing Python libraries such as Keras and TensorFlow for implementing machine learning models ensures robust and scalable backend development for video similarity detection.

Pros:

- Rich ecosystem of libraries and tools.
- Strong support for deep learning and neural network-based approaches.

Cons:

Requires expertise in machine learning and Python programming.

3. Detailed/Component Design

When a user uploads a video, the system starts by checking its authenticity and similarity to reference videos. First, it looks for any signs of editing. If the video appears unedited, the system quickly extracts its features and compares them to those of the reference videos. If no significant manipulations are found, the video is confirmed as original.

However, if the video shows signs of editing, the system performs a more detailed analysis. It breaks the video into smaller segments and examines each part for similarities to the reference videos using advanced techniques like similar image detection.

If the video has multiple frames, the system analyzes each frame individually. It separates the video into single frames, extracts features from each frame, and creates digital representations (embedding) of the media. These embedding are then compared to those in the reference dataset to identify the original source and highlight potential matches. The system shows possible matches on the screen and highlights the corresponding temporal segments.

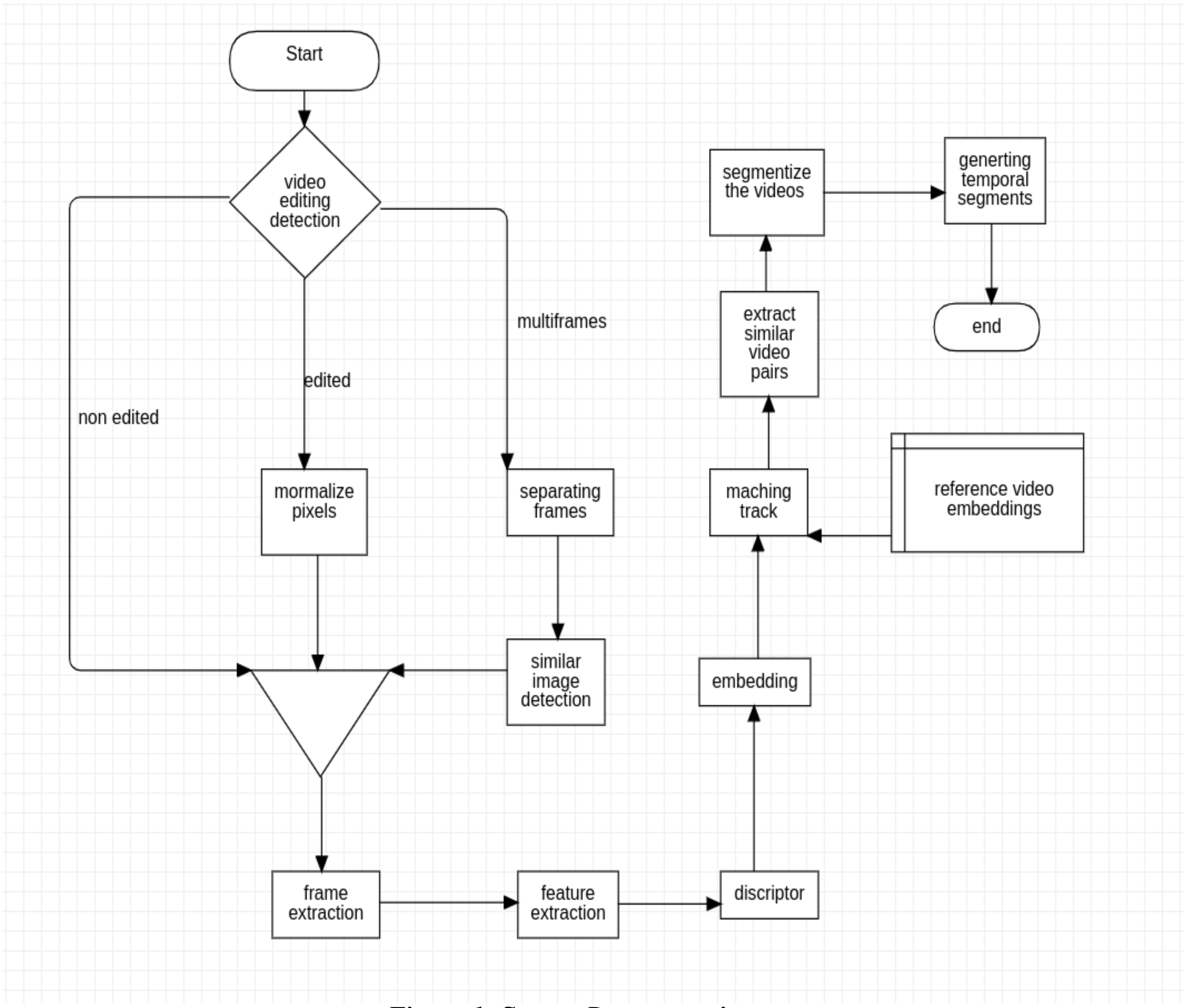


Figure 1: System Demonstration

3.1 Descriptor Module

The descriptor in the proposed video similarity detection system extracts meaningful features from videos, transforms them into compact embeddings, calculates similarity between videos, and aids in detecting manipulations within queried videos.

3.2 Matching Module

The matching module in the video similarity detection system aligns temporal segments between a query video and reference videos, measures similarity, and provides localization information for matched segments.

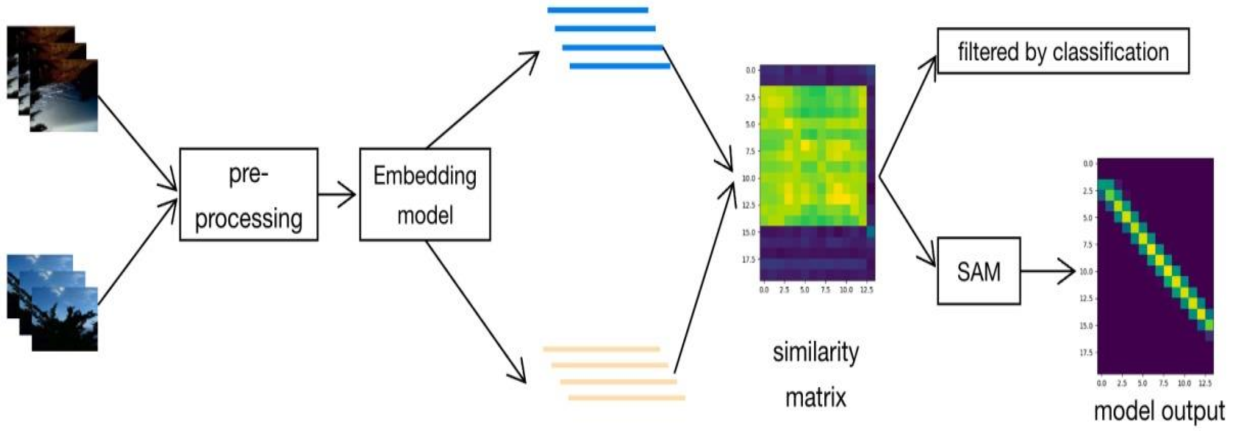


Figure 2: Matching Module

3.3 Dataset

video copy detection dataset composed of approximately 100,000 videos derived from the YFCC100M dataset which is freely available on keggle. The training dataset contains 8,404 query videos, 40,311 reference videos, and the ground truth for the query videos which contain content derived from reference videos. Edited query videos may have been modified using a number of techniques including blending. Due to lack of resources this is not being used this model is trained on few number of dataset to mimic the real world.

4. Screenshots/Prototype

4.1 Workflow

When a user uploads a video, the system initiates a process to determine its authenticity and similarity with reference videos. It begins by dissecting the video into smaller segments, examining each fragment for resemblances with videos in the reference dataset. This meticulous analysis involves employing advanced techniques, such as similar image detection, to identify any matching components within the uploaded video and the reference set.

Furthermore, if the uploaded video comprises multiple frames, the system meticulously separates and analyzes each frame individually. By scrutinizing every frame in isolation, the system ensures comprehensive coverage in detecting similarities or discrepancies between the uploaded video and the reference dataset. The video is split into single frames, from each frame the features are extracted then its embedding are made which is the digital representation of the of media. These representations are then compared from the embedding of the referenced data set and provide the original source of the video and possible potential pairs of videos are shown on the screen and temporal segments are highlighted.

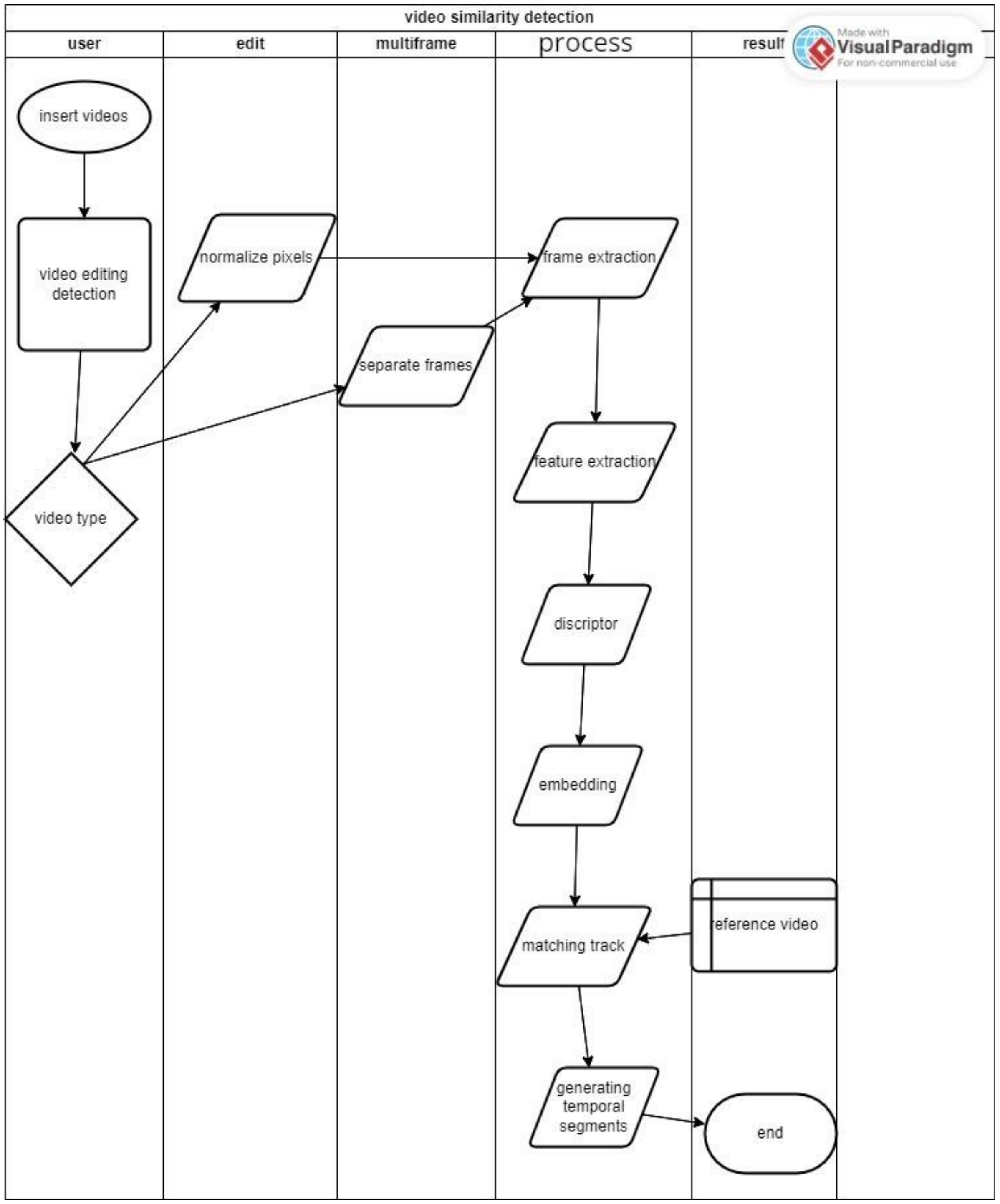


Figure 3: System Working Sequence

4.2 Screens

4.2.1 Screen 1

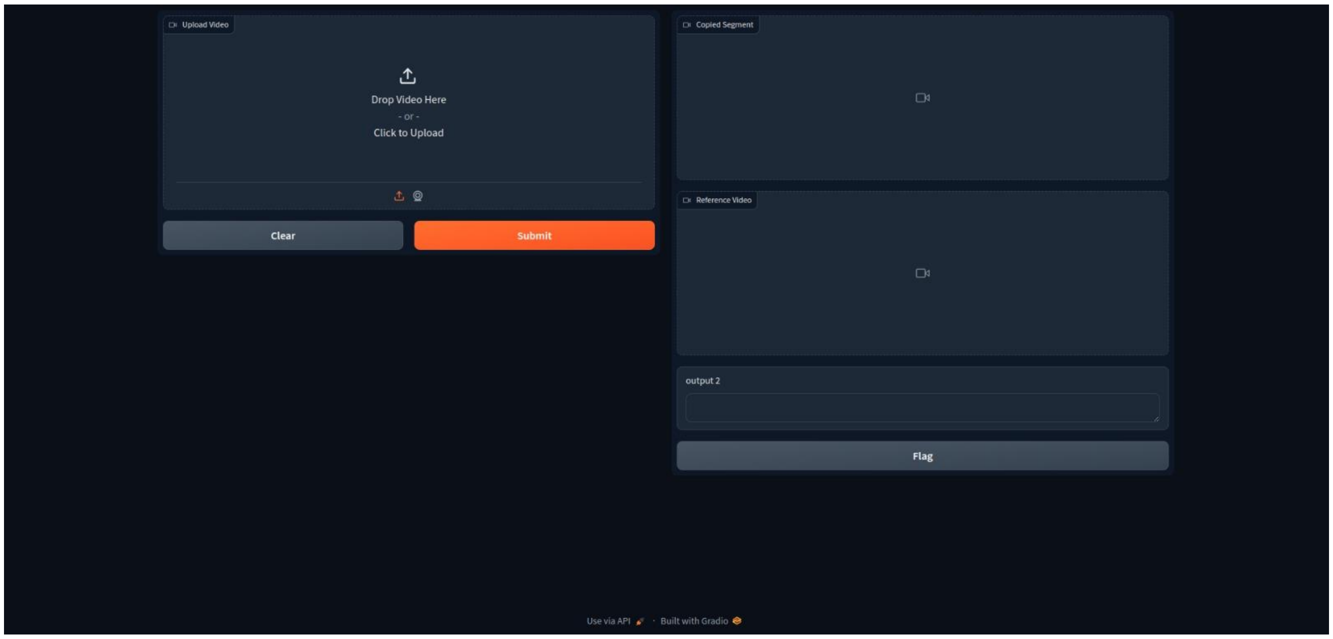


Figure 4: Prototype 1

4.2.2 Screen 2

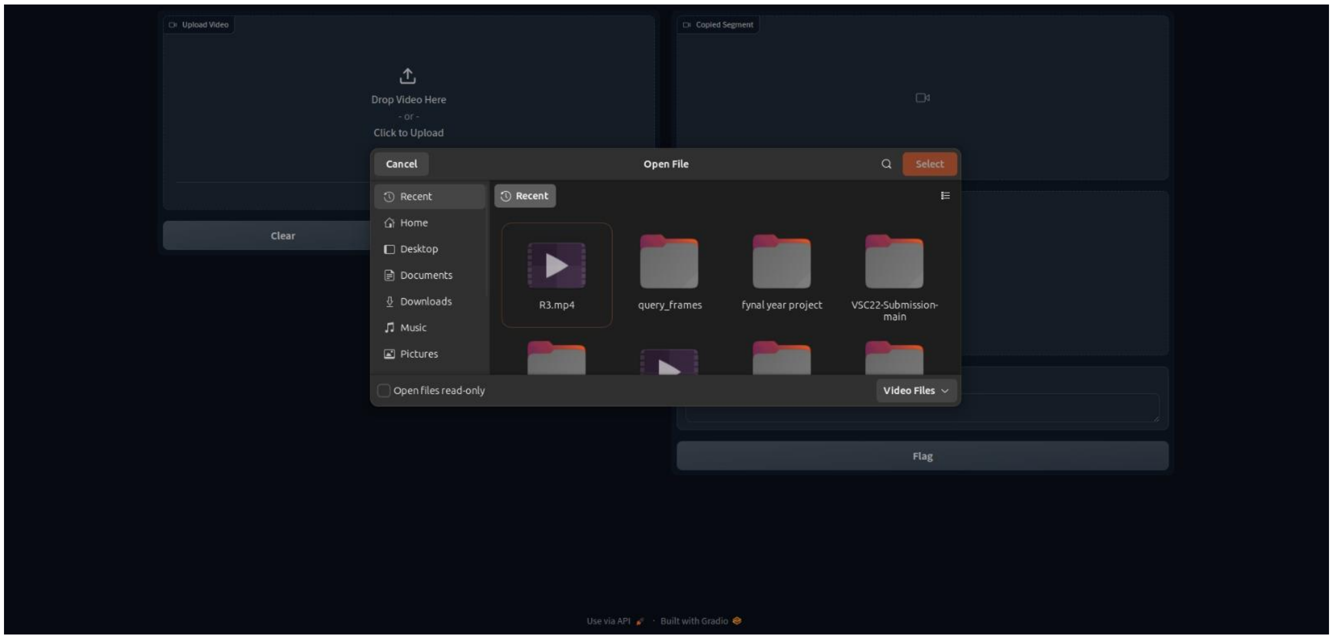


Figure 5: Prototype 2

4.2.3 Screen 3

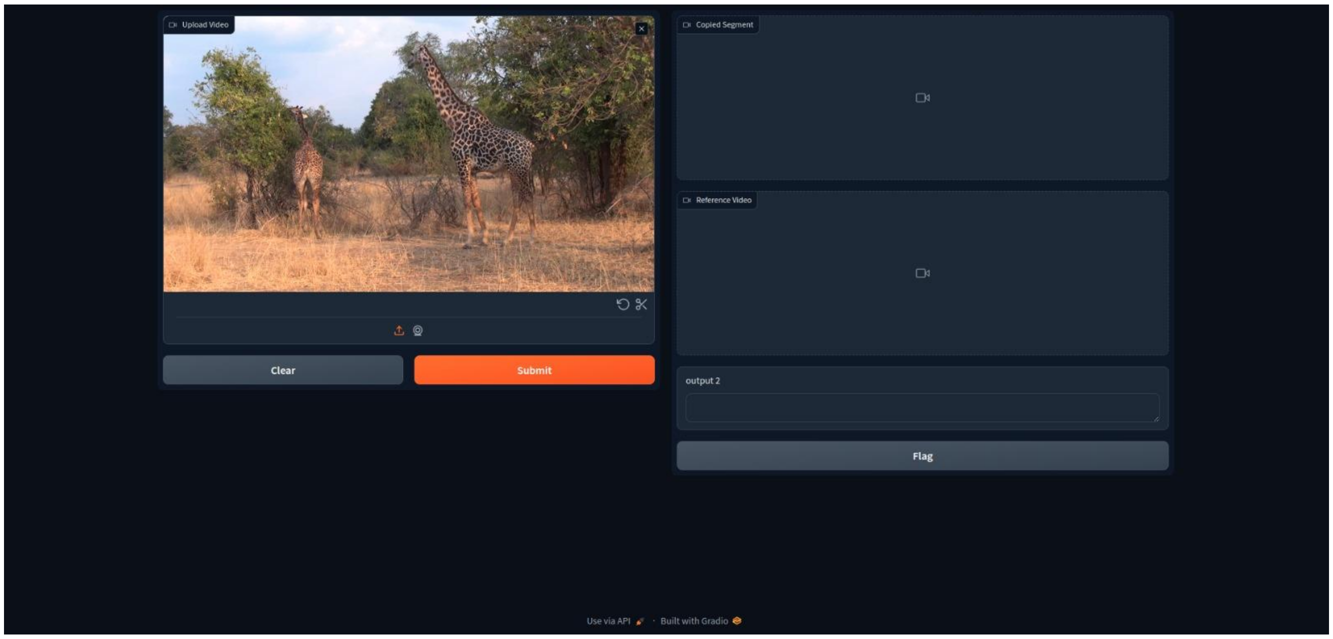
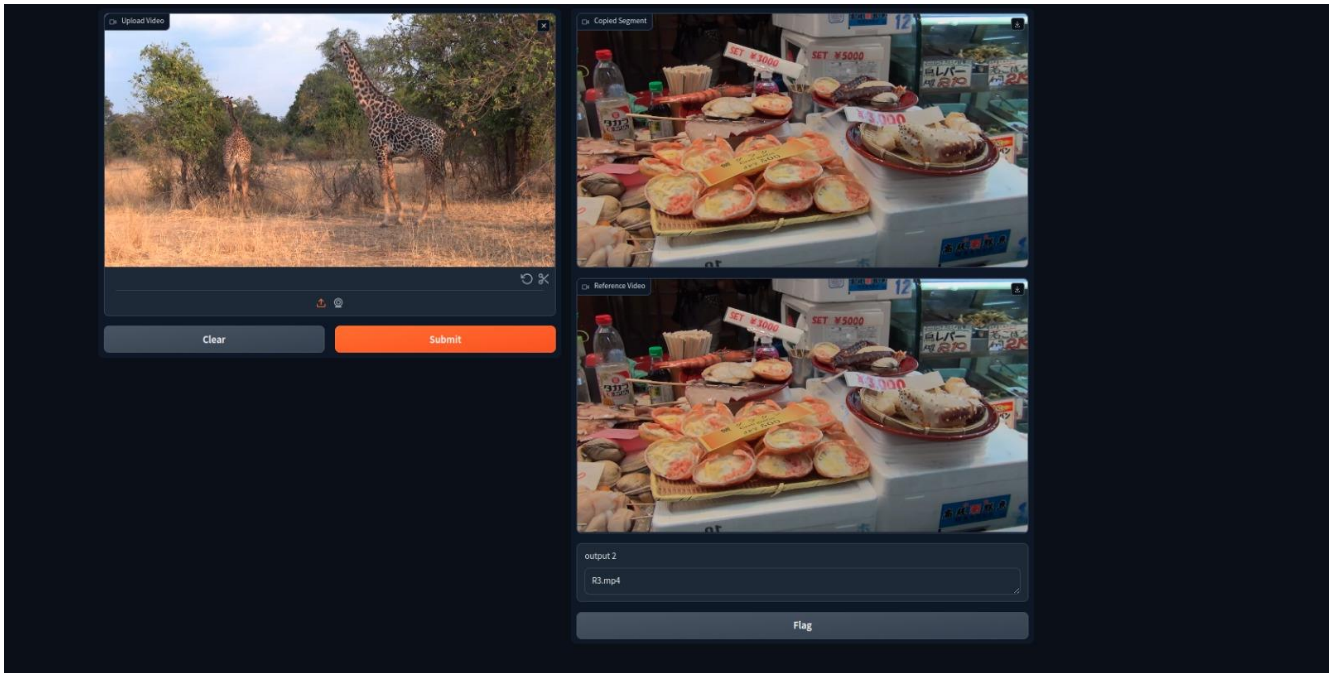


Figure 6: Prototype 3

4.2.4 Screen 4



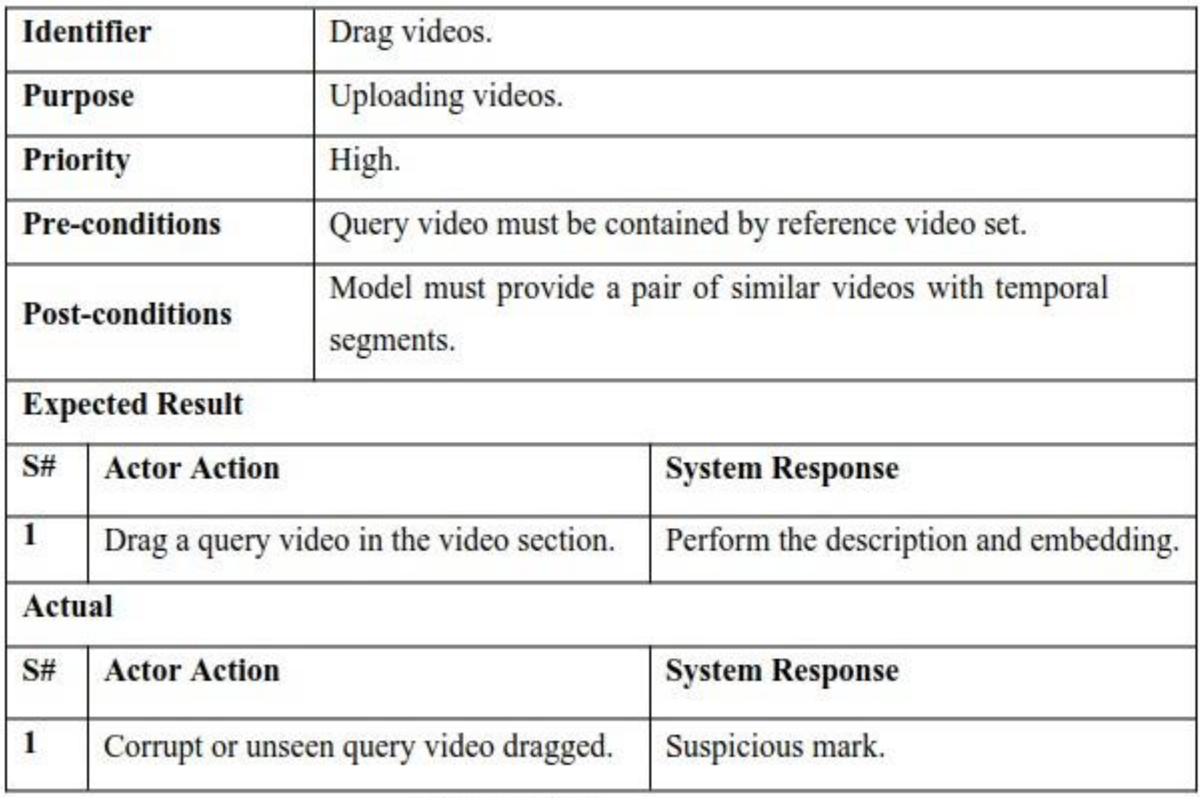
R3.mp4

Figure 7: Prototype 4

5. Test Specification and Results

5.1 Test Case Specification

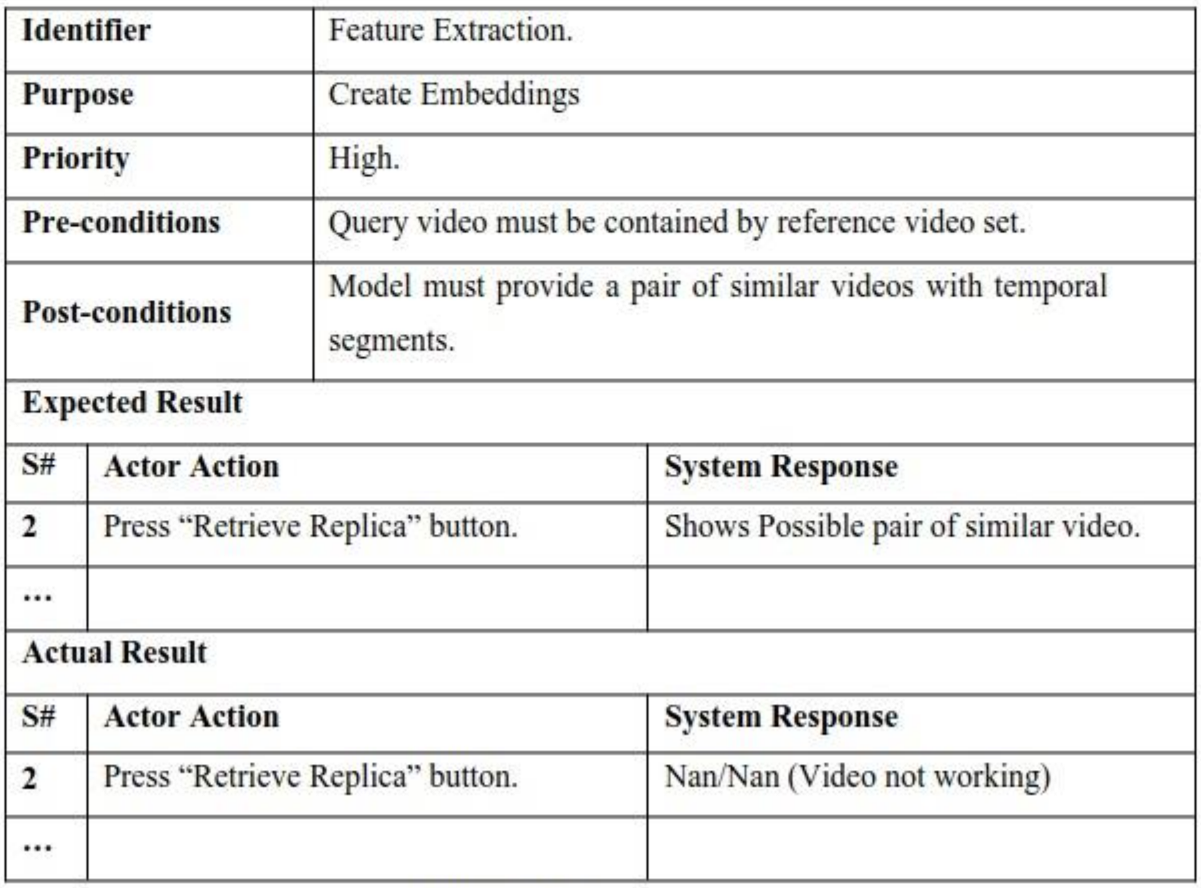
5.1.1 TC: 1



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

Table 1: TC-1

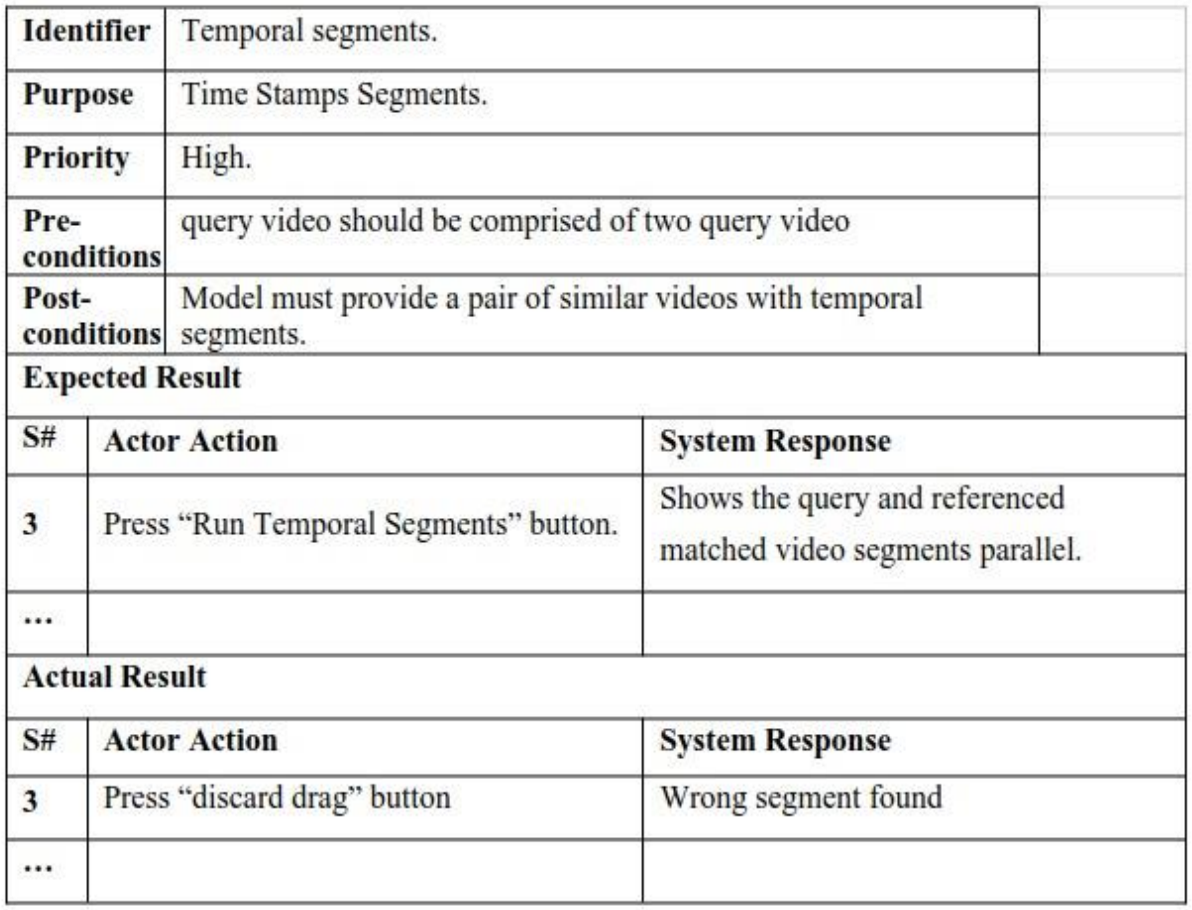
5.1.2 TC: 2



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.	
Expected Result		
S#	**Actor Action**	**System Response**
2	Press “Retrieve Replica” button.	Shows Possible pair of similar video.
…		
Actual Result		
S#	**Actor Action**	**System Response**
2	Press “Retrieve Replica” button.	Nan/Nan (Video not working)
…		

Table 2: TC-2

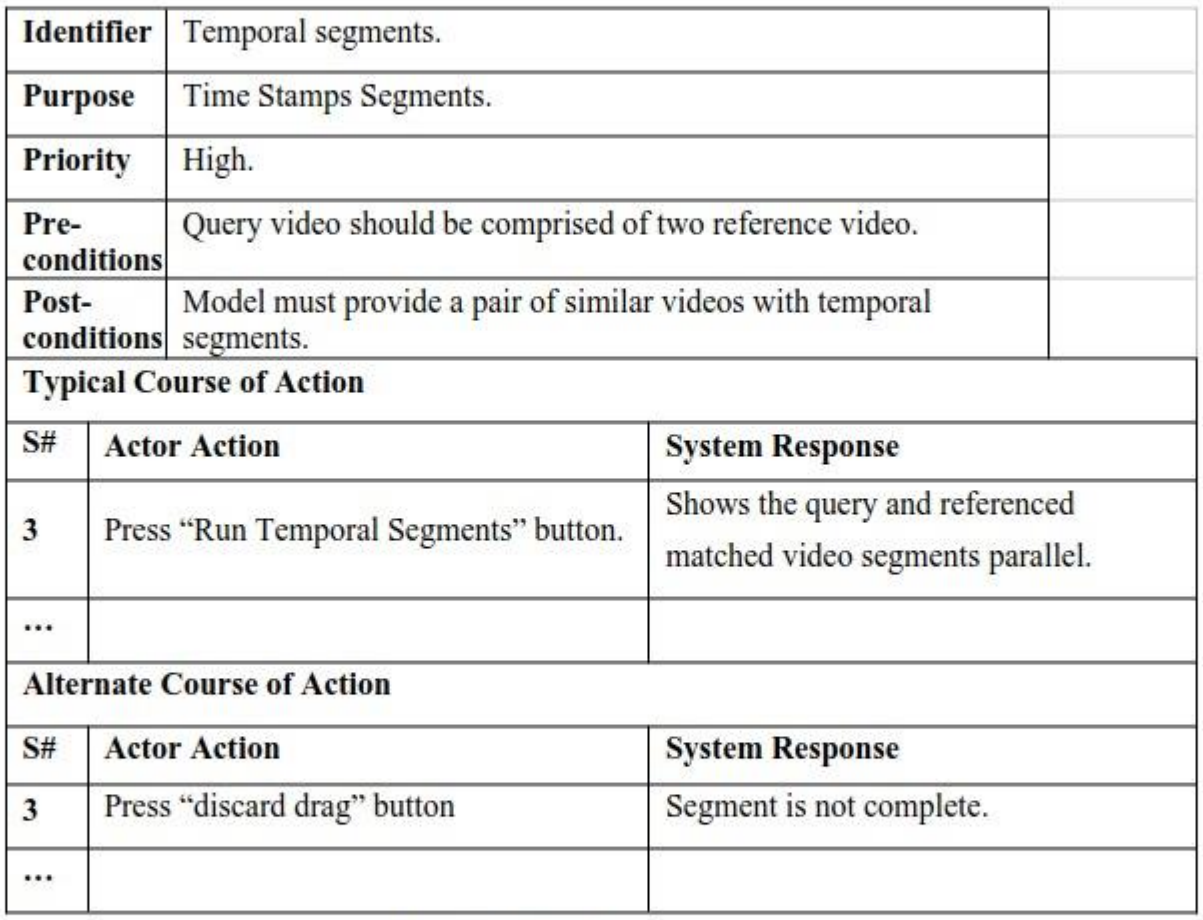
5.1.3 TC: 3



Identifier	Temporal segments.	
Purpose	Time Stamps Segments.	
Priority	High.	
Pre-conditions	query video should be comprised of two query video	
Post-conditions	Model must provide a pair of similar videos with temporal segments.	
Expected Result		
S#	**Actor Action**	**System Response**
3	Press "Run Temporal Segments" button.	Shows the query and referenced matched video segments parallel.
…		
Actual Result		
S#	**Actor Action**	**System Response**
3	Press "discard drag" button	Wrong segment found
…		

Table 3: TC-3

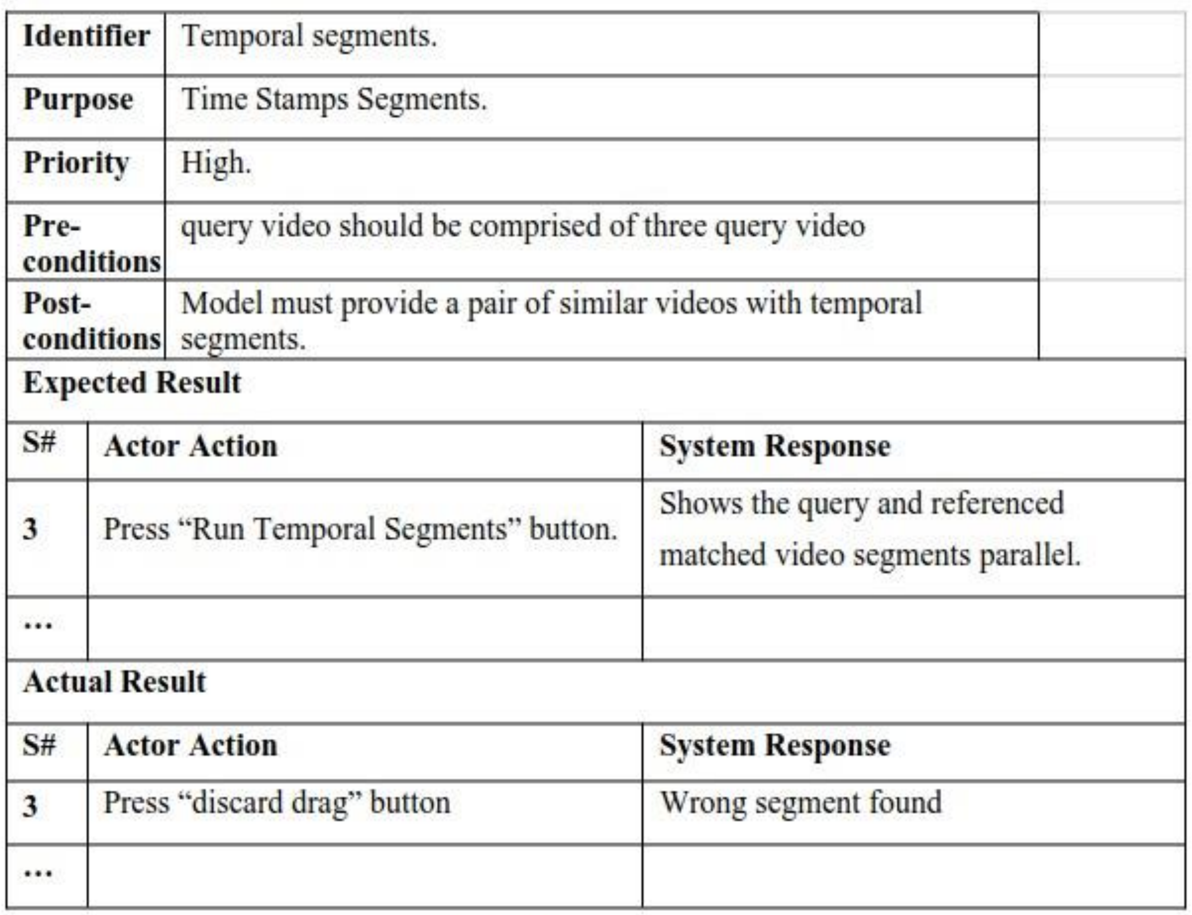
5.1.4 TC: 4



Identifier	Temporal segments.	
Purpose	Time Stamps Segments.	
Priority	High.	
Pre-conditions	Query video should be comprised of two reference video.	
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 parallel.
…		
Alternate Course of Action		
S#	**Actor Action**	**System Response**
3	Press "discard drag" button	Segment is not complete.
…		

Table 4: TC-4

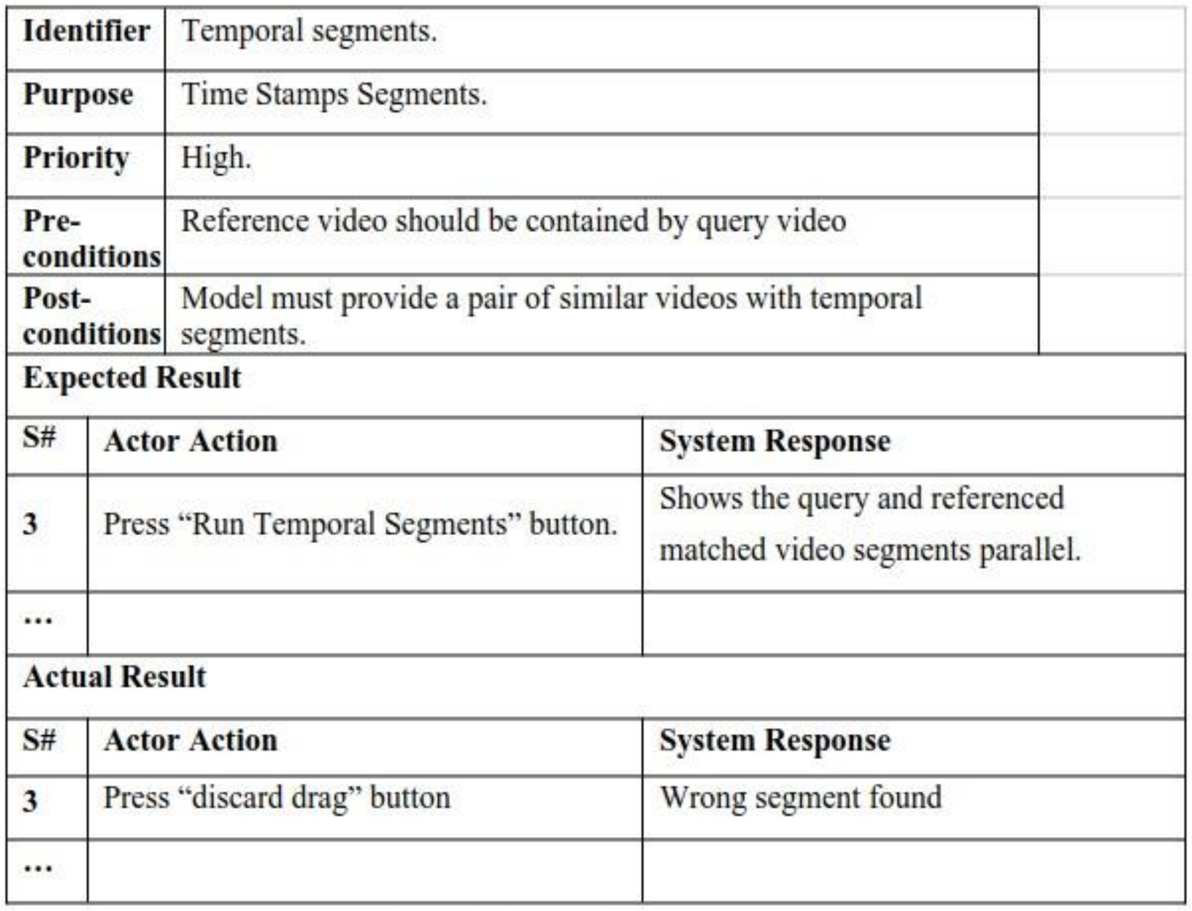
5.1.5 TC: 5



Identifier	Temporal segments.	
Purpose	Time Stamps Segments.	
Priority	High.	
Pre-conditions	query video should be comprised of three query video	
Post-conditions	Model must provide a pair of similar videos with temporal segments.	
Expected Result		
S#	**Actor Action**	**System Response**
3	Press "Run Temporal Segments" button.	Shows the query and referenced matched video segments parallel.
…		
Actual Result		
S#	**Actor Action**	**System Response**
3	Press "discard drag" button	Wrong segment found
…		

Table 5: TC-5

5.1.6 TC: 6

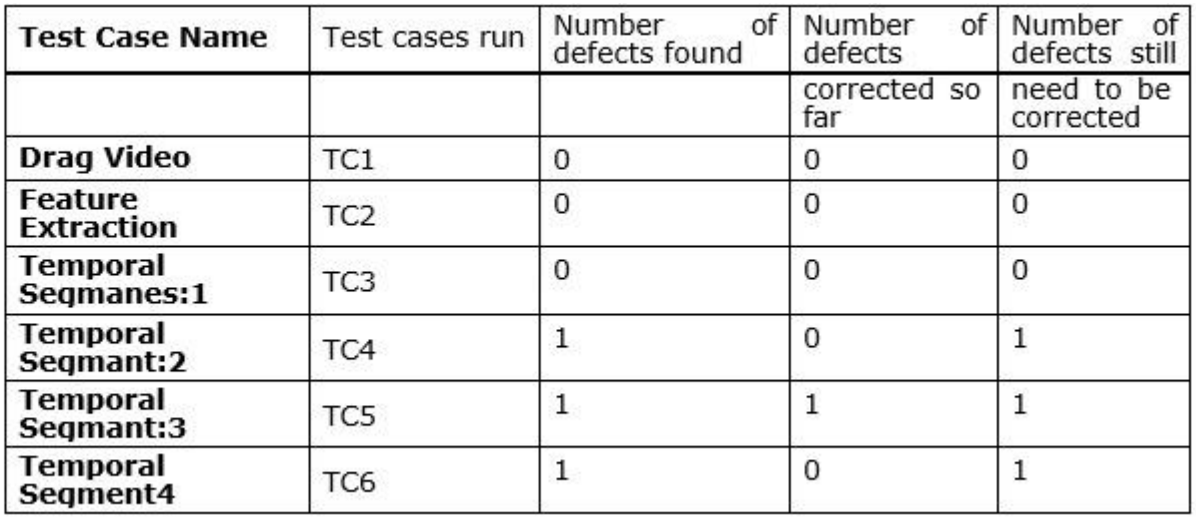


Identifier	Temporal segments.	
Purpose	Time Stamps Segments.	
Priority	High.	
Pre-conditions	Reference video should be contained by query video	
Post-conditions	Model must provide a pair of similar videos with temporal segments.	
Expected Result		
S#	**Actor Action**	**System Response**
3	Press "Run Temporal Segments" button.	Shows the query and referenced matched video segments parallel.
…		
Actual Result		
S#	**Actor Action**	**System Response**
3	Press "discard drag" button	Wrong segment found
…		

Table 6: TC-6

5.2 Summary of Test Results

6. Table 6.2: Summary of All Test Results



Test Case Name	Test cases run	Number of defects found	Number of defects	Number of defects still
			corrected so far	need to be corrected
Drag Video	TC1	0	0	0
Feature Extraction	TC2	0	0	0
Temporal Seqmanes:1	TC3	0	0	0
Temporal Seqmant:2	TC4	1	0	1
Temporal Seqmant:3	TC5	1	1	1
Temporal Segment4	TC6	1	0	1

7. Project Completion Status

Table 7.1: Project Completion Status

	Status (Complete, Partially Implemented, Not Implemented)
Module 1 (Descriptor track)	Completed
Module 2 (Matching Track)	Completed
Module 3 (Interface Connectivity)	Completed
Complete System	Completed

Table 7.2: Objective(s)/Target(s) Status

Target/Objective	**Status** (Completed, Partially Completed, Not Completed)	**Reason(s)**
Frame Extraction	Completed	Null
Frame Embedding	Completed	Null
Descriptor Database	Completed	Null
Check Similarity	Completed	Null
Retrieve Similar Video	Completed	Null
Point Matched Temporal Segment	Completed	Null
Number of Targets Completed	Frame Extraction, Frame Embedding, Check Similarity, Retrieve Similar Video, Point Matched Temporal Segment	Null
Number of Targets Partially Completed	Null	Null
Number of Targets Not Completed	Null	Null

8. Deployment/Installation Guide

1. Run git clone Video Similarity Detection.

2. Install dependencies by pip install -r requirements.txt (if necessary).

9. User Manual

Clone The Project: Run git clone Video Similarity Detection.

Install Dependencies: Install dependencies by pip install -r requirements.txt (if necessary).

Preparation and Training Model:

```
Detection Model
├── Ref
├── Lag
└── Model
```

1. Place your dataset videos in Ref folder.
2. Using Extract Database function to preprocess dataset in Lag.py file.
3. Train model on your dataset by running model.py file.
4. Save trained model.
5. Use model in further Video Detection.
6. Run the Lag.py file.

10. 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.

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

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

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

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

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

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

[9]: George Awad, Paul Over, and Wessel Kraaij. Content-basedvideo copy detection benchmarking at trecvid. ACMTransactions on Information Systems, 32(3):1–40, 2014.

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

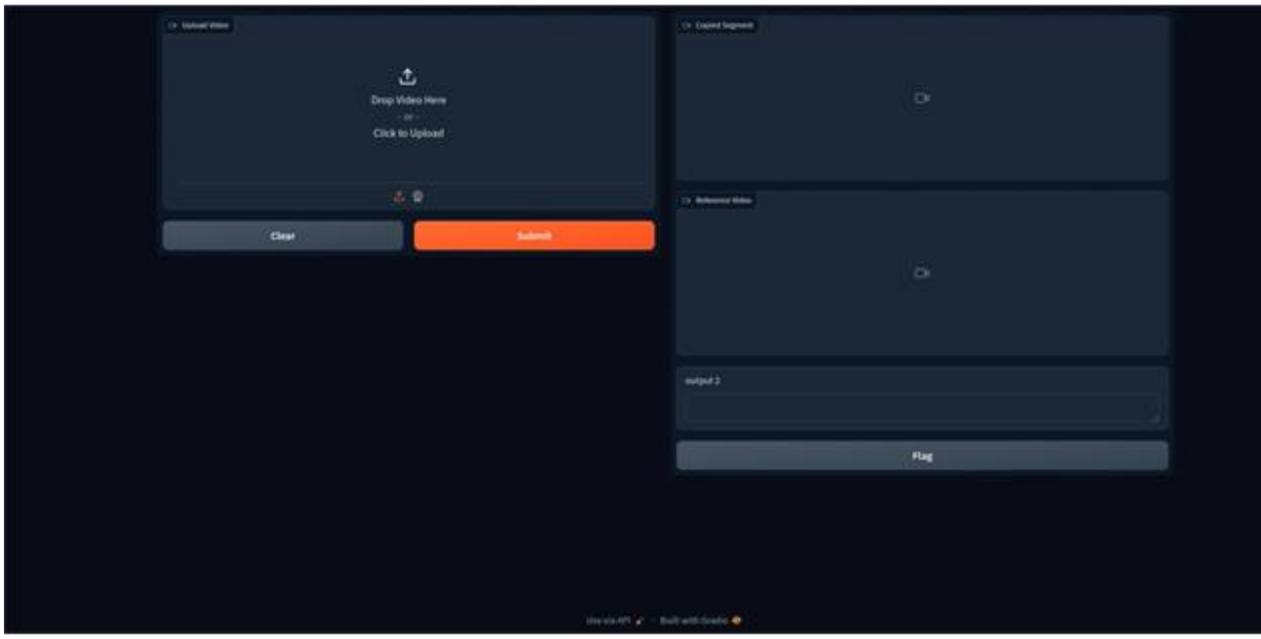
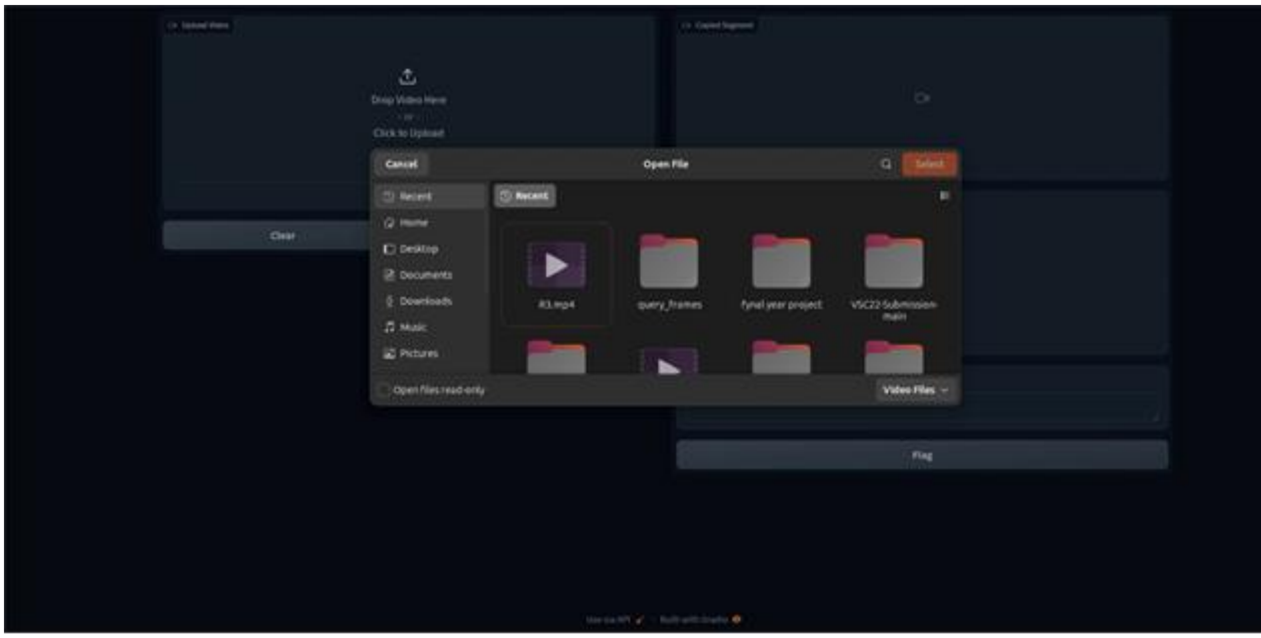
[11]:Jérôme Revaud, Matthijs Douze, Cordelia Schmid, and HervéJégou. Event retrieval in large video collections with circulanttemporal encoding. In Proc. CVPR, pages 2459–2466, 2013.

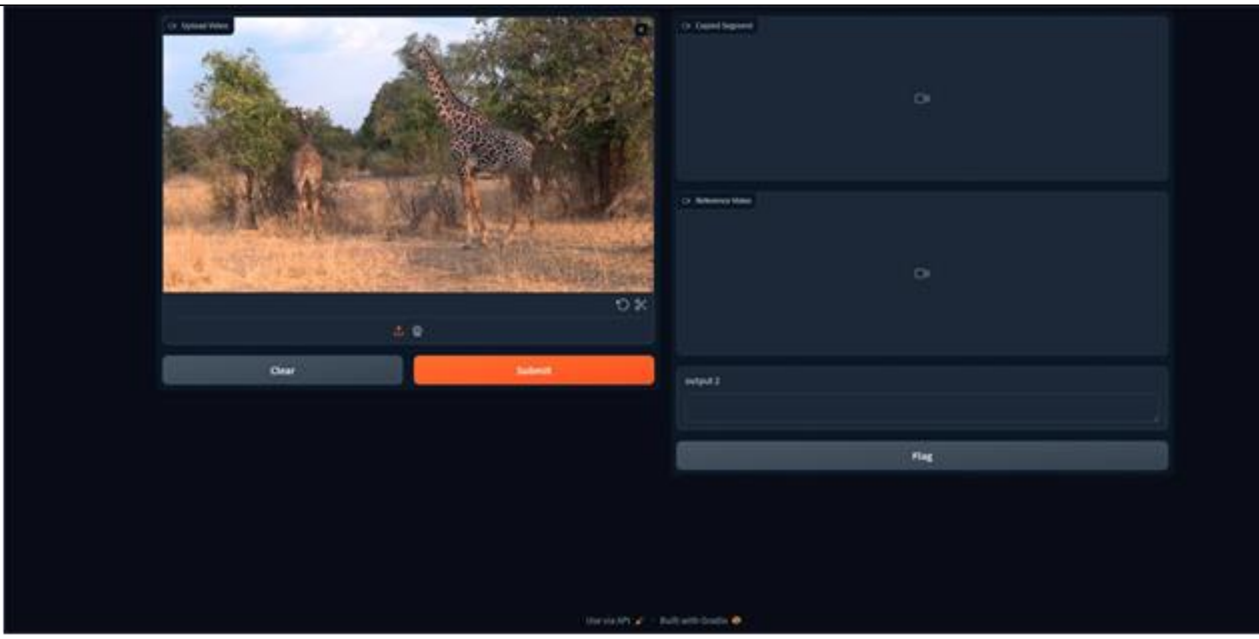
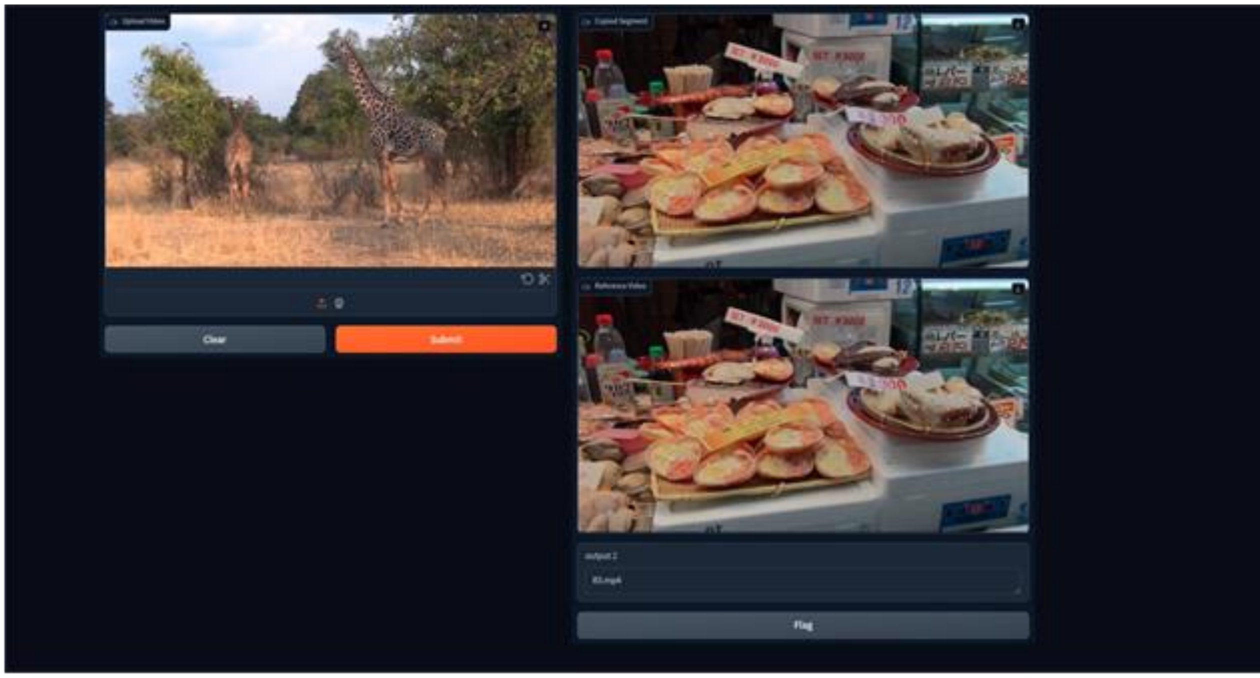
[12]: Matthijs Douze, Hervé Jégou, Harsimrat Sandhawalia,Laurent Amsaleg, and Cordelia Schmid. Evaluation of gistdescriptors for web-scale image search. In Proc. CIVR, 2009.

[13]: Matthijs Douze, Giorgos Tolias, Ed Pizzi, Zoë Papakipos, Lowik Chanussot, Filip Radenovic, Tomas Jenicek, MaximMaximov, Laura Leal-Taixé, Ismail Elezi, et al. The 2021 imagesimilarity dataset and challenge. ArXiv preprintarXiv:2106.09672, 2021.

11. Project Summary Form

Name of Project	Video Similarity Detection.
Project Type	Research
Department	FOIT
Start Date	23-06-2024
Completion Date	05-07-2024
Supervisor / Team Leader	Mohammad Ali
Team Members (if any)	Hassan Saeed, Ali Raza, Bilal Ashoob
Funding Agency (if any)	NO
Amount of Funding (if any)	No
Assign SDGs to Project	Responsible consumption and production, Industry innovation and infrastructure.
Motivation of Project	Social Media Visual Content Moderation and Control.
Practical / Potential Application	Video Similarity Detection
Abstract	The rise of internet socialization has led to the widespread sharing of text, images, videos, resulting in issues like copyright protection, duplication, misinformation, harmful content. Manual handling of these problems is costly and time-consum given the volume of daily uploads. To address this, a model is needed for accurate v copy detection with minimal resource consumption. This involves two tasks: descr tracking, where the model extracts video frames and creates embeddings for a datal and matching tracking, which compares these descriptors to identify copied segm The solution leverages the YFCC100M dataset for training, with Frame Scene Dete (FSD) and similarity alignment models (SAM) enhancing efficiency and accuracy.
Key	The project's primary target is to generate vector representations of videos using

Technical Features	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 creati 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.
Projects Images / Screenshots	 Upload Video Drop Video Here - or - Click to Upload Clear Submit Copied Segment Reference Video output 2 Flag Use via API · Built with Gradio  Upload Video Drop Video Here - or - Click to Upload Clear Cancel Open File Select Recent Home Desktop Documents Downloads Music Pictures R3.mp4 query_frames fynal year project VSC22-Submission-main Open files read-only Video Files Copied Segment Flag Use via API · Built with Gradio

	 Upload Video Copied Segment Reference Video output 2 Clear Submit Flag Use via API · Built with Gradio  Upload Video Copied Segment Reference Video output 2 Clear Submit Flag 83.mp4

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 B.1: List of non-trivial defects

This document has been adapted from the following:

1. Previous project templates at UCP
2. High-level Technical Design, Centers for Medicare & Medicaid Services. (www.cms.gov)