BSCS FINAL PROJECT

Software Design Specification

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

University of Central Punjab

Design and Test Specification

SDP Phase III

Video Similarity Detection

Advisor: Mr. Waqas Ali

Group ID: F23SE074

Member Name	Primary Responsibility
Mohammad Ali	Requirement Specification, Implementation and Documentation
Mohammad Bilal Ashoob	Requirement Specification, Implementation and Documentation
Mohammad Hassan Saeed	Requirement Specification, Implementation and Documentation

Ali Raza	Requirement Specification, Implementation and Documentation

Table of Contents

a

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, 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, not edited, second, and the third, in which frames contain 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, 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) which takes similarity matrix as an input and gives results in the score matrix.

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: 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.

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.

APIs and Services:

APIs and services facilitate communication between backend and frontend components of the system. This includes APIs for data retrieval, model inference, and result delivery.

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

Dual level detection and descriptor Module Interaction:

Form dual level detection the normalized edited videos are given to the descriptor module extracts features from videos, which are then used by the matching module to compare and find similarities between the query video and reference videos.

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 and Design Patterns

2.3.1 2.3.1Design 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.

Adopting modular architectures:

Reusing modular design architectures, such descriptors and matching tracks as microservices or component-based architectures, allows for the integration of existing components and facilitates scalability and maintainability.

2.3.2 2.3.2Desing Patterns

Observer pattern:

The descriptor module responsible for extracting features from videos could act as the subject, while the matching module that identifies similar segments could be an observer.

Adapter pattern:

Our system has a couple of components that are written in C++ language for achieving the time efficiency, the other components written in python for their interactions adapter pattern is used.

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 a dual-level detection method with Video Editing Detection (VED) 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 Streamlit's capabilities to create a seamless and responsive user interface for the video similarity detection system, all within the Python environment.

API development:

Descriptor Track, we propose a dual-level detection method with Video Editing Detection (VED) and Frame Scenes Detection (FSD) to tackle the core challenges on Video Copy Detection.

2.5 Architecture Evaluation

The architecture evaluation describes the rationale behind the selection of the specific technologies and infrastructure used in the video similarity detection system. This section discusses the pros and cons of the chosen technologies and provides a comparison with alternative options.

2.5.1 2.5.1Selection 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.

Development 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.

APIs and Services

Custom APIs for Data Retrieval and Model Inference

Reason for Selection: Developing custom APIs ensures tailored functionality and seamless integration between the backend machine learning models and the frontend user interface.

Pros:

- High degree of customization.
- Direct control over performance and scalability.

Cons:

- Requires more development effort and maintenance.

Alternative:

Third-Party APIs:

Pros: Faster implementation, reduced development effort.

Cons: Limited customization, dependency on external services.

3. Detailed 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 (embeddings) of the media. These embeddings 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.

3.1 Discriptor 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 matchedsegments.

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.

4. Screenshots/Prototype

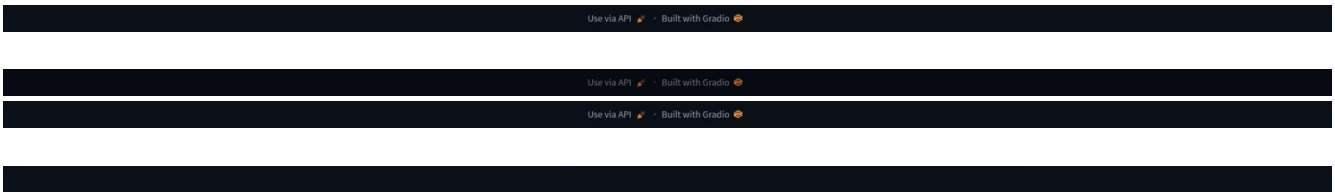
4.1 Workflow

When a user uploads a video, the system initiates a process to determine its authenticity and similarity with reference videos. Initially, it examines whether the video has undergone any editing. If the uploaded video appears unedited, the system swiftly extracts its features and compares them with those of the reference videos. If no significant deviations or manipulations are detected, the process concludes, affirming the video's originality.

Conversely, if the uploaded video shows signs of editing or manipulation, the system delves deeper into its content. 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 embeddings are made which is the digital representation of the of media. These representations are then compared from the embeddings 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.

4.2 Screens



5. Test Specification and Results

5.1 Test Case Specification

5.2 Summary of Test Results

Table 6.2: Summary of 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 Segmanes:1	TC3	0	0	0
Temporal Segmant:2	TC4	1	0	1
Temporal Segmant:3	TC5	1	1	1
Temporal Segment4	TC6	1	0	1

6. Revised Project Plan

7. References

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SDS, including acronyms and abbreviations.>

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

This document has been adapted from the following:

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