

***Dissertation on***

**“Indexing and Summarization of Sports Videos using Multi-Modal Approach”**

*Submitted in partial fulfillment of the requirements for the award of the degree*

*of*

**Bachelor of Technology**

**in**

**Computer Science & Engineering**

**UE21CS461A** **– Capstone Project Phase - 2**

***Submitted by:***

| **Krupashree MV**  **Meenal Bagare**  **Melvin Jojee Joseph**  **Naveen Kumar Reddy G** | **PES2UG21CS242**  **PES2UG21CS289**  **PES2UG21CS294**  **PES2UG21CS324** |
| --- | --- |

*Under the guidance of*

| **Dr. Sandesh B J**  Professor & Chairperson  PES University |
| --- |

**June - Nov 2024**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

FACULTY OF ENGINEERING

**PES UNIVERSITY**

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

Logo, company name

Description automatically generated

**PES UNIVERSITY**

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

**FACULTY OF ENGINEERING**

**CERTIFICATE**

*This is to certify that the dissertation entitled*

**‘Indexing and Summarization of Sports Videos using Multi-Modal Approach’**

*is a bonafide work carried out by*

| **Krupashree MV**  **Meenal Bagare**  **Melvin Jojee Joseph**  **Naveen Kumar Reddy G** | **PES2UG21CS242**  **PES2UG21CS289**  **PES2UG21CS294**  **PES2UG21CS324** |
| --- | --- |

In partial fulfillment for the completion of sixth-semester Capstone Project Phase - 2 (UE21CS461A) in the Program of Study -Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period June. 2024 – Nov. 2024. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 6th-semester academic requirements in respect of project work.

| Signature  Dr. Sandesh B J  Chairperson | Signature  Dr. Sandesh B J  Chairperson | Signature  Dr. B K Keshavan  Dean of Faculty |
| --- | --- | --- |

**External Viva**

| **Name of the Examiners**  **1.** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **2.** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **Signature with Date**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| --- | --- |

**DECLARATION**

We hereby declare that the Capstone Project Phase - 2 entitled **“Indexing and Summarization of Sports Videos using Multi-Modal Approach”** has been carried out by us under the guidance of **Dr. Sandesh B J, Professor & Chairperson** and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester June – Nov. 2024. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

| **Krupashree MV**  **Meenal Bagare**  **Melvin Jojee Joseph**  **Naveen Kumar Reddy G** | **PES2UG21CS242**  **PES2UG21CS289**  **PES2UG21CS294**  **PES2UG21CS324** |  |
| --- | --- | --- |

**ACKNOWLEDGEMENT**

I would like to express my gratitude to Dr. Sandesh B J, Professor and Chairperson, Department of Computer Science and Engineering, PES University, for his continuous guidance, assistance, and encouragement throughout the development of this UE21CS461A Capstone Project Phase-2.

I am grateful to the Capstone Project Coordinators, Dr. Farida Begam, Professor and Prof. Vandana M. Ladwani, Associate Professor, Department of Computer Science and Engineering, PES University for organizing, managing, and helping with the entire process.

I take this opportunity to thank Dr. Sandesh B J, Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support I have received from the department. I would like to thank Dr. B.K. Keshavan, Dean of Faculty, PES University for his help.

I am deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor, PES University, Dr. Suryaprasad J, Vice Chancellor, PES University and Prof. Nagarjuna Sadineni, Pro Vice Chancellor, PES University for providing to me various opportunities and enlightenment every step of the way. Finally, this project could not have been completed without the continual support and encouragement I have received from my family and friends.

**ABSTRACT**

Traditional methods of sports summarization heavily depend on large teams performing manual editing, where enormous portions of game footage are picked over through by humans to select the most crucial moments for the compilation of highlights. This is, however, a time-consuming, resource-intensive, and leads to uneven-coverage of events. Our work proposes a novel, multi-modal approach to sports summarization, in which we combine multiple modalities: Twitter data, audio features, and video content to automate and enhance the entire process of sports summarization. This approach is based on the integration of diverse sources of information, which should streamline the summarization process and hence enhance efficiency in covering sports events. This methodological novelty has the potential to transform sports summarization into a scalable and efficient solution, delivering engaging and informative highlights to sports enthusiasts from all over the globe.

**TABLE OF CONTENTS**

| **Chapter No.** | **Title** | **Page No.** |
| --- | --- | --- |
|  | **INTRODUCTION** | **01** |
|  | **PROBLEM STATEMENT** | **03** |
|  | **LITERATURE REVIEW** | **04** |
|  | **PROJECT REQUIREMENTS SPECIFICATION** | **08** |
|  | **SYSTEM DESIGN** | **23** |
|  | **PROPOSED METHODOLOGY** | **30** |
|  | **IMPLEMENTATION AND PSEUDOCODE** | **37** |
|  | **RESULTS AND DISCUSSION** | **45** |
|  | **CONCLUSION AND FUTURE WORK** | **52** |
| **REFERENCES/BIBLIOGRAPHY** | | **53** |
| **APPENDIX A DEFINITIONS, ACRONYMS AND ABBREVIATIONS** | | **55** |
|  | |  |

**LIST OF FIGURES**

| **Figure no.** | **Figure Title** | **Page No.** |
| --- | --- | --- |
| **5.1** | **High Level Architecture Design Diagram** | **23** |
| **5.2** | **Master Class Diagram** | **26** |
| **5.3** | **Use Case Diagram** | **28** |
| **5.4** | **Deployment Diagram** | **29** |
| **8.5.1** | **Homepage** | **50** |
| **8.5.2** | **Generating Summary** | **50** |
| **8.5.3** | **Successfully Generated Summary** | **51** |

**LIST OF TABLES**

| **Table no.** | **Title** | **Page No.** |
| --- | --- | --- |
| **8.1.1** | **Goal Detection by Twitter** | **45** |
| **8.3.1** | **Audio Modality Metrics Table** | **47** |
| **8.4.1** | **Multi-modal Metrics Table** | **48** |
| **8.4.2** | **Multi-modal Goal Ratio** | **49** |
|  |  |  |

**CHAPTER 1**

**INTRODUCTION**

Sports Video summarization is the condensing of lengthy sports videos into shorter summaries that capture the important events of a game. In an age where people do not have the time to sit for hours together to watch their favorite game, this can help viewers efficiently enjoy the entire game in very little time.

Traditional approaches to summarizing sports videos rely heavily on manual efforts by video editors who sift hundreds of hours of video footage from multiple camera angles and positions in order to identify the significant events and then manually edit this into a highlights video. This process is not only expensive but also time consuming. It also lacks an element of audience engagement. However, this approach can be made better by leveraging the advancements in deep learning, computer vision and natural language processing. Furthermore, the audience reactions can be captured from various social media platforms to help create a better experience for the sports viewers.

The goal of this project is to build a system that can efficiently summarize sports videos by using the advancements in deep learning, computer vision and natural language processing. Further, this project also aims to capture user sentiments towards significant events in the sports videos by analyzing the tweets related to the game, thereby enabling us to effectively capture audience reactions and insights to create more engaging highlights for the viewer.

The core of this project involves a multi-modal approach that seamlessly integrates multiple modals of data - visual, audio and textual in addition to the twitter data of that game. This ensures that our system is accurate and effective in generating summaries of sports videos and also adds audience engagement and insights into our video summarization process.

Audio Analysis is crucial for our system in order to help identify events based on valuable information obtained from the audio track such as crowd intensity, commentator fervor, and player reactions, adding the summaries with a deeper layer of context and emotion.

Visual Analysis plays a key role in ensuring that no event goes unnoticed. The system will analyze the visual element which is scoreboard to identify pivotal moments and highlights within the videos.

Textual Analysis is performed both on the commentary data of the game as well as on the twitter feed provided. This will ensure that there is a human element in summaries generated. The commentary is analyzed using a large language model to classify the event as significant or not significant.

By incorporating various modalities into our project our aim was to create a superior system than the traditional systems being used to generate highlights and additionally by adding audience reactions and insights from twitter we aim to generate a more enriching summary of the sports video.

**CHAPTER 2**

**PROBLEM STATEMENT**

The traditional approach of producing sports highlights is normally time consuming and the teams involved in the editing process need to put many hours in browsing through the overall in-game data and going through the diversity of possible moments and looking for those ones with the greatest potential to be included in the video summary.

The existing approaches of video summarization rely on basic algorithms that fail to capture the dynamic energy and excitement of the game. They lack the integration of content from social media platforms, which could enrich the summarization process with real time insights and reactions from the audience.

Our project addresses this challenge by proposing a multi-modal approach, leveraging the power of Twitter data, audio features, and video content to automate and enhance the summarization process.

This process enhances the accessibility and usability of sports content and it also improves user engagement and satisfaction and it provides analysis by identifying key moments through audio, video and twitter analysis to extract details from those moments.

It provides user-driven highlights based on detected events and user preference (favorite teams, player names, event types). It provides concise and engaging highlights based on user choice.

Overall, it redefines the traditional method of sports content creation, providing a platform for dynamic and engaging content generation and empowers sports enthusiasts to stay connected with the latest sports events efficiently and effectively.

**CHAPTER 3**

**LITERATURE SURVEY**

The aim of the research paper [[1]](#_heading=h.vr9u3b7msasy) by Kaito Hirasawa, Keisuke Maeda,Takahiro Ogawa, and Miki Haseyama is to propose a method for the detection of important scenes in baseball videos by leveraging both Twitter data and video analysis in order to generate highlights automatically. They have proposed a method termed, Time-Lag-Aware Multimodal Variational Autoencoder (Tl-MVAE) that considers time-lags between tweets and multiple previous events in the video to improve the accuracy of scene detection. The Tl-MVAE comprises an encoder, decoder and an important scene detector. In the encoder architecture, the features extracted from tweets and videos are transformed into latent features and this is where they include the consideration of time-lags. The time-lags represent the influence of past events on tweets, assuming a Poisson distribution where events from present to past influence tweets. The decoder reconstructs the original data from the latent features, its aim is to closely resemble the original features extracted from tweets and videos. The important scene detector consists of multiple fully connected layers and takes the latent features from the encoder as input. Its objective is to classify scenes as important or normal. A predetermined threshold is used to determine if a scene is important based on the calculated probability. In order to evaluate the performance of the proposed model, they made use of specificity recorded to be 0.409.

The aim of the paper [[2]](#_heading=h.vr9u3b7msasy) is that A new approach toward summarizing football match videos through advanced technologies tries to secure the end-users' accessibility and experience in consuming video content.The paper proposes a new approach toward video summarization with the integration of deep neural networks and semantic mapping techniques. This new approach will alter the perspective of how videos can be summarized more effectively. The method may heavily depend upon the

availability and quality of the training data. In case of small and biased training data, the method is likely to produce worse performance or biased summaries.

The aim of the paper [[3]](#_heading=h.vr9u3b7msasy) is to leverage large language models to aid in the generation of sports summaries. The approach mentioned here uses the YOLO model in order to split the video into 20 second segments based on the bowler’s position and later feeds the commentary data into a large model that is trained to identify events based on words used for each event. The approach used in this paper has an accuracy of roughly 97% which has been achieved due to the large corpus of words that has been identified for each type of event. The use of BERT that has been fine tuned for commentary data has also aided in achieving such a high level of accuracy. Misclassifications occur when the game’s bigger picture is being described by the commentator, which varies from the actual ball actions, like wickets and boundaries. Sometimes the ball actions might come to an abrupt end, which creates a discrepancy between the extracted commentary and the actual event sequence. The commentary’s temporal misalignment might also cause misclassifications.

The aim of paper [[4]](#_heading=h.vr9u3b7msasy) is that A novel approach weighted dynamic heartbeat graph for detection events from twitter stream.The text stream was systematically transformed into a series of temporal graphs. These graphs inherited temporal frequencies and co-occurrence relationships of the words appearing in the text stream. Each graph was further used to extract a heartbeat score using two features: growth factor and aggregated centrality. A rule-based classifier labeled the graphs as event candidates. Multiple event candidates were merged to extract a list of ranked topics. For the performance evaluation of the proposed approach, three benchmarks FA Cup, Super Tuesday, and the US Election were used.

The aim of the paper [[5]](#_heading=h.vr9u3b7msasy) is to recognize and clip crucial occurrences in a cricket match by considering event-based attributes. The dataset used in this paper has been gathered from YouTube and Hotstar. The approach followed in this paper involves identifying key frames based on energy levels and then the features are extracted using CNNs(VGG16 and ResNet50). The features vectors are fed to an LSTM network to obtain word embedding and caption for each ball. The advantages of this paper is that it categorizes scenes in matches accurately, also this model covers a wide range of events including fours, sixes, wickets and player celebration. The limitations of this paper is that the ResNet model used for feature extraction requires significant processing power.

The aim of paper [[6]](#_heading=h.vr9u3b7msasy) Video summarization techniques were described, turning long videos into those with only important events.The process of video summarization involved the deep semantic features wherein the original video was cut into parts, and only relevant features were extracted from those parts.In conjunction with other techniques like DPP (Determinantal Point Process ) that allow frame selection to provide diversity, LSTM approaches seem to be particularly highlighted.Limitations of activities include the requirement for large and diversified training datasets and the slow manual annotation process and the changes of video content over time.Future directions mention its potential application in sports to automate match highlights and filter video rightly.

The aim of paper [[7]](#_heading=h.vr9u3b7msasy) is to present a high-accuracy framework to automatically clip the sports video stream by using a three-level prediction algorithm based on two classical open-source structures, YOLO-v3 and OpenPose. The research paper focuses on racquet turn based events. The three-level prediction algorithm comprises low level prediction where a boolean decision for each player engaged in “playing” action or not is taken. The middle level prediction involves a boolean decision for every frame. It captures temporal aspects of player actions within the video, The high level prediction is to counter the errors that may arise in the low level prediction. The boolean decision is taken over a short video segment. In this paper, each frame in the short video segment represents whether there is a playing action or not. Confidence levels are calculated for each frame and if it crosses a certain threshold value, the segment is classified as “playing”.

The aim of paper [[8]](#_heading=h.vr9u3b7msasy) is to develop automatically detects and summarizes important events in cricket match.This model uses techniques like optical character recognition(OCR),sound detection, and replay detection to extract important events such as boundaries and wicket.This method combine both CNN and OCR for event detection and it gives higher accuracy compare to other techniques.The system splits up video into separate shots,detect keyframes using audio cues,recognizes scoreboard information using OCR, and generates highlights based on detected events.

# CHAPTER 4

**PROJECT REQUIREMENTS SPECIFICATION**

# Introduction

# This chapter outlines the requirements for the development of a sports video summarization , leveraging multi-modal data analysis techniques. The system aims to enrich the experience and comprehension of sports content by extracting key events from videos through the analysis of Twitter streams, Video Analysis , and Audio Analysis.

# Project Scope

**Project Description:**The development of a summary and indexing system for sports videos employs multi-modal data analysis techniques: the analysis of peak activity in Twitter streams, Computer Vision, and Audio Analysis. In short, the system focuses on automatically extracting key events from sports videos, generating concise summaries so sports content can be better made accessible, enjoyable, and understood by enthusiasts, analysts, and broadcasters alike.

**Objective:** The objective of the project is to ease the pain of browsing and watching long sports videos by offering users an efficient and informative summarization tool that makes consumers go through key moments quicker with content discovery facilitated.

**Benefits:**

* Access and availability of sports content.
* Saves a lot of time for the user because it presents the summarized version.
* Enables easy navigation and discovery of the content.
* Facilitates deep analysis and understanding of sport events.
* Improves engagement and satisfaction among users.

**Objectives and Goals:**

1. Make algorithms for multi-modal data analysis, including peak activity in Twitter streams, Computer Vision and Audio Analysis.
2. Develop an accessible interface for the summarization system through human-computer interaction.
3. Extract key events from sports videos based on multi-modal insights.
4. Generate short and informative video summaries highlighting key moments.
5. Analyze the performance of the system by its metrics and user feedback
6. Scalability and adaptability toward different sports and events.

**Coverage:**

* This encompasses a wide range of sports and events, including but not limited to football, cricket, and basketball.
* The proposed system's overarching goal is the analysis of multi-modal data sources; that is, the peak activities occurring in Twitter streams, audio comments, and visual cues coming from sports videos.

**Limitations:**

* Therefore, the system may be less likely to generalize to a different sport since it consists of various rules for each sport and has diversity in conditions for different activities.
* Hence, the system may often fail to correctly identify the key events of a sports event scenario or in an especially fast-paced action scenario.
* The performance will depend on the quality and availability of sources such as Twitter streams and audio commentary.
* Sport events being very complex and diverse, the summarization of the system might be adversely affected.
* It may sometimes require human intervention or validation to be certain of its correctness in generating the summaries.

# Product Perspective

# Product Features

* **Scoreboard Analysis:**

Extracts and gets scoreboard information to provide key events into the match's progress.

* **Video Summarization:**

Uses video processing techniques to generate small summaries of key moments and highlights.

* **Multimodal Analysis:**

Merges information from video, text, and scoreboard sources for analysis.

# User Classes and Characteristics

* **Casual Viewers**: These users are interested in and aware of cricket, but probably would not care much about the nitty-gritty details of the game. They would prefer the gist or a summary where they would like to see wickets, boundaries, and key moments of an innings.
* **Excited Fans**: These people are passionate about the game, cricket, and might even have some knowledge about the game. They are bound to search for summaries in addition with more inputs like player analysis, strategic considerations, and match in-depths.
* **Coaches and players :** The video summarization is used both by coaches and players for analyzing performance as well as planning in terms of tactics. They demand strict summaries, where there are complete details of player performances and strategies and areas calling for improvement.
* **Media Professionals:** Journalists, broadcasters, and others who create content will need video summarization of cricket for news articles, match reports, and highlight reels. Such persons require video summary content that is very informative and has good visuals along with the statistics that will create such content.

# Operating Environment

* **Hardware Platform**: Compatible with standard desktop computers, laptops, and mobile devices.
* **Operating System**: Supports multiple operating systems, including Windows, macOS, and Linux.
* **Software Components**: Requires web browsers with modern HTML5 and CSS3 support for the user interface.

# General Constraints, Assumptions and Dependencies

* **Legal Problems**:All of the data protection laws, copyright law, and terms of services from social media platforms such as Twitter will be observed so that no legal problem would arise.It has to get any kind of permissions or licenses in the usage of any copyrighted content such as the sports broadcast.
* **Restrictions in Use:** Actual sports data feeds would depend upon the availability of such real-time sources, like Twitter, as well as a live video stream. There could be third-party APIs or data sources that could be restrictive of such an application to consumption for usage such as with regards to rate limiting, and data access.
* **Assumptions of the Project:**Availability of Sufficient Valid Online Data Sources: The existence of proper and authentic online real-time data sources for analysis and summarization.Format and Structure of Various Data Sources: Fair consistency in formats and structures of different data sources for processing.

# Risks

* **Resource Constraint:** The system might not scale well due to the lack of computational resources.
* **Availability of data from outside sources:** Data availability relying on outside sources raises some connected risks of reliability and access of data.
* **Technical Complexity:** Advanced techniques are associated with technical complexities and limitations.
* **Regulatory Compliance:** Failure to comply with legal procedures also may pose legal risks and implications on the organization.

# Functional Requirements

* **Automated Data Analysis:** The system automatically identifies the key event in sports data through event extraction during high twitter activity and cross-verifies with commentator excitement,audience cheers and incorporates real time scoreboard data.
* **Efficient Data Processing:** Automated large quantities of data in real time, dynamically adjusting capacities to handle increased accumulation of twitter data during high activity.
* **Noise Processing:** Successfully deals with the removal of noise and still ensures accuracy in the detection of key events.
* **Highlight Generation:** The system would automatically select and compile together the most exciting or interesting moments in the game as a highlight video.

It should get a high accuracy in detecting those key moments within the game video.More fast processes compared to human techniques. Ideally, it should produce summaries close to real time, especially for live games

# External Interface Requirements

# User Interfaces

* The system should have a user-friendly graphical user interface (GUI) that can be accessed through a web or a desktop application.
* Styles of formats on the required screens will need to strictly adhere to standards in GUI styles both for consistency and friendliness toward a user.
* The screen layout should be intuitive; that is, it should have standard functions such as navigation menus, a search bar, and help documentation.
* Error messages will be clear, brief, and specific and allow the user to debug his problem.

# Hardware Requirements

* The system shall comply with normal computing hardware; desktops, laptops, and servers.
* It must support all types of input devices, including keyboards, mice, touchscreens, and microphones, for user interaction.
* Hardware requirements should be minimal to ensure accessibility and scalability in different computing environments.

# Software Requirements

The software requirements are as follows:

* **Python**
  + **Name and Description**: Python is a high-level, general-purpose programming language known for its readability and wide usage in machine learning and data science.
  + **Version / Release Number:** Python 3.8+ recommended.
  + Databases: While Python doesn’t directly interact with databases, libraries like pandas can be used to load data from databases.
  + **Operating Systems:** Works with Windows, macOS, Linux.
  + **Tools and Libraries:** Python supports a big ecosystem of libraries required for machine learning and natural language processing.
  + **Source:** Available on Python’s official website.
* **Transformers**
  + **Name and Description:** The Transformers library, developed by Hugging Face, is an open-source library giving pre-trained transformer models, including BERT.
  + **Version / Release** Number: Transformers version 4.0+
  + Databases: Does not directly interact with databases but gives data through compatible data formats.
  + **Operating Systems**: Works with Windows, macOS, Linux.
  + Tools and Libraries: Includes pretrained models such as BERT, GPT, RoBERTa, and a trainer API for streamlined model fine-tuning.
  + **Source:** GitHub - Transformers
* **PyTorch**
  + **Name and Description:** PyTorch is an open-source machine learning library by Facebook that allows tools to build and train neural networks.
  + **Version / Release Number:** PyTorch 1.7.0+
  + It works not only with databases but also combines multiple data-loading features for machine learning.
  + **Operating Systems:** Works with Windows, macOS, Linux.
  + **Tools and Libraries:** Gives automatic differentiation, data parallelism, and supports the BERT model training through the Transformers library.
  + **Source:** GitHub - PyTorch
* **Hugging Face Datasets**
  + **Name and Description**: Datasets is a library by Hugging Face that allows users to quickly load and preprocess large datasets for NLP and machine learning tasks.
  + **Version / Release Number:** Datasets version 1.2+
  + **Databases:** Does not directly work with databases but can load data from formats like CSV, JSON, and plain text.
  + **Operating Systems:** Works with Windows, macOS, Linux.
  + **Tools and Libraries:** Transformers library for data handling.
  + **Source:** GitHub - Datasets
* **Pandas**
  + **Name and Description:** pandas is a Python library giving data manipulation and analysis features, widely used for handling tabular data.
  + **Version / Release Number:** pandas 1.2+
  + **Databases:** SQL
  + **Operating Systems**: Works with Windows, macOS, Linux.
  + **Tools and Libraries:** Employed numpy for structured data manipulation in Python.
  + **Source**: GitHub - Pandas
* **NumPy**
  + **Name and Description:** numpy is the Python library for numerical computing that has facilities for operations on arrays and matrices.
  + **Version / Release Number**: numpy 1.19+
  + **Databases**: Does not work with databases directly.
  + **Operating Systems:** Works fine with Windows, macOS, Linux.
  + **Tools and Libraries:** Very commonly used for mathematical operations and data manipulation, in conjunction with libraries like pandas.
  + **Source:** GitHub - NumPy
* **OpenCV**:
  + **Name and Description:** OpenCV-Open Source Computer Vision Library is a library of programming functions mostly aimed at real-time computer vision applications
  + **Version/Release Number**: OpenCV 4.5.3
  + **Databases**: OpenCV does not interact directly with the database but can be used for image and video processing.
  + **Operating Systems:** Works on all Windows and macOS, Linux, Android, iOS.
  + **Tools and Libraries:** OpenCV is equipped with an all-round set of tools and algorithms for image processing that cover feature detection, object recognition, and video analysis.
  + **Source:** OpenCV is open-sourced and available on GitHub, thus allowing sources via source code, documentation, and community contributions.
* **Librosa**
  + **Name and Description**: Librosa- Libraries for audio analysis and processing. It is mostly used in extraction and audio processing.
  + **Version / Release Number:** librosa >= 0.8
  + **Databases**: No direct database support is given
  + **Operating Systems:** It supports Windows, mac OS and Linux.
  + **Source**: GitHub - Librosa
* **NLTK (Natural Language Toolkit)**
  + **Name and Description**: NLTK is a Python library for processing human language data (text) and usually applied in tokenization, text processing, and NLP-related functions.
  + **Version / Release Number**: nltk >= 3.5
  + **Databases**: Not considering the direct database support but mainly on the text and NLP tasks.
  + **Operating Systems:** Works on Windows, macOS, and Linux.
  + **Source**: GitHub - NLTK
* **SciPy** (Signal Processing)
  + **Name and Description**: SciPy is a library for mathematics, science, and engineering that is free and open-source. Most applicable modules in this library are the signal modules that are applied to data associated with signal processing, for example finding peaks in audio data.
  + **Version / Release Number:** scipy >= 1.5
  + **Database:** It doesn't have direct utilization when it comes to databases. Scientific and signal processing.
  + **Supported OS:** Windows, macOS, and Linux
  + **Source:** GitHub - SciPy
* **FuzzyWuzzy**
  + **Description and Name:** FuzzyWuzzy is a Python close string matching library. It is particularly good for matching text approximately.
  + **Version / Release Number**: fuzzywuzzy >= 0.18
  + **Databases**: No database support; it is focused on string matching.
  + **Operating Systems**: Supports Windows, macOS and Linux.
  + **Source**: GitHub - FuzzyWuzzy
* **MoviePy**
  + **Name and Description:** MoviePy is a Python library for video editing, that supports cutting, editing, and composing video files. Here, it is applied on VideoFileClip.
  + **Version / Release Number:** moviepy >= 1.0
  + **Databases**: Not supported by database directly
  + **Operating Systems:** Supports Windows, macOS, and Linux
  + **Source:** GitHub - MoviePy
* **Whisper**
  + **Name and Description:** Whisper is an open-source Automatic speech recognition (ASR) system developed by OpenAI with the function to transcribe audio into text.
  + **Version / Release Number**: whisper >= 1.0
  + **Databases:** No direct database support.
  + **Operating Systems:** Available for Windows, macOS, and Linux.
  + **Source:** GitHub - Whisper
* **OS (Operating System)**
  + **Name and Description**: The OS module in Python is the way of using operating system-dependent functionality, such as file and directory manipulation.
  + **Version / Release Number:** Built-in Python library
  + **Databases:** Not applicable; OS module handles file system operations.
  + **Operating Systems:** Works on Windows, macOS, and Linux.
  + **Source**: Python OS Module Documentation
* **Logging**
  + **Name and Description:** Logging is a standard Python library for tracing events during code execution, useful for debugging and tracking execution.
  + **Version / Release Number:** Built-in Python library
  + **Databases:** No database support; manages logging events and messages.
  + **Operating Systems:** Cross-platform (Windows, macOS, Linux).
  + **Source**: Python Logging Module Documentation

# Communication Interfaces

* The system shall support communication between the client and the server components using local area network protocols, such as TCP/IP, for data interchange.
* The compatibility and interoperability of other systems shall not be compromised in pursuit of adherence to the standard communication standards and protocols.
* It should be optimized such that the line speed and buffer size offer the precedence so that the data transmitted would get processed as quickly as possible with all forms of latencies eliminated and throughput maximized.

# Non-Functional Requirements

# Performance Requirement

**5.1.1 Processing Speed:**

* Objective: The system must be able to analyze sports videos and produce highlights in real-time or near-real time.
* Metric: For a given video, a time of one minute or less should pass while it is being processed. Average should be within 30 seconds.

**5.1.2 Scalability:**

* Objective: Support increasing number of users and increasing different sports events without degradation in performance.
* Metric : Simultaneous handling of 1000 requests and video processing should not affect the performance of the system. Scalability tests must be performed periodically for the system to demonstrate continuous good performance with increased loads.

**5.1.3 Reliability:**

* Objective: The system should reliably and effectively identify significant events in sport videos.
* The metric will be: Detection accuracy of 90% or more in major sports events. MTBF should reach at least 500 hours.

**5.1.4 Robustness:**

* Objective: The system must gracefully deal with differences in video quality, noise, and unpredicted data patterns.
* Metric: The system shall maintain consistency in its performance irrespective of the resolution of the video (for example SD, HD) and variations in frame rates and commentaries. Test cases for robustness should also include scenarios that involve heavy background noises as well as variations in audio qualities.

# Safety Requirements

**5.2.1 Data Integrity:**

* It is very important to ensure that the data obtained is actual, to achieve good quality sports highlights.
* Metric Use checksum and digital signature to ensure data integrity. Carry out routine checks and audits on data integrity. Maintain a data integrity rate of 99.99% over any given time on the system.

**5.2.2 User Safety:**

* Objective: Consumer privacy is ensured by anonymizing and securing any personal information retrieved in extracting tweets of significant events.
* Metric: Anonymize user-related data using anonymization techniques and maintain the secrecy of personally identifiable data. Conduct third-party privacy audits to ensure that the data is handed over as per the privacy policy designed.

**5.2.3 Backup and Recovery:**

* Requirement Ensure there are strong backup/recovery measures in place to protect from losing data and thereby ensuring system stability.
* Metric: System data, which includes event metadata, comments, and generated highlights are regularly backed up. A policy for retaining the backups is in place as well as guarantee that it is possible to recover the system to full operational state within [specific time, e.g., 4 hours] after failure or loss of data. Periodically test backup restoration procedures.

# Security Requirements

**5.3.1 Data Privacy:**

* To respect the confidentiality of a user's data especially for comments, tweets, and other personally identifiable information.
* Implement strong encryption mechanisms for data in motion and at rest to ensure unauthorized access. Conduct periodic privacy impact assessments and verify that compliance with relevant data protection regulations is met. The system should maintain a privacy compliance rate of 98% or better.

**5.3.2 Access Control:**

* Objective Implement Access Control Implementation of controls on the actions that should only be performed and accessed by authorized users allows them to open up sensitive information.
* Metric: Enforce role-based access control: User's access shall be based on his or her role. He must frequently review and update the permissions of users. The system should monitor, log all access to sensitive data, and investigate every incident of unauthorized access.

**5.3.3 Secure Communication:**

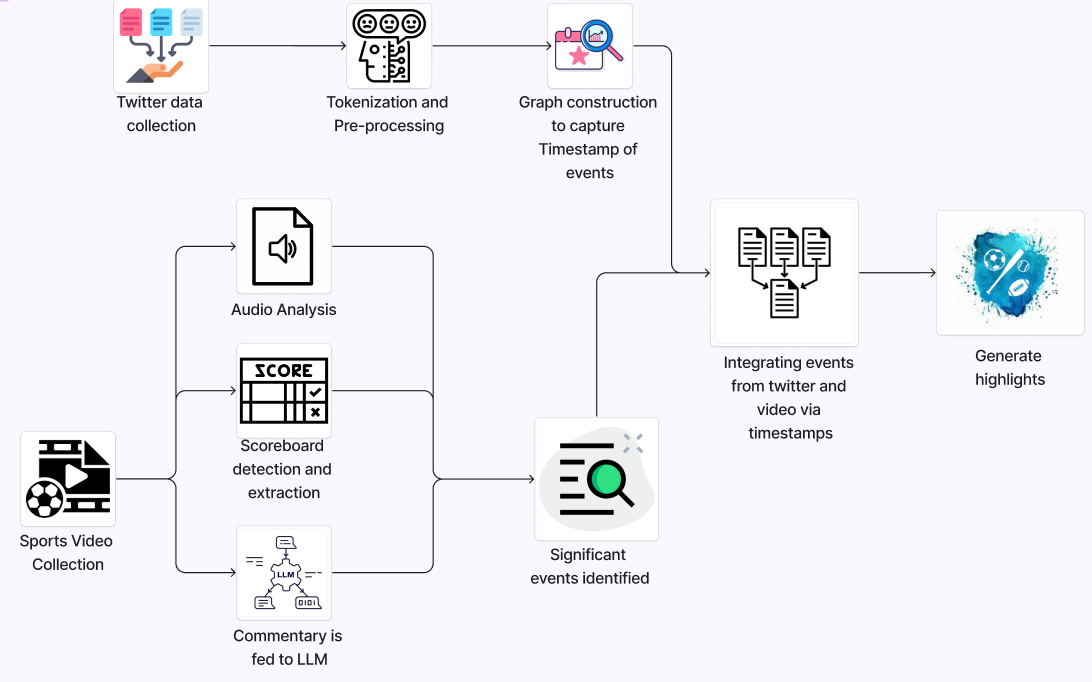
* Objective: Apply best practices in any integration with third-party APIs, especially with Twitter, ensuring security in communication and data exchange. API keys and tokens should be protected and regularly rotated for more advanced security.
* Metric: Update and store API keys and tokens safely with regular consistency. Secure communication to/from third-party APIs through secure protocols. Access logs for APIs usage are monitored in case of detecting even the slightest activity.

# 

# CHAPTER 5

**SYSTEM DESIGN**

**5.1 High Level Architecture Design**

****

**Fig 5.1 High Level Architecture design diagram**

The figure 5.1, shows the high level architecture diagram for our project Indexing and Summarization of Sports Videos using Multi-Modal Approach

**Twitter Data Collection:** In this step we gather data related to sports events from twitter with the help of hashtags and keywords.

**Tokenization and Preprocessing:** After the twitter data is collected, we tokenize the cleaned text into words, remove stopwords and irrelevant tokens. We then group tweets by time into “super documents” representing each time bucket.

**Timestamp of events identified:** The process of identifying significant events includes graph construction to which we provide the super documents as input. For each bucket we create a co-occurrence graph where words are nodes and edges exist between words that co-occur within a window of size 5. We calculate Growth Factor, aggregated centrality and heartbeat score. The absolute difference between consecutive heartbeat scores are calculated, if the difference exceeds a threshold, label it as a significant event (1), else no event (0). Then we construct a co-occurrence graph for the event and compute centrality and extract top 10 words as event keywords.

**Sports Video Collection:** This component , collecting video footage of sports events.

**Audio Analysis:** This component helps us identify significant events in the video footage by analyzing the crowd noise by analyzing the changes in the audio intensity.

**Scoreboard detection :** This component focuses on detecting the scoreboard within the video frames using YOLOv4. Once the scoreboard is identified and localized, the content of the scoreboard is extracted and analyzed for score updates using OCR. The timestamp associated with a new significant event is recorded.

**Commentary fed to LLM:** In this component, commentary data that is fetched from video footage is fed into a large language model to identify significant events within the sports video.

**Significant events identified:** After the video analysis done to identify significant events from various models we aggregate all of the significant events identified.

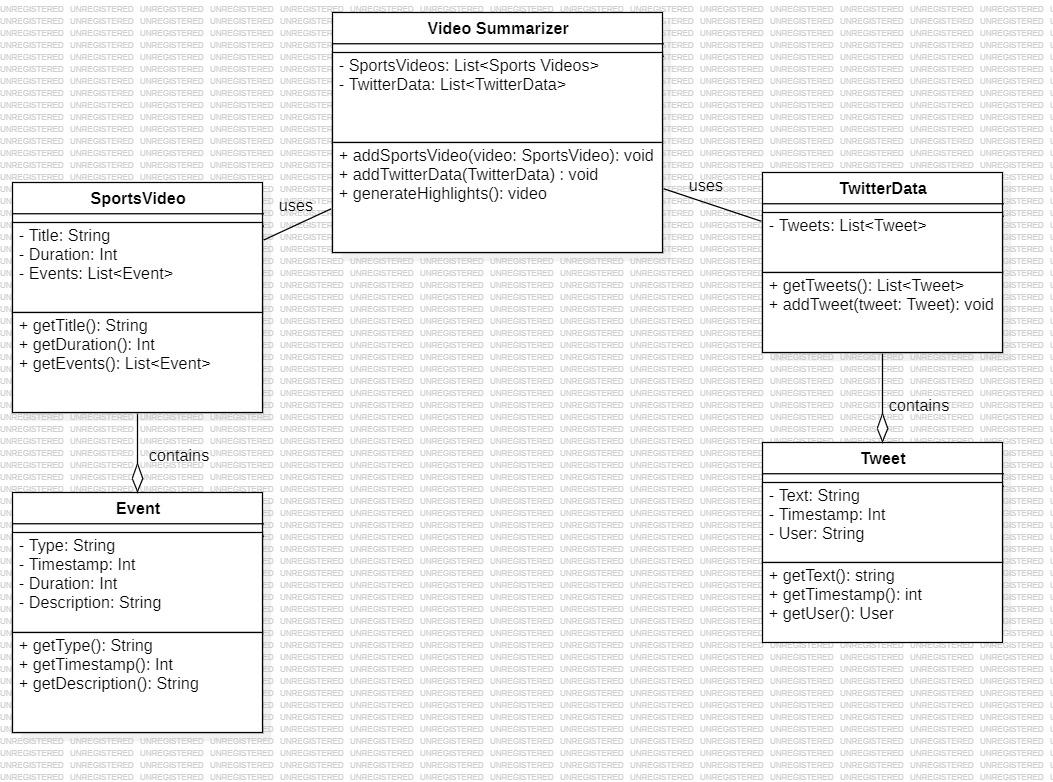
**Integrating events from twitter and video :** The significant events identified from both video and twitter data are combined together to facilitate the final process.

**Generation of highlights:** Based on the identified significant events, this component generates highlights clips from the video footage using computer vision techniques.

**Conceptual or Logical View (UML Component Diagram):**

* **Logical User Groups:**
  + System Administrators
  + Analysts
  + Commentators/Broadcasters
* **Application Components:**
  + Twitter and Video Data Collection Component
  + Event Detection Component: Includes sentiment analysis, WDHG event detection, scene detection, scoreboard analysis, and audio analysis.
  + Event Linking and Analysis Component :Matches events detected in the video footage with the ones in twitter data, generates highlights, and integrates with LLM for commentary.
  + Highlights UI
* **Data Components:**
  + Social Media Data: Tweets collected for analysis.
  + Video and Audio Data: Game footage and audio streams processed for event detection.
  + Event Data: Structured data representing detected events and associated metadata.
* **Interfacing Systems:**
  + External APIs
  + Large Language Model
* **Process - Runtime View :**
  + Sequence of Interactions: User interacts with the User Interface Component.
  + Data flows through Data Collection, Event Detection, and Event Linking Components.
  + Results are presented back to the User Interface for visualization.
* **Physical - Distributed System View :**
  + Server Node hosting application components.
  + Client Devices accessing the User Interface Component.
  + External Services (Twitter, LLM Service) integrated via network interfaces.
* **Security - System Security Features:**
  + Twitter users data encryption
  + Access Control: Assign roles and a set of permissions to various user groups.
  + Secure communication with external APIs using authentication and authorization mechanisms.
  + Data Privacy: Compliance with data privacy regulations and best practices.

**5.2 MASTER CLASS DIAGRAM**



**Fig 5.2 Master class diagram**

According Fig 5.2 there are 5 classes:

1. **The Video Summarizer:**

* This is the main class of our system and it uses the sports videos and twitter data
* Both of these components are used by this class in order to generate the sports video’s summaries.

1. **Sports Video:**

* This class contains a list of sports videos that is provided to the system in order to generate summaries.

1. **Twitter Data:**

* This class contains the tweets associated with each video and it is provided to the video summarizer to generate summaries that incorporate the user’s reactions and insights.

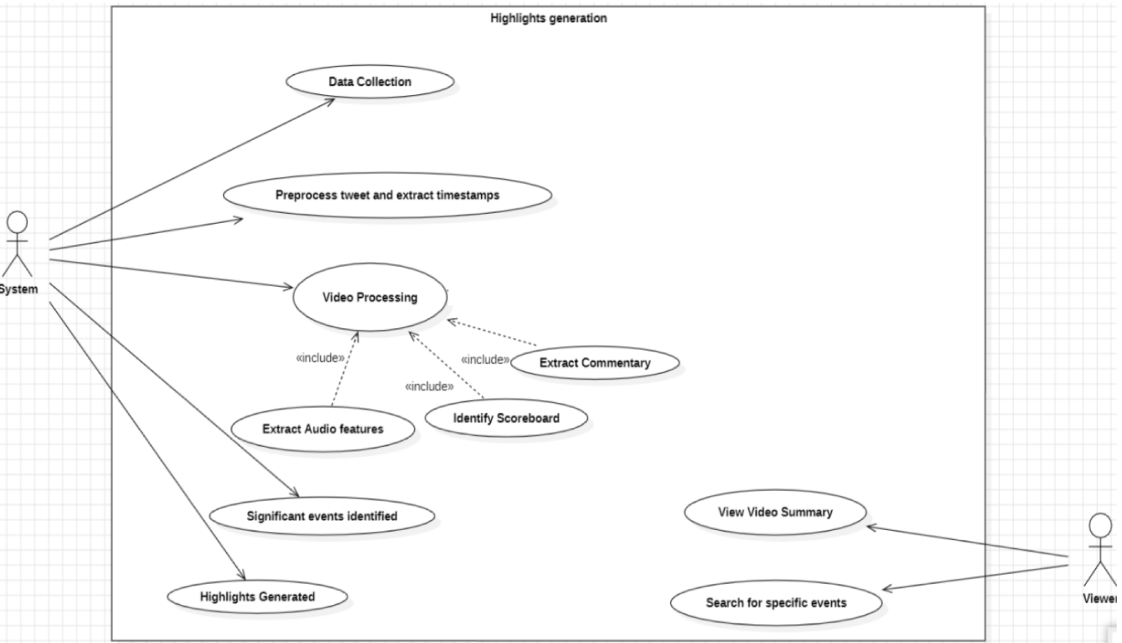
1. **Events:**

* This class contains various events that will be used by the system to flag as an important event or not

1. **Tweets:**

* This class contains the individual tweets that comprise the twitter data.

**5.3 USE CASE DIAGRAM**

****

**Fig 5.3 Use case diagram**

In the figure there are a few actors and use cases as below:

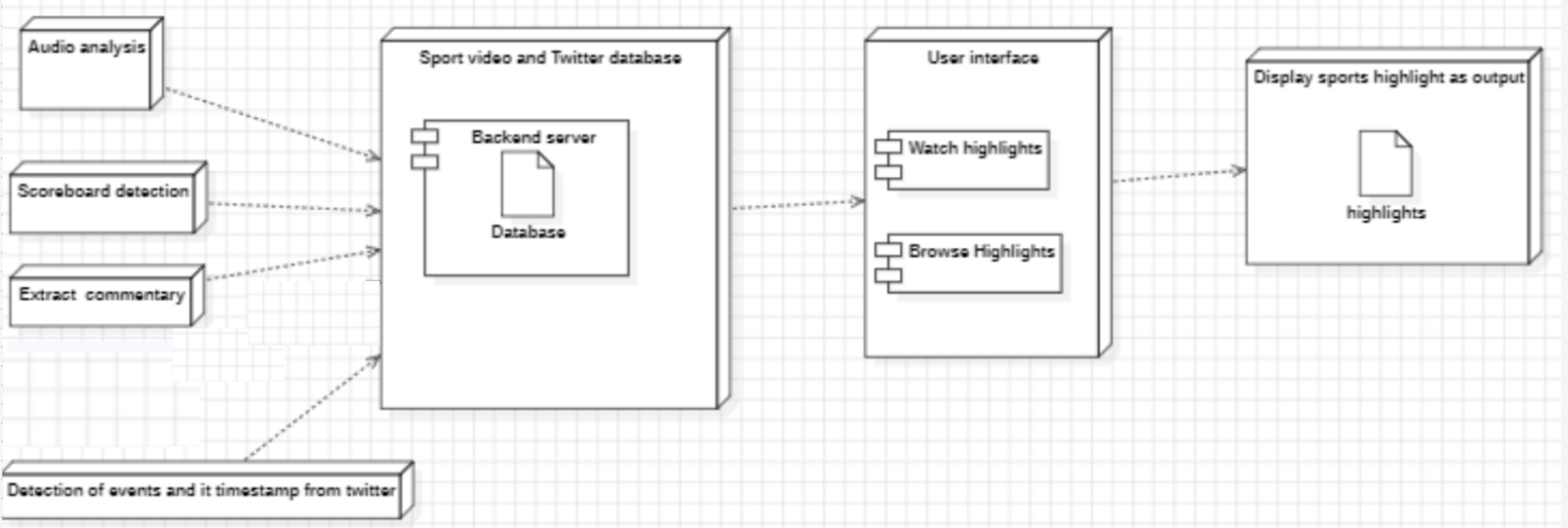
**Actors:**

* **System**: It interacts with the twitter and video data, performs analysis and identifies significant events which will be aggregated to provide the video summarization.
* **Viewer**: The viewer is the user, analyst or broadcaster/commentators who will make use of the system to view highlights.

**Use cases:**

* **Data Collection :** This use case represents the process of gathering data of video footage and twitter from their respective sources.
* **Video Processing :** This use case signifies the analysis of sports video. There are sub-processes associated with this use case.
* **Extract Commentary :** The system extracts audio commentary from the video to feed it to the LLM to identify important moments based on the commentator's emphasis.
* **Identify Scoreboard :** The system identifies and extracts information from the scoreboard regarding the score updates.
* **Extract Audio Features :** This use case which is the sub process analyzes crowd noise to identify exciting moments from the game.
* **View Video Summary :** This use case of the Viewer allows them to view the generated highlights video.
* **Search for specific events:** The viewer can search for specific events in the footage as we extract the timestamp associated with each significant event.

**5.4 Deployment Diagram**



**Fig 5.4 Deployment Diagram**

* Backend server- this is main program that runs the entire system, it houses the analysis software that analyzes the commentary and identifies the events
* Database: This stores the sports videos and Twitter data.
* Audio analysis: This module extracts the commentary from the sports videos.
* Scoreboard detection: This module finds the scoreboard and extracts timestamp from it.
* Extracting tweets and its timestamp from twitter: Extract the tweets and its timestamp and process it to identify significant events.
* User interface: This is the main user interface that the user can use to interact with
* the system.

# CHAPTER 6

**PROPOSED METHODOLOGY**

The methodology that we propose for multi-modal sports video summarization involves creating a brief yet highly relevant and informative highlights video by using a variety of different data sources: Tweets during the game, crowd and commentator background audio, scoreboard changes, and professional commentary. By integrating these different sources together, we are able to record significant events from multiple diverse perspectives, allowing our sports summary engine to provide a comprehensive and enriching summary for the viewers, thus improving user experience and engagement.

**6.1 Twitter**

The boom in social media in general and Twitter in particular has simplified access to vast real-time data during that major event. However, this data is often referred to as "noisy," diverse, and dynamic; therefore, it is difficult for accurate event detection. The scaling and adaptability in environments with fast-changing trends become quandary with normal techniques. The ultimate goal is to have a powerful system capable of taking high data spikes along with accuracy in detecting important events and getting as less redundancy as possible.

To address this, a hybrid approach combining graph-based Event detection with focused goal-driven keyword analysis adoption was used to provide accuracy and thoroughness in detection.

**Graph-Based Event Detection**

The movement starts with a tweet cleaning process, removing URLs or mentions or hashtags and discarding non-English tweets and moving to preprocess the data to make it normalized and eliminate punctuation, tokenize text, and remove stop words to bring forth relevance. Then the tweets would be bucketed in time intervals to capture the temporal structure for analysis to be done according to timeframes looked for.

A co-occurrence graph for each time bucket is constructed with nodes containing unique words and edges referring to co-occurrences among them in a determined window (e.g. two words). Each of the edges is weighted by the number of co-occurrences that this edge indicates, allowing for dynamic relationships between words to evolve with the data stream over time.

The following two key characteristics are extracted from the graph:

**Growth Factor (GF)**: This tells the extent to which the edge weights change from one time bucket to the next and thereby indicates between which the relationships between words change and could suggest important events.

**Aggregated Centrality (AC)**: This shows the relative importance of each word in terms of degree centrality in the graph. The aggregated centrality is computed as the total of these centrality nodes and is a measure of overall importance.

These measures, Growth Factor, and Aggregated Centrality, are fused to produce accumulated scores that would enable the identification of potentially important events. A time bucket is denoted as significant on the criteria that its score surpasses a predefined mark-and that's a reflection of changing underlying dynamics.

**Goal-Related Keyword Analysis**

Besides graph-based detection, a keyword analysis should also be performed to capture tweets related to some specific goals during a match. Tweets containing the term "goal" will thus be collected, organized in intervals of five minutes, and analyzed to assess the frequency value for the keyword "goal" within each interval for any possible peaks or dips in count.

Time One should also rank each of the timestamps that showed the highest counts of the "goal" keyword to be able to identify some of the moments that might turn to be important in the event and can include when a goal was scored or when possibilities for major discussions happened about a goal.

**Combined Method**

Based on the insights obtained from both the graph-derived aggregated score and the goal-related keyword analysis, event detection is possible with much higher accuracy. The graph-based approach would have provided the total trends of the events in general based on the structure and dynamics of the tweeting stream, while the keyword analysis would delve into very specific moments like goals. This means that event detection will be high in precision and recall for critical moments but will also adapt and become extremely robust to changing trends in the data.

**6.2 Scoreboard detection and extraction**

In sports video broadcasts, a superimposed scoreboard typically displays game details, such as the current score, game time and team names to improve the viewer's game progression. As the scoreboard updates live for each major event (for example, goals; or other timekeeping, points scored), the detection and recognition of the scoreboard plays a key role in automated sports video analysis, serving as a primary source of information for detecting game events and extracting relevant statistics.

Our approach to this involves several principal stages, the first stage being frame extraction. It involves segmenting the video into individual frames. This segmentation allows for precise localization and analysis of the scoreboard across different frames.

Following frame extraction, in order to have a dynamic scoreboard detection model we employed YOLOv4, a deep learning object detection framework. This model has been trained to recognize the scoreboard dynamically within each frame thus adapting to diverse scoreboard placements. This detection model ensures flexibility and robust performance across various broadcast layouts without depending on fixed coordinates, making it highly suitable for general sports video analysis.

After locating the scoreboard, we made use of Optical Character Recognition (OCR) to extract the displayed scores, the game time and the team names. The OCR process focuses on the stable regions of the detected scoreboard, ensuring that extracted text remains consistent, capturing any updates in score or team information accurately across the video. As the scoreboard position remains fixed, the OCR can reliably interpret the changing details without interference from other screen elements. Then, a mechanism was implemented to capture the team names, score and game time from the scoreboard only post a significant event (e.g., goals or wickets) giving us an accurate time stamp for the event.

**6.3 Audio Analysis**

In this paper, this approach is a multimodal framework that aids in the detection and annotation of key events in sports videos through audio analysis. The process can be broken down into four main steps: audio extraction, audio transcription, audio peak detection, and peak annotation with context-specific keywords. First, a video file undergoes a process wherein the audio is first extracted in WAV format and then used as an input for audio analysis and transcription.

After audio extraction, it is transcribed using an effective ASR model that enables English transcription. The outcome of the transcription is a full textual representation of the audio commentary describing the events as they appear in the video. It is further processed to include predefined sports-related keywords such as "goal," "foul," and "penalty," which are indicators of key events.

The audio signal contains peaks of interest that are analyzed through the help of both RMS energy and spectral flux as indicators of sudden change in intensity that are often represented with an event. These peaks are calculated within the audio signal by computing segments of higher energy or flux than average using thresholding to keep the highest peaks. This enables the system to identify the essential moments correlated with events of large intensity.

In order to annotate, peak times are aligned with transcriptions corresponding to relevant keywords; therefore, there will be contextual insights on keywords. A fuzzy matching algorithm compares words in the transcript around each peak with a set of predefined sports-related keywords, applying a high match threshold to actually capture key event mentions; thus this step allows for precise association between audio peaks and event keywords.

Finally, with the aim of indexing annotated events over their timestamps, it forms another index which further describes what happens in the video for the occurrences of each event. The found events are then saved in a structured format for easy retrieval and summarizing so that users can access important moments in quick time, review them as well, and generate summaries easily. This multi-modal method uses both audio signal analysis and natural language processing to realize efficient yet accurate sports video summarization.

**6.4 Commentary Classification Using BERT**

In the proposed multi-modal summarization approach, we fine tune a BERT (Bidirectional Encoder Representations from Transformers) model to classify football commentary text into specific events in football. This modality makes use of BERT’s natural language understanding to recognise and classify specific patterns in football commentary, and map each commentary segment to a specific event category such as "free kick," "foul," or "attempt."

The event classification process begins with us making use of the BERT model to obtain dense vector textual representation for the commentary text. Each commentary segment is cut off to 512 tokens because that is the maximum context window size in BERT and it is also long enough such that we do not lose relevant context around the specific event. This ensures that the model processes the entire necessary input without any loss due to truncation, therefore capturing unique patterns in commentary that will help in effectively classifying football events.

During training, we trained the BERT model for multi-class classification with 5 training epochs, achieving a final classification accuracy of 98%.

This commentary-based classification helps our system to accurately recognize and prioritize significant moments by using commentary (text modality), enhancing the multi-modal summarization framework. This approach shows how LLM based fine-tuning can be used to achieve high precision for real-world sports event classification tasks.

# 6.5 Multi-modal Integration

In the proposed framework for multi-modal sports video summarization, integrating data from various different sources is of utmost importance to accurately identify and prioritize significant events. For this we propose a timestamp clustering and weighting algorithm that synthesizes temporal data from all the modalities (commentary, crowd noise, score board).

Let **Ti,j** ​represent the **jth** timestamp of the **ith** modality, where **i ∈ {1,2,…,M}** for each of the **M** modalities, and **j ∈ {1,2,…,Ni}** for **Ni**​ timestamps from the **ith** modality.

To determine overlapping timestamps across multiple modalities, we define a **clustering window of** **Δt seconds** that groups timestamps into clusters if they fall within **Δt** seconds of each other. Let **Ck**​ denote the **kth** cluster, which includes timestamps **{Ti,j}** such that:

**∣Tij−Ti′j′∣ ≤ Δt, ∀ i,i′ ∈{1,2,…,M},j,j′∈{1,2,…,Ni.**

Each cluster **Ck** is calculated a **weight** **wk**​ based on the number of modalities contributing to that cluster:

**wk =∣ {i Ti,j ∈ Ck} ∣**

Thus, the weight **wk**​ signifies the significance of the cluster, with higher values indicating more importance because of multi-modal consensus on the event's importance.

For customisable durations of summaries, clusters **Ck** are sorted in descending order by weight **wk**, giving higher priority to events that have a multi-modal consensus. Based on the desired length of summary (short, medium, long), a subset of the top clusters is selected to fit the target duration. Thus, the number of clusters selected **Nselected** varies dynamically to produce a summary of the desired length.

**Nselected = Top clusters sufficient to match the target summary duration**

This selection mechanism allows the algorithm to effectively integrate the timestamps generated by each modality into a single summary of desired length

# 

# 

# 

# 

# CHAPTER 7

**IMPLEMENTATION AND PSEUDOCODE**

**7.1 Event Detection From Twitter Data**

* **Technology Stack :** 
  + **Programming Language:** Python
* **Libraries and Tools :**
  + **Regular Expressions :** For cleaning the tweet text and removing texts or patterns which are not needed.
  + **LangDetect:** For the detection and filtering of only English tweets.
  + **NLTK:** For text tokenizing and inferring stop words.
  + **NetworkX:** For making co-occurrence images and analysis of computer graphs.
* **Modal Workflow Overview :**
  + **Stage 1: Data Loading -** Loaded tweets from a particular CSV file .
  + **Stage 2: Text Preprocessing -** Texts were merged and cleaned by eliminating urls, mentions, and special characters.
  + **Stage 3: Tokenization & Stopwords Removal -** The text was cut into pieces (words) and most irrelevant words were eliminated (stopped).
  + **Stage 4: Time Bucketing -** Batches or buckets of Twitter feeds were created every two minutes as per the timestamps.
  + **Stage 5:Super Document Creation -** Time bucket verbal constituents were fused together to make super documents.
  + **Stage 6: Co-occurrence Graph Construction -** Build co-occurrence graphs for each time bucket with the help of word pairs.
  + **Stage 7: Growth Factor Calculation -** The growth factor is determined based on the alteration of total edge weights in regards to the two graphs.
  + **Stage 8: Aggregated Centrality Calculation -** Total degree centrality was estimated for each of the graphs.
  + **Stage 9: Heartbeat Score Calculation -** Heartbeat scores were estimated as the multiplication of the growth factor and the aggregated centrality.
  + **Stage 10:Classify Events-** Mark timestamps as significant events if the change in heartbeat scores exceeds a threshold.
  + **Stage 11: Extract Significant Timestamps -** Extract timestamps corresponding to significant events based on classifications.
  + **Stage 12: Count "Goal" Keyword -** Count occurrences of the word "goal" in each tweet and add as Goal\_Count.
  + **Stage 13: Time Bucketing -** Group tweets into 2-minute intervals using the Date column as time\_bucket.
  + **Stage 14:Create Lists/Dictionaries-**  Make a list of significant event timestamps and a dictionary of top goal timestamps with their counts.
  + **Stage 15: Convert to DataFrames -** Turn both the list and dictionary into pandas DataFrames.
  + **Stage 16: Sort DataFrames -** -Sort the Data Frames by timestamp so they are in order.
  + **Stage 17: Merge Data Frames -** Combine the two DataFrames based on matching timestamps, keeping all significant events.
  + **Stage 18:Fill and Filter-** Replace missing goal counts with 0 and remove rows with 0 goals to keep only relevant events.
* **Pseudocode and Algorithm Workflow**

For each row in tweets:

Take ‘Tweet’ and ‘Orig\_Tweet’ together, clean the text (remove Urls, @mensions) and filter english

Further clean ‘Cleaned\_Tweet’:

Make them lowercase, remove punctuations, along the process, cut and sieve the elements and exclude stop words

Define periods in a group say two minutes and aim the tokens at ‘super documents’

For each super document:

Construct graph G such that edges exists between each two words in a circumferential window of a fixed size

Growth factors and aggregated centrality are calculated for successively labeled graphs.

In the form of GF\*AC computes the heartbeat scores and relates the scores with the appearance of noteworthy events.

create time buckets (e.g., 2-minute intervals).

Count occurrences of the word "goal" in each tweet and aggregate by time bucket.

Sort time buckets by goal count, and extract the top time buckets with the highest goal counts.

Convert the significant event timestamps and top goal data into DataFrames, ensuring 'time\_bucket' is in datetime format.

Merge the Data Frames on 'time\_bucket' using an outer join and fill missing Goal\_Count values with 0.

**7.2 Scoreboard Detection and Extraction**

* **Technology Stack :** 
  + **Programming Language:** Python
* **Libraries and Tools :**
  + **YOLOv4 :** For dynamic object detection of the scoreboard.
  + **OpenCV:** For video frame processing.
  + **OCR:** For extracting textual data (score, time and team names).
  + **CSV:** For storing timestamps of events.
* **Modal Workflow Overview :**
  + **Stage 1: Frame Extraction -** Individual frames extracted from the video for analysis.
  + **Stage 2: Scoreboard Detection -** YOLOv4 was used to dynamically locate and detect the scoreboard.
  + **Stage 3: Text Extraction -** Used OCR to interpret score, time and team names from the detected scoreboard.
  + **Stage 4: Event Detection -** Changes in scores observed helped capture the significant events details.
  + **Stage 5: Storing timestamps -** Captured the timestamps of significant events derived in stage 4 and stored into csv file for further processing.
* **Pseudocode and Algorithm Workflow**

for each frame in video:

detect scoreboard location using YOLOv4

if scoreboard detected:

crop scoreboard region

extract score, time, team names with OCR

if score changes from previous frame:

record timestamp in CSV

**7.3 Audio Analysis:**

* **Technology Stack**:
  + **Programming language :** Python
* **Libraries and Tools:**
  + **Whisper :** For transcription of commentator speech for keywords detection.
  + **librosa** : used for tasks such as loading audio, calculating audio energy, and detecting onset strength.
  + **nltk** : it handles tasks like removing common stop words (e.g., "the," "and") from the transcribed audio text.
  + **fuzzywuzzy** : Here, it’s used to detect similar words in the transcribed text (e.g., "goal" might match with "goaaal") by comparing words in the text to a predefined set of keywords.
  + **moviepy.editor.VideoFileClip** : This module is used to load the video and extract its audio track, which is then analyzed to detect events.
* **Modal Workflow Overview**
  + **Stage1:**First convert extract the audio from the video using moviepy library.
  + **Stage2**:Define the keywords that are related to the key events.
  + **Stage3**:Detect the peaks in the audio using RMS intensity using the librosa library using the peak threshold.
  + **Stage 4**:Also get the transcription of the whole video and check if any predefined keywords are present or not closely with a match threshold at that detected peak through RMS intensity.
  + **Stage 5**:Now check if both the RMS intensity and the transcription contains the key event then save the timestamp in csv file with the annotation as the nearby transcript.
  + **Stage 6**:Now we have all the timestamps in the csv files which is one of the inputs to the multi modal.
* **Pseudocode and Algorithm Workflow**
  + Initialize Processors and Load Resources
  + Create instances of AudioProcessor, Transcription Processor, Audio Analyzer, and Text Processor classes.
  + Define keywords for detection, audio processing parameters, and other constants.
  + Extract Audio from Video
  + Use the Audio Processor to extract audio from the video file, saving it to a temporary audio file.
  + Transcribe Audio
  + Call the Transcription Process to convert the audio into text using a transcription API (e.g., Whisper or Google API).
  + Detect Audio Peaks
  + Use the Audio Analyzer to analyze audio energy and detect peaks (high-intensity moments).
  + Peaks represent possible key events, identified by detecting spikes in RMS and spectral flux.
  + Generate and output a list of events with timestamps.

**7.4 Commentary Classification Using BERT**

* **Technology Stack :** 
  + **Programming Language:** Python
* **Libraries and Tools :**
  + **Transformers:** For Natural Language Processing tasks involving transformers
  + **Datasets:** For data loading and preprocessing efficiently
  + **PyTorch:** For training deep learning models
  + **Pandas:** for Data handling
  + **Numpy:** for Mathematical operations
* **Workflow Overview :**
* **Stage 1: Data Preparation -** Label each event category and divide the dataset into test, validation, and training sets.
* **Stage 2: Tokenization -** Use BERT's tokenizer to set the maximum padding length for each commentary segment.
* **Stage 3: Model Setup -** Provide the BERT model with event categories that have been labeled for classification.
* **Stage 4: Training -** The model is trained on the training set and validated on the validation set, which involves configuring the training parameters (checkpoints, batch size, and epochs).
* **Stage 5: Evaluation -** Assess the model's ability to accurately classify the test set.

**7.5 Multi-Modal integration Algorithm**

* **Technology Stack :** 
  + **Programming Language:** Python
* **Libraries and Tools :**
  + **CSV:** for handling CSV files.
* **Workflow Overview :**
  + **Stage 1: Input Data Parsing -** 
    - Events from each modality are saved in separate CSV files, each of which has timestamps in the "mm:ss" format.
    - The module loads the timestamps for each modality into lists after reading each modality CSV file using the CSV library.
  + **Stage 2: Timestamp Conversion -**
    - To ensure cross-modal comparisons happens, timestamps are transformed from "mm:ss" format to seconds.
  + **Stage 3: Multi-modal integration -** 
    - The algorithm combines events by determining timestamps from several modalities that occur inside a predetermined time window.
    - Each event is given a weight that reflects its significance and is determined by the number of modalities that capture the same event in the given window.
  + **Stage 4: Output formatting -** 
    - The finished product is a collection of timestamp pairs (start and end timings) that indicate which events should be included in the summary video.
* **Pseudocode :**

For each CSV file in modality\_files:

Read file using csv library

Convert each timestamp to seconds

Append converted timestamps to modality data list

Initialize empty list for merged events

For each unique timestamp in sorted list of all modality timestamps:

Calculate weight based on modality overlap within window\_size

Append (timestamp, weight) to merged events list

Initialize empty list for formatted summary segments

For each event in merged events:

Convert start and end times to "mm:ss" format

Append (start, end) pair to summary segments

# 

# CHAPTER 8

**RESULTS AND DISCUSSION**

This section shows the performance of our proposed multi-modal system for sports video summarization showcasing how it can incorporate information from Tweets, score board detection and audio analysis.

**8.1 Events Detection from Twitter Data Result**

The hybrid Twitter-oriented event detection system is able to combine both graph-based event detection and goal-related keyword analysis into an effective method for identifying salient moments during the event. At the same time, as the analysis of the structure of tweet streams and frequencies of the keyword "goal" within a certain time interval was achieved, the system could detect important events within a goal timestamp with an excellent level of accuracy-95\% of the time. Although the occasional peaks and troughs in tweet activity brought about missing event detection, trends in this model were quickly adapted to it so that important events were detected appropriately.

**Table 8.1.1: Goal Event Detection By Twitter**

| **Match** | **Detected Goal** | **Total Goals** | **Goal Ratio** |
| --- | --- | --- | --- |
| France vs. Croatia | 5 | 6 | 83.33% |
| Qatar vs. Ecuador | 2 | 2 | 100% |

#### 

#### 

#### 

#### 8.2 Scoreboard Extraction and Detection Result

YOLOv4 successfully detected scoreboards in different videos, as well as successfully extracted essential information (team names, scores, time) from high-resolution frames (1280×720, 30 frames per second) with an accuracy of 89.06 percent. Sometimes low-resolution frames and noisy frames negatively impacted the accuracy for the algorithm.

**8.3 Audio Analysis and Commentary Classification Result**

The audio analysis module, using Whisper for transcription, managed to attain 95% transcription accuracy. Event detection based on RMS intensity and peak finding was generally accurate in detecting actual key events; however, sometimes the timestamp detection was noisy, especially from suboptimal quality audio inputs. While using BERT to classify the commentary, the model was able to get over 95% accurate classifications. It was understood that certain misclassifications occurred due to the differences between commentary and the actual events taking place.  
We chose **Goal & Scoring Ratio** () as the primary metric for evaluating the audio modality, as it directly measures the effectiveness of capturing goal and near goal events. Accuracy was not selected as a metric because most of the moments in a football match do not involve key events like goals. A model could get a high accuracy just for predicting ‘no event’ for most of the match thus not testing its capability of easily finding important highlights. On the other hand, Precision and Recall are more suitable metrics in evaluating the models capabilities of correctly identifying relevant important events.

**Table 8.3.1: Audio Modality Metrics Table**

| **Match** | **Goals and scoring opportunities Detected by Model** | **Total Goals and scoring opportunities** | **Goal and Scoring Ratio** |
| --- | --- | --- | --- |
| France vs. Croatia | 10 | 15 | 66% |
| Portugal vs. Spain | 8 | 12 | 66.6% |
| Belgium vs. Brazil | 10 | 16 | 62.5% |

The audio modality's **Goal Ratio** indicates its ability to capture goals based on sound intensity. While it missed some goals, it consistently detected key moments, demonstrating its importance in event detection.

#### 8.4 Multi-modal Integration Result and Comparative Analysis

In order to evaluate our overall model, we compare its output with popular highlights available on YouTube for that match, as these summaries are curated to showcase key moments of a match and serve as a widely accepted benchmark for event detection quality.

Our system-generated summaries captured all events included in popular YouTube highlights and additional near-goal events not present in the highlights. While these extra events might be considered false positives, they represent key moments that enhance the viewer's experience, making our summaries more enriching.

**Metric Definitions**:

* **Precision**:
* **Recall**:

**Performance Metrics for Multi-modal Integration**:  
A comparison between events captured by our system and YouTube highlights is summarized below:

**Table 8.4.1: Multi-modal metrics Table**

| **Match** | **Events in YouTube Summary** | **Events in Generated Summary** | **Relevant Events In Detected Highlights** | **Precision** | **Recall (%)** |
| --- | --- | --- | --- | --- | --- |
| France vs. Croatia | 7 | 10 | 9 | 70% | 90% |
| Portugal vs. Spain | 8 | 11 | 9 | 72% | 81.8% |
| Belgium vs. Brazil | 6 | 9 | 7 | 66.6% | 77% |

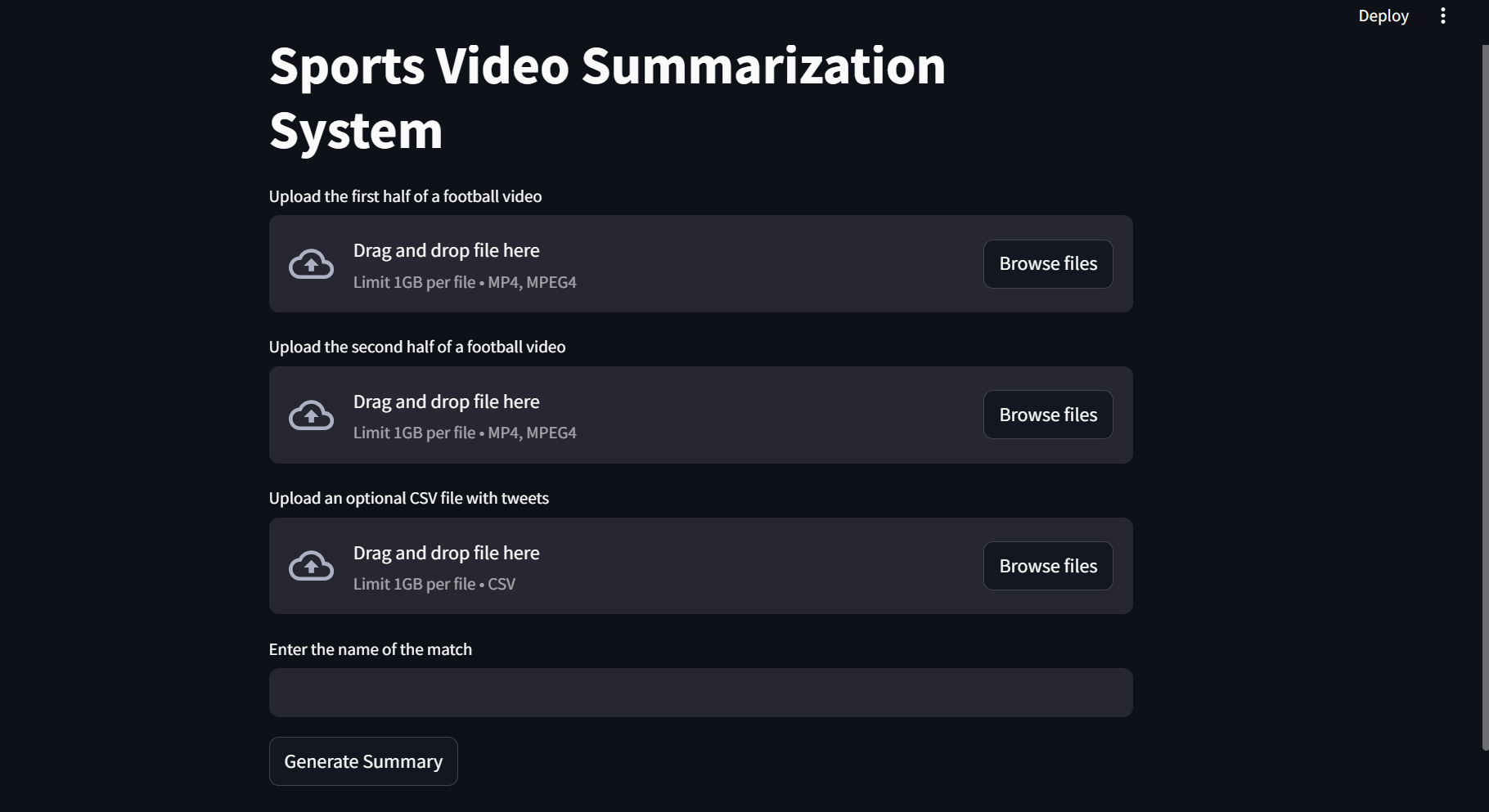
**Goal Detection in Overall System**:  
The scoreboard detection modality ensured all goals were successfully identified, yielding a **100% Goal Ratio** in a few matches. This result highlights the crucial role of scoreboard detection in augmenting the system's accuracy for key event identification.

**Table 8.4.2: Multi-modal Goal Ratio**

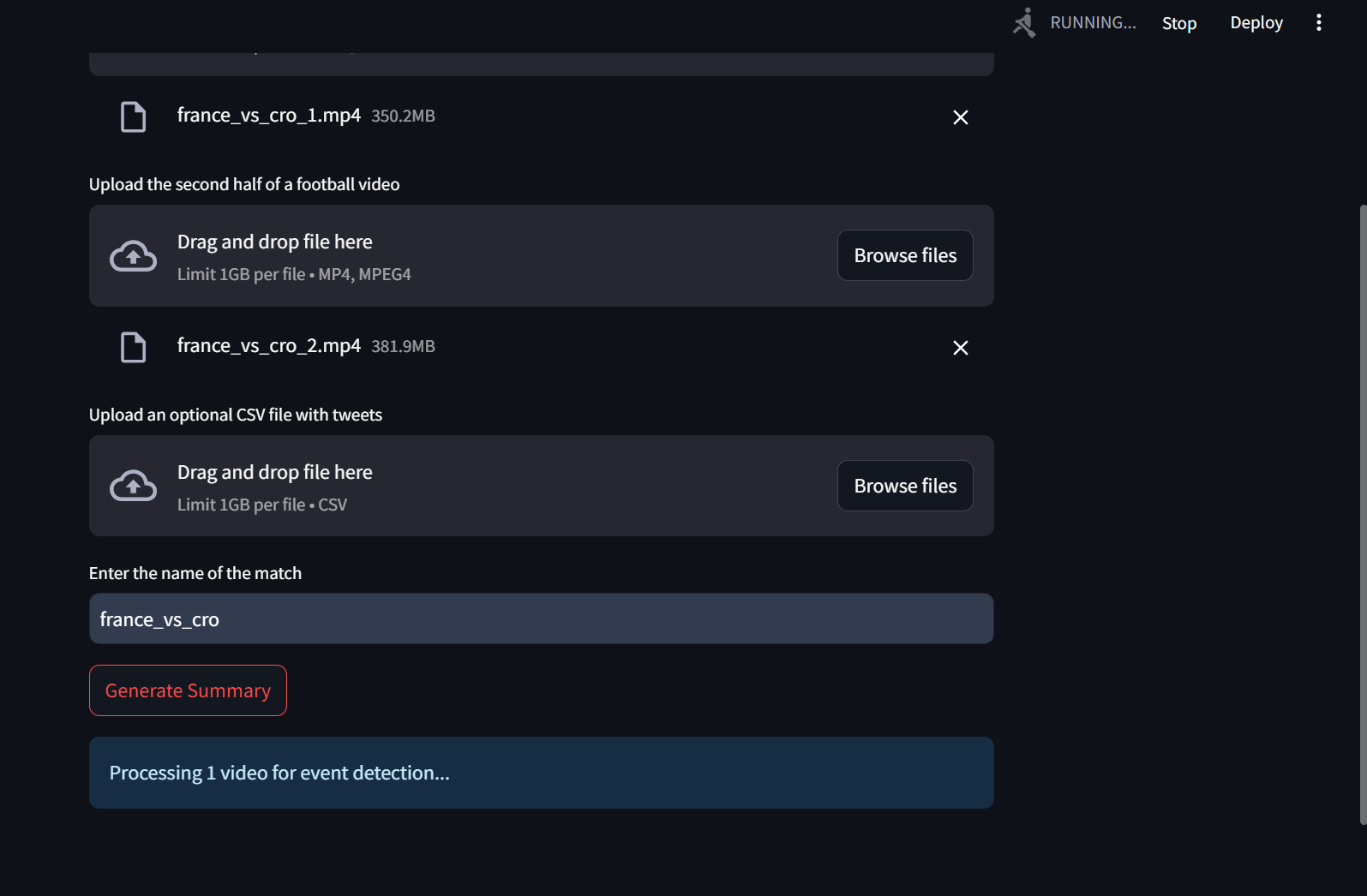
| **Match** | **Goals Detected by Model** | **Total Goals** | **Goal Ratio (%)** |
| --- | --- | --- | --- |
| France vs. Croatia | 6 | 6 | 100% |
| Portugal vs. Spain | 5 | 6 | 83.3% |
| Belgium vs. Brazil | 3 | 3 | 100% |

By integrating scoreboard detection, our system ensures complete coverage of goals, addressing the limitations of audio-based detection. This proposed multi-modal approach provides a more reliable summarization of critical moments in football matches.

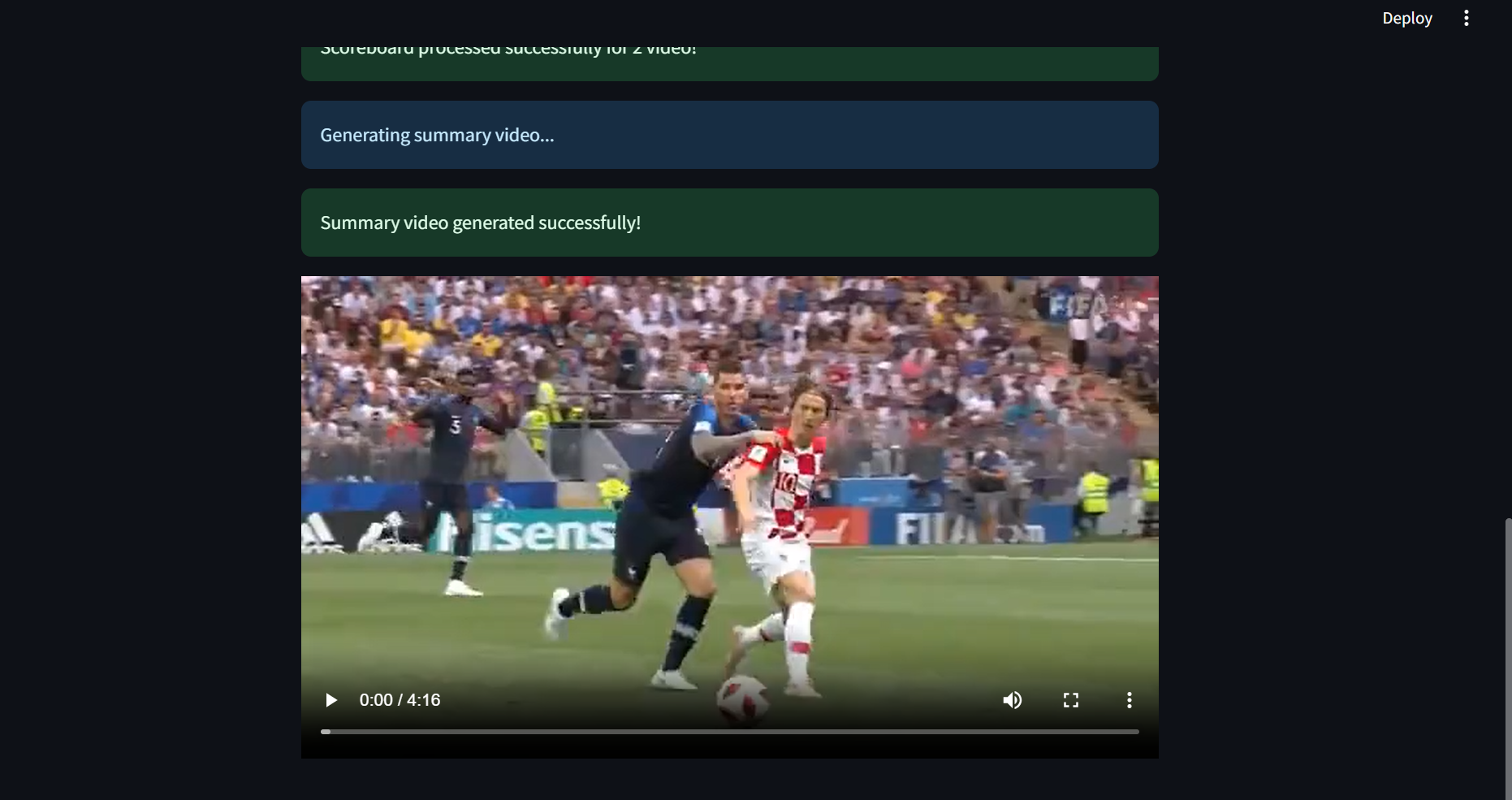
**8.5 User interface:**

****

**Fig 8.5.1 :Homepage**



**Fig 8.5.2 :Generating Summary**



**Fig 8.5.3: Successfully generated the summary**

# 

# CHAPTER 9

**CONCLUSION AND FUTURE WORK**

The proposed multi-modal integration framework successfully summarizes sports events by merging data from multiple modalities, such as audio commentary, scoreboard detection, and social media analysis. By using the state of the art technologies in transcription, object detection and event classification (BERT), this system creates game highlights, significantly reducing the need for intensive manual editing. By integrating data from multiple different modalities, the system is able to detect key events accurately.

While the framework is able to achieve positive results, transitions between clips at times seem abrupt, sometimes cutting into the middle of an event. To enhance the viewing experience, future work needs to be carried out in order to improve the current switch from segment to segment by predictive models that capture the entire context of the highlight.

Additionally, exploring how this framework could integrate with AR/VR could enable fans to experience highlights in a more immersive and interactive way, potentially creating new avenues for engagement and fan retention.

Overall, the system gives promising results for automated sports summarization. As techniques in multimodal integration grow, this framework can be extended to provide more personalized, fluid, and engaging highlight presentation methodologies that enhance relations between fans and sports content.

# 

# REFERENCES/BIBLIOGRAPHY

[1] Hirasawa K, Maeda K, Ogawa T, Haseyama M. “Detection of Important Scenes in Baseball Video Videos via a Time-Lag-Aware Multimodal Variational Autoencoder.” Sensors (Basel). 2021 Mar 14;21(6):2045. doi: 10.3390/s21062045. PMID: 33799412; PMCID: PMC7999231.

[2] Raj, R., Bhatnagar, V., Singh, A. K., Mane, S., & Walde, N. (2021, January 21). Video Summarization: Study of various techniques. arXiv.org. https://arxiv.org/abs/2101.08434

[3] H. Sattar, M. S. Umar, E. Ijaz and M. U. Arshad, "Multi-Modal Architecture for Cricket Highlights Generation: Using Computer Vision and Large Language Model," 2023 17th International Conference on Open Source Systems and Technologies (ICOSST),2023, pp. 1-6, doi: 10.1109/ICOSST60641. 2023.10414235

[4] Z. Saeed, R. Ayaz Abbasi, M. I. Razzak and G. Xu, "Event Detection in Twitter Stream Using Weighted Dynamic Heartbeat Graph Approach [Application Notes]," in IEEE Computational Intelligence Magazine, vol. 14, no. 3, pp. 29-38, Aug. 2019, doi: 10.1109/MCI.2019.2919395

[5] R. S. Bhat, J. O, P. P. P, P. Kumar Vedurumudi and D. K. N, "Cricket Video Summarization Using Deep Learning," 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), Lonavla, India, 2023, pp. 1-6, doi: 10.1109/I2CT57861.2023.10126359

[6]Video Summarization Study Of Various Techniques Proceedings of IRAJ International Conference,26th May, 2019

[7]C. Yan, X. Li and G. Li, "A New Action Recognition Framework for Video Highlights Summarization in Sporting Events," 2021 16th International Conference on Computer Science & Education (ICCSE), Lancaster, United Kingdom, 2021, pp. 653-666, doi: 10.1109/ICCSE51940.2021.9569708.

[8]A. Bhalla, A. Ahuja, P. Pant and A. Mittal, "A Multimodal Approach for Automatic Cricket Video Summarization," 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2019, pp. 146-150, doi: 10.1109/SPIN.2019.8711625

[9]Helenca Duxans, Xavier Anguera and David Conejero ,Telefónica Investigación y Desarrollo,"Audio based Soccer Game Summarization",June 2009,Broadband Multimedia Systems and Broadcasting, 2009. BMSB '09. IEEE International Symposium on,DOI: 10.1109/ISBMSB.2009.5133759.

# Appendix

# Appendix A: Definitions, Acronyms and Abbreviations

1. **Multi-modal Data Analysis**: Analysis of data from multiple sources or modalities, such as Twitter streams, video analysis, and audio analysis, to extract insights or information.
2. **Computer Vision**: A field of artificial intelligence that focuses on enabling computers to interpret and understand visual information from the real world, often involving image or video processing.
3. **Audio Analysis**: The process of analyzing audio signals to extract meaningful information or features, such as speech recognition, sound classification, or sentiment analysis.
4. **Twitter Streams:** Continuous, real-time flow of tweets or messages posted on the social media platform Twitter.
5. **User Interface:** The point of interaction between users and a computer system, including visual elements such as screens, buttons, and menus.
6. **OCR (Optical Character Recognition):** The process of converting images of text into machine-encoded text.
7. **OpenCV (Open Source Computer Vision Library):** A library of programming functions for real-time computer vision tasks.
8. **Large Language Models:** AI models that can perform natural language processing (NLP) tasks like generating text, translation, and question-answering
9. **BERT:** Bidirectional Encoder Representations from Transformers, is a machine learning framework that helps computers understand the meaning of text by analyzing the relationships between words in a sentence
10. **Whisper:** Whisper is a machine learning model that can transcribe speech in English and other languages, and translate some non-English languages to English. It was created by OpenAI and released as open-source software in September 2022.
11. **Librosa:** A python package for music and audio analysis.
12. **Natural Language Toolkit (NLTK):** A Python programming environment for creating applications for statistical natural language processing (NLP).
13. **Fuzzywuzzy:** To measure the similarity between two strings.
14. **moviepy.editor.VideoFileClip** : This module is used to load the video and extract its audio track, which is then analyzed to detect events.
15. **YOLOv4:** For dynamic object detection of the scoreboard.
16. **Machine Learning Library**: Transformers and datasets library by hugging face.
17. **Regular Expressions : For cleaning tweet text by removing unwanted characters and patterns.**
18. **LangDetect:** For detecting and filtering only English tweets.
19. **NLTK:** For text tokenization and stopword removal.
20. **NetworkX:** For creating co-occurrence graphs and performing graph-based analysis.