A
Major Project Report

On

"A Deep Insight on Cricket Video to Text Summarization Using Neural Networks"

Submitted in partial fulfillment of the

Requirements for the award of the degree of

Bachelor of Technology

In

Computer Science & Engineering –

Artificial Intelligence & Machine Learning

By

Dhatrika Kamal Kumar	**20R21A6613**
Maddhe Sai Prashanth	**20R21A6633**
Vuppala Praneeth Kumar	**20R21A6658**
I Nikhil Sri Sai Teja	**20R21A6620**

Under the guidance of
Dr. SIVAKRISHNA KONDAVEETI
Associate professor

Department of Computer Science & Engineering

MARRI LAXMAN REDDY GROUP OF INSTITUTIONS
MLR INSTITUTE OF TECHNOLOGY
(UGC AUTONOMOUS)
Affiliated to JNTUH,Approved by AICTE
Laxman Reddy Avenue, Dundigal, Hyderabad-500 043, Telangana, India
NATIONAL BOARD of ACCREDITATION
NAAC A

2024

MARRI LAXMAN REDDY GROUP OF INSTITUTIONS

MLR INSTITUTE OF TECHNOLOGY

(UGC AUTONOMOUS)

Affiliated to JNTUH, Approved by AICTE

Laxman Reddy Avenue, Dundigal, Hyderabad-500 043, Telangana, India

Department of Computer Science & Engineering-

Artificial Intelligence & Machine Learning

CERTIFICATE

This is to certify that the project entitled **"A Deep Insight on Cricket Video to Text Summarization Using Neural Networks"** has been submitted **by D.Kamal Kumar (20R21A6613), M.Sai Prashanth (20R21A6633), V.Praneeth Kumar (20R21A6658), I Nikhil Sri Sai Teja (20R21A6620)** in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering - Artificial Intelligence & Machine Learning from Jawaharlal Nehru Technological University, Hyderabad. The results embodied in this project have not been submitted to any other University or Institution for the award of any degree or diploma.

Internal Guide **Head of the Department**

Project coordinator **External Examiner**

MARRI LAXMAN REDDY GROUP OF INSTITUTIONS
MLR INSTITUTE OF TECHNOLOGY
(UGC AUTONOMOUS)
Affiliated to JNTUH,Approved by AICTE
Laxman Reddy Avenue, Dundigal, Hyderabad-500 043, Telangana, India

Department of Computer Science & Engineering-

Artificial Intelligence & Machine Learning

DECLARATION

We hereby declare that the project entitled **"A Deep Insight on Cricket Video to Text Summarization Using Neural Networks"** is the work done during the period from **January 2024 to May 2024** and is submitted in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering - Artificial Intelligence & Machine Learning from Jawaharlal Nehru Technology University, Hyderabad. The results embodied in this project have not been submitted to any other university or Institution for the award of any degree or diploma.

Dhatrika Kamal Kumar	**20R21A6613**
Maddhe Sai Prashanth	**20R21A6633**
Vuppala Praneeth Kumar	**20R21A6658**
I Nikhil Sri Sai Teja	**20R21A6620**

MARRI LAXMAN REDDY GROUP OF INSTITUTIONS
MLR INSTITUTE OF TECHNOLOGY
(UGC AUTONOMOUS)
Affiliated to JNTUH, Approved by AICTE
Laxman Reddy Avenue, Dundigal, Hyderabad-500 043, Telangana, India

Department of Computer Science & Engineering-

Artificial Intelligence & Machine Learning

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we now have the opportunity to express our guidance for all of them.

First of all, we would like to express our deep gratitude towards our internal guide **Dr. Sivakrishna kondaveeti,** Associate professor, Department of CSE -AIML for his support in the completion of our dissertation. We wish to express our sincere thanks to **Dr. K. SAI PRASAD,** HOD, Dept. of CSE -AIML and principal **Dr. K. SRINIVAS RAO** for providing the facilities to complete the dissertation.

We would like to thank all our faculty and friends for their help and constructive criticism during the project period. Finally, we are very much indebted to our parents for their moral support and encouragement to achieve goals.

Dhatrika Kamal Kumar	**20R21A6613**
Maddhe Sai Prashanth	**20R21A6633**
Vuppala Praneeth Kumar	**20R21A6658**
I Nikhil Sri Sai Teja	**20R21A6620**

MARRI LAXMAN REDDY GROUP OF INSTITUTIONS MLR INSTITUTE OF TECHNOLOGY (UGC AUTONOMOUS) Affiliated to JNTUH,Approved by AICTE Laxman Reddy Avenue, Dundigal, Hyderabad-500 043, Telangana, India

Department of Computer Science & Engineering-

Artificial Intelligence & Machine Learning

ABSTRACT

Cricket is one of the most followed sports by audience throughout the world. It is a highly sought out form of entertainment with 2.5 billion spectators though it's a niche in terms of geography but still leaves a lot of untapped audience and applications due to its long matches and underperforming summarizers. In this Study, we dive into a new totality of the framework for cricket match video summarization. We propose the use of advanced Deep Learning techniques like VGG16 Convolutional Neural Networks (CNNs), Optical Character Recognition (OCR), Long Short-Term Memory Recurrent Neural Networks (RNNs) and You Only Look Once (YOLO) for text and object detection from the match. This ensures the quality summary and also makes sure that there - exist no bias and get a better version than existing summarizing systems. From this study we get an ultimate summarization tool which performs better while capturing crucial events and display text for the user to consume.

Keywords: Cricket video summarization, Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), Optical Character Recognition (OCR), You Only Look Once (YOLO), Object Detection, Machine Learning.

APPENDIX-1

LIST OF FIGURES

LIST OF FIGURES

APPENDIX-2

LIST OF TABLES

	LIST OF TABLES	
Table No	**Description of Figure**	**Page No**
Table 2.2	**Comparison table**	**71**
Table 2.3	**Work Evaluation table**	**74**
Table. 3	**Possible combinations of data that can extracted from video frames**	**110**

APPENDIX-3

LIST OF ABBREVIATIONS

ABBREVIATIONS

YOLO	**You Only Look Once**
OCR	**Optical Character Recognition**
CNN	**Convolutional Neural Network**
LSTM	**Long Short-Term Memory**
BART	**Bidirectional and Auto-Regressive Transformers**

APPENDIX-4

REFERENCES

REFERENCES

References

[1] R. Agyeman, R. Muhammad and G. S. Choi, "Soccer Video Summarization Using Deep Learning," 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), San Jose, CA, USA, 2019, pp. 270-273, doi: 10.1109/MIPR.2019.00055.

[2] C. Lin and Y. Chen, "Sports Video Summarization with Limited Labeling Datasets Based on 3D Neural Networks," 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, Taiwan, 2019, pp. 1-6, doi:10.1109/AVSS.2019.8909872.

[3] Y. Takahashi, N. Nitta and N. Babaguchi, "Video Summarization for Large Sports Video Archives," 2005 IEEE International Conference on Multimedia and Expo, Amsterdam, Netherlands, 2005, pp. 1170- 1173, doi: 10.1109/ICME.2005.1521635.

[4] M. Z. Khan, S. Jabeen, S. ul Hassan, M. A. Hassan, and M. U. G. Khan, "Video Summarization using CNN and Bidirectional LSTM by Utilizing Scene Boundary Detection," 2019 International Conference on Applied and Engineering Mathematics (ICAEM), Taxila, Pakistan, 2019, pp. 197-202, doi: 10.1109/ICAEM.2019.8853663.

[5] M. B. Andra and T. Usagawa, "Automatic Lecture Video Content Summarizationwith Attention-based Recurrent Neural Network," 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), Yogyakarta, Indonesia, 2019, pp. 54-59, doi: 10.1109/ICAIIT.2019.8834514.

[6] S. H. Emon, A. H. M. Annur, A. H. Xian, K. M. Sultana and S. M. Shahriar, "Automatic Video Summarization from Cricket Videos Using Deep Learning," 2020 23rd International Conference on Computer and Information Technology (ICCIT), DHAKA, Bangladesh, 2020, pp. 1- 6, doi: 10.1109/ICCIT51783.2020.9392707

[7] Shingrakhia, Hansa, and Hetal Patel. "SGRNN-AM and HRF-DBN: a hybrid machine learning model for cricket video summarization." The Visual Computer 38, no. 7 (2022): 2285-2301.

[8] Besta Srikanth, Sagarla Aravind, Mopuri Veera Narayana, Narayana Satya Narayana, “Sports Match Video to Text Summarization Using Neural Network.”, 2023 INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY (IJIRT).

[9] Guntuboina C, Porwal A, Jain P, Shingrakhia H. Deep learning based automated sports video summarization using YOLO. ELCVIA Electronic Letters on Computer Vision and Image Analysis. 2021 May 27;20(1):99-116.

[10] Dilawari, Aniqa and Muhammad Usman Ghani Khan. “ASoVS: Abstractive Summarization of Video Sequences.” IEEE Access 7 (2019): 29253-29263.

[11] Abhishek Yadav, Anjali Vishwakarma, Shyama Panickar, Prof. Satish Kuchiwale, "Real Time Video to Text Summarization using Neural Network", 2020 International Research Journal of Engineering and Technology (IRJET).

[12] Joys Princia A, Ms. J Sangeetha Priya, Kalai Selvi J, Rithi Afra J, Rukshana S, "Video and Text Summarization Using VDAN and RNN",2021 INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY (IJIRT).

[13] Hansaraj Wankhede, Rachana Chawke, R Bharathi Kumar, Sushant Kawade, & Ashish Ramtekkar. (2023). AI-based Video Summarization using FFmpeg and NLP. International Journal of Innovative Science and Research Technology, 8(4), 1140–1145. https://doi.org/10.5281/zenodo.7888972

[14] J. Mun, L. Yang, Z. Ren, N. Xu and B. Han, "Streamlined Dense Video Captioning," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 6581-6590, doi: 10.1109/CVPR.2019.00675.

[15] V.Vijayakumar and R.Nedunchezhian, "A Novel Method for Super Imposed Text Extraction in a Sports Video", International Journal of Computer Applications 15(1):1–6, February 2011.

[16] Jingxu Lin , Sheng-hua Zhong , Ahmed Fares “Deep hierarchical LSTM networks with attention for video summarization” .(2022) Computers and Electrical Engineering, 97,art.no. 107618 https://doi.org/10.1016/j.compeleceng.2021.107618

[17] Mayu Otani, Yuta Nakashima, Esa Rahtu, Janne Heikkilä, Naokazu Yokoya ,”Video Summarization using Deep Semantic Features” 16 pages, the 13th Asian Conference on Computer Vision (ACCV'16) https://doi.org/10.48550/arXiv.1609.08758

[18] Maria Nektaria Minaidi, Charilaos Papaioannou, Alexandros Potamianos “Self-Attention Based Generative Adversarial Networks For Unsupervised Video Summarization” https://doi.org/10.48550/arXiv.2307.08145

[19] Shruti Jadon, Mahmood Jasim “Unsupervised video summarization framework using keyframe extraction and video skimming” 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA) 10 November 2020 https://doi.org/10.1109/ICCCA49541.2020.9250764

[20] Zawbaa, H.M., El-Bendary, N., Hassanien, A.E., Kim, Th. (2011). Machine Learning-Based Soccer Video Summarization System. In: Kim, Th., et al. Multimedia, Computer Graphics and Broadcasting. MulGraB 2011. Communications in Computer and Information Science, vol 263. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-27186-1_3

INDEX

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

In this modern era, the demand for efficient and automated cricket video summarization techniques is rapidly increasing. This paper introduces an innovative and advance neural network system that transforms the way cricket match videos are summarized. Cricket video-to-text summarization system overcomes the limitations of traditional manual summarization approach by utilizing various deep learning techniques to completely automate the summarization process, our system can extract crucial insights from lengthy cricket match footage and convert them into easily readable text formats. The system employs three-pronged approach which involves extraction of visual features using VGG16 Convolutional Neural Network (CNN), scoreboard information is extracted through Optical Character Recognition (OCR) technology, and Text Summarization performed by Long Short Term Memory (LSTM) network. Our system revolutionizes the way cricket enthusiasts engage with match videos, providing a game-changing experience for fans worldwide.

1.2 PURPOSE OF THE PROJECT

The purpose of the project is to address the existing challenges and limitations in cricket match summarization by leveraging advanced technologies such as Deep Learning and Computer Vision. By automating the process of extracting insights from cricket match footage, the project aims to provide stakeholders with efficient and accurate summaries that capture key events and trends. Through the integration of techniques like object detection, text recognition, and sequence modeling, the project seeks to enhance the accessibility and comprehensiveness of cricket analysis, benefiting coaches, players, researchers, and enthusiasts alike. Ultimately, the project aims to revolutionize the way cricket events are analyzed and understood, paving the way for more informed decision-making and deeper engagement with the sport.

The purpose of this project is to revolutionize the process of summarizing cricket match videos through advanced deep learning techniques. By leveraging technologies such as Convolutional Neural Networks (CNNs), Optical Character Recognition (OCR), Long Short-Term Memory Recurrent Neural Networks (LSTM), and You Only Look Once (YOLO) for text and object detection, the project aims to deliver high-quality summaries of cricket matches. The ultimate goal is to provide stakeholders, including coaches, players, and enthusiasts, with a reliable tool for capturing crucial events and insights from matches, thereby enhancing their understanding and enjoyment of the game.

1.3 MOTIVATION

Our motivation stems from the widespread popularity of cricket coupled with the challenges many face in keeping up with the sport's lengthy matches. Recognizing the need for accessible and efficient means of understanding cricket events, we aim to bridge the gap between the sport and its audience. By leveraging advanced technologies like neural networks, we seek to transform hours of match footage into concise textual summaries. Our goal is to empower both new and regular viewers with the ability to grasp the key moments and insights from cricket matches quickly and effortlessly. Through this project, we aspire to enhance the accessibility and enjoyment of cricket for a diverse audience, thereby fostering greater engagement and appreciation for the sport.

CHAPTER 2
LITERATURE SURVEY

An extensive literature survey has been conducted by studying existing systems of Cricket video to text summarization. A good number of research papers, journals, and publications have also been referred before formulating this survey.

2.1 EXISTING SYSTEM

The current systems for summarizing cricket matches face notable challenges. Manual summarization demands expertise and incurs significant labor costs, hindering its efficiency and affordability. Despite efforts to ensure objectivity, interpretations can vary due to differing perspectives and potential biases. Additionally, reliance on audio cues, predominantly commentary, introduces another layer of complexity. Commentary may lack neutrality and often digress from match-related topics, detracting from the accuracy and spirit of the summary. These challenges collectively impede the extraction of crucial information from cricket matches and hinder the creation of reliable evaluation processes. Moreover, smaller teams and organizations encounter difficulties accessing match data due to the high expenses associated with manual assessments. With cricket's growing global popularity, addressing these obstacles is paramount to enhancing fans' and stakeholders' comprehension and summarization of cricket events.

The responses to various research articles are documented below by the order of the number that have been used to specify them in the references in the end.

1

Reference in APA format	R. Agyeman, R. Muhammad and G. S. Choi, "Soccer Video Summarization Using Deep Learning," 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), San Jose, CA, USA, 2019, pp. 270-273, doi: 10.1109/MIPR.2019.00055.	
URL of the Reference	**Authors Names and Emails**	**Keywords in this Reference**
https://ieeexplore.ieee.org/document/8695329	Rockson Agyeman Rafiq Muhammad Gyu Sang Choi	Convolutional Neural Network (CNN), Long Short Term Memory (LSTM), Residual Network (ResNet), Feature extraction, Mean Opinion Score (MOS), Batch normalization, Rectified Linear Units (ReLU).
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	**The Goal (Objective) of this Solution & What is the problem that need to be solved**	**What are the components of it?**
Soccer Video Summarization Using Deep Learning	The goal (objective) of the proposed solution in this research paper is to develop an effective video summarization technique specifically tailored for soccer videos. The primary problem that this solution addresses is the time-consuming and labor-intensive process of manually analyzing and summarizing soccer match videos for performance evaluation and strategic analysis.	Author used two key components: a 3D Convolutional Neural Network (3D-CNN) for feature extraction and an LSTM network for modeling temporal evolution. The 3D-CNN is designed based on a modified ResNet architecture, tailored for effective recognition of soccer actions. The LSTM network processes these features to model the temporal evolution of actions. These elements create a framework that identifies and combines pertinent video segments, generating a concise summary for streamlined analysis.

The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process

The authors introduce a novel soccer video summarization model, harnessing a blend of cutting-edge deep learning techniques - a 3D Convolutional Neural Network (3D-CNN) and Long Short Term Memory (LSTM) Recurrent Neural Network (RNN). The 3D-CNN, based on a modified ResNet architecture, adeptly identifies intricate soccer actions, extracting spatiotemporal features from meticulously annotated clips. The LSTM complements this by modeling temporal progression, enhancing highlight identification. This integrated framework autonomously identifies and concatenates pertinent video segments, creating a concise summary. Through rigorous evaluation, the authors demonstrate the model's efficacy, offering a powerful tool for insightful soccer match review. This research significantly advances sports video analysis.

	Process Steps	Advantage	Disadvantage (Limitation)
1	The proposed model employs a modified 3D Convolutional Neural Network (3D-CNN) based on ResNet architecture to effectively extract spatiotemporal features from annotated soccer clips.	Saves Time: Automates the task of summarizing soccer videos, saving analysts' time.	Dependence on Manual Annotation: The model's effectiveness relies on the quality and accuracy of manual annotations, potentially introducing bias or errors in the training data.
2	A Long Short Term Memory (LSTM) Recurrent Neural Network (RNN) is utilized to model the temporal progression of actions, enhancing the system's ability to identify crucial highlights.	Accurate Recognition: Effectively identifies important soccer actions using advanced neural networks.	The use of 3D-CNN, may require significant computational resources, potentially limiting its accessibility for smaller-scale applications.
3	The integrated framework autonomously identifies pertinent video segments, based on the extracted features and temporal modeling, discerning key events within the soccer match.	Considers Timing: Understands the timing of actions, providing a more accurate summary.	While the model is tailored for soccer, its adaptability to other sports may require extensive modifications and additional training data, potentially limiting its utility.

4	Relevant video segments are concatenated to create a concise summary of the soccer match footage, facilitating efficient analysis for coaches and analysts.	Potential for Other Sports: Can be adapted for similar sports like basketball or volleyball with slight adjustments.	Fine-tuning hyper parameters is crucial for optimal performance, making the model potentially sensitive to parameter selection.

Major Impact Factors in this Work

Find all main factors and variables that are related to each solution. Then find the relationship between factors. (Independent variable) causes a change in (Dependent Variable) and it isn't possible that (Dependent Variable) could cause a change in (Independent Variable).

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
Mean Opinion Score (MOS)	The manually annotated soccer Dataset (Soccer-5)		

Relationship Among The Above 4 Variables in This article
The independent variables collectively contribute to the system's ability to recognize and summarize soccer actions, ultimately impacting the overall performance as evaluated by the Mean Opinion Score (MOS).

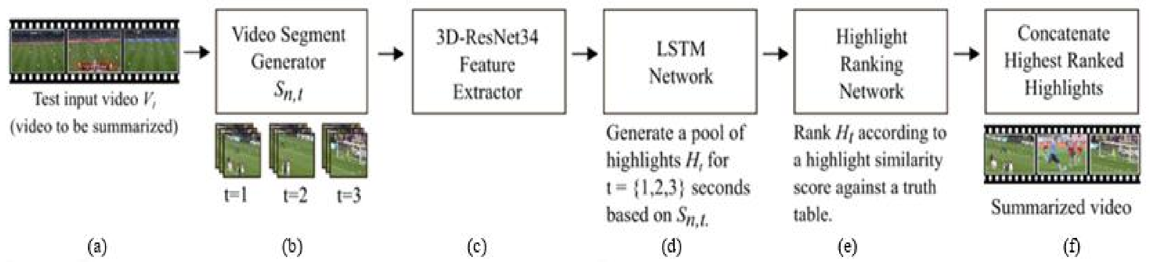
Input and Output	Feature of This Solution	Contribution & The Value of This Work
<table><tr><th>Input</th><th>Output</th></tr><tr><td>Soccer video</td><td>Video Summary</td></tr></table>	This solution employs (3D-CNN) and (LSTM) RNN. It automatically identifies crucial moments in soccer match videos, providing a concise summary for efficient analysis, revolutionizing the process of performance evaluation and strategic planning.	Good to have this knowledge from this paper as we reviewing of ideologies for developing cricket sports summarization using neural networks.

Positive Impact of this Solution in This Project Domain	Negative Impact of this Solution in This Project Domain
Efficient automated analysis saves time, enhances accuracy, and provides coaches with strategic insights, revolutionizing performance	Dependency on advanced technology may limit accessibility, potential bias in training data, and customization

evaluation.	challenges for different sports or specialized scenarios.

Analyse This Work By Critical Thinking	The Tools That Assessed this Work	What is the Structure of this Paper
This paper presents an innovative approach to soccer video summarization, leveraging advanced deep learning techniques. The model's strengths lie in automated highlighting and temporal modeling. However, potential biases in training data and computational demands may limit its broader application. Overall, it significantly advances sports video analysis.	Mean Opinion Score (MOS), the manually annotated soccer Dataset (Soccer-5)	Abstract I. Introduction II. Related Works III. Proposed Approach IV. Experiment and Performance Evaluation V. Conclusion

Diagram/Flowchart



---End of Paper 1---

2

Reference in APA format	C. Lin and Y. Chen, "Sports Video Summarization with Limited Labeling Datasets Based on 3D Neural Networks," 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, Taiwan, 2019, pp. 1-6, doi:10.1109/AVSS.2019.8909872.

URL of the Reference	Authors Names and Emails	Keywords in this Reference
[https://ieeexplore.ieee.org/document/890 9872](https://ieeexplore.ieee.org/document/8909872)	ChingShun Lin YuChing Chen	Video Summarization, 3D Neural Networks, Major League Baseball (MLB), Deep Learning, Convolutional Neural Networks (CNN), Long

		Short-Term Memory (LSTM), Audio- Based Detection, Visual-Based Detection, Keyframe Detection.Video Summarization, 3D Neural Networks, Major League Baseball (MLB), Deep Learning, Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), Audio- Based Detection, Visual-Based Detection, Keyframe Detection.
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	**The Goal (Objective) of this Solution & What is the problem that need to be solved**	**What are the components of it?**
Sports Video Summarization with Limited Labeling Datasets Based on 3D Neural Networks	The goal is to create an efficient video summarization technique for MLB games, focusing on key events and reducing redundant content to enhance the viewing experience.	3D Neural Networks: Detect key frames in MLB video. Audio-Based Detection: Utilize audio cues (e.g., cheering, hitting sounds) for event identification.Visual-Based Detection: Use score display data for event recognition. Advanced Highlight Detection: Combine audio and visual clues for precise event detection. Experimental Results: Empirically test technique's effectiveness and efficiency.

The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process
This work employs 3D neural networks, including 3D Convolutional Neural Networks and 2D Convolutional Long Short-Term Memory (LSTM), for efficient event detection in Major League Baseball (MLB) videos. It integrates audio-based detection using distinct sounds like cheering and hitting, and visual-based detection using score display information. These methods collectively enhance event identification accuracy. Experimental validation demonstrates the approach's effectiveness, efficiency, and robustness. While offering precise event timestamps, computational resources and sensitivity to display format changes are potential limitations.

	Process Steps	Advantage	Disadvantage (Limitation)
1	Video Segmentation: Divide the video into smaller segments using Keyframe detection.	Efficient Summarization: The method condenses sports videos, saving time without compromising quality.	Baseball Specific: Tailored for baseball, limiting versatility across other sports.
2	3D CNN: Apply 3D Convolutional Neural Networks for enhanced spatiotemporal feature representation.	Robust Deep Learning: 3D CNN and 2D Convolutional LSTM enhance feature extraction for accurate summarization.	Audio Quality Impact: Effectiveness relies on clear audio, affected in noisy environments.
3	2D Convolutional LSTM: Combine spatial and temporal context for better video understanding.	Audio Clues: Incorporating sound analysis improves event detection precision.	Score Display Dependency: Hinges on visible score display, potentially missing obscured information.
4	Highlight Detection: Utilize audio and visual cues for precise identification of key moments.	Score Display Integration: Utilizing score information ensures accurate highlight identification.	Technical Complexity: Implementation demands computational resources, may be challenging for non-technical users.

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
Recall Rate (RR), Precision Rate (PR), and F1-score.	Model Architectures, Training Data, Audio and Visual Features		

Relationship Among The Above 4 Variables in This article

The dependent variable (output) in this paper is the effectiveness of video summarization, influenced by the independent variables (inputs) such as deep learning models and features used to bridge the semantic gap in capturing relevant video content.

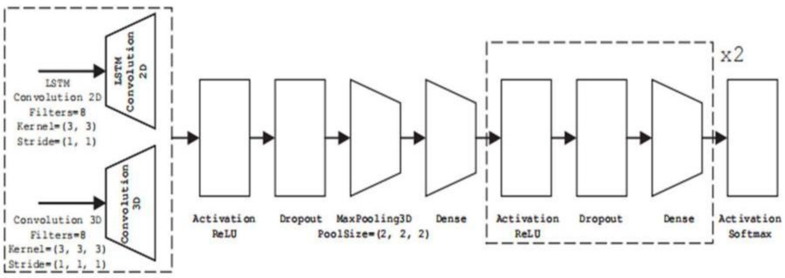
Input and Output	Feature of This Solution	Contribution & The Value of This Work
<table><tr><th>Input</th><th>Output</th></tr><tr><td>Baseball video</td><td>Summarized video containing key events</td></tr></table>	The feature of this solution is its ability to automatically detect and extract key moments and events in baseball games, allowing for a concise	Good to have this knowledge from this paper as we reviewing of ideologies for developing cricket sports summarization using neural networks.

	and highlights	and focused video summary.	

Positive Impact of this Solution in This Project Domain	Negative Impact of this Solution in This Project Domain
This solution applies deep learning for efficient sports video summarization, enhancing viewer experience by highlighting key events and removing redundancy.	Nothing new in terms of core logic. Used two algorithms which are already defined.

Analyse This Work By Critical Thinking	The Tools That Assessed this Work	What is the Structure of this Paper
Innovative deep learning approach enhances sports video experience, but limited applicability and dependence on audio quality pose challenges.	Recall Rate (RR), Precision Rate (PR), and F1- score.	Abstract I. Introduction II. 3D NN for Video Summarization III. Advanced Highlight Detection IV. Experiment Results V. Conclusion and Future Research

Diagram/Flowchart



---End of Paper 2---

3		
Reference in APA format	Y. Takahashi, N. Nitta and N. Babaguchi, "Video Summarization for Large Sports Video Archives," 2005 IEEE International Conference on Multimedia and Expo, Amsterdam, Netherlands, 2005, pp. 1170- 1173, doi: 10.1109/ICME.2005.1521635.	
URL of the Reference	**Authors Names and Emails**	**Keywords in this Reference**
[https://ieeexplore.ieee.org/document/152 1635](https://ieeexplore.ieee.org/document/1521635)	Yoshimasa Takahashi, Naoko Nitta, Noboru Babaguchi	Video Summarization, Sports Video Archives, Metadata, MPEG-7, Image Keyframes, Highlight Extraction, Recall

		and Precision, Greedy Method, Play-Cut Method
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	**The Goal (Objective) of this Solution & What is the problem that need to be solved**	**What are the components of it?**
Video Summarization for Large Sports Video Archives	create concise video summaries for large sports video archives by leveraging metadata and prioritizing significant play scenes.	Play Scene Significance, Summarization Methods, Metadata Usage, Play Scene Selection, Visualization Techniques, Evaluation Metrics, Comparative Methods, Experimental Results

The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process

This work uses metadata to summarize large sports video collections. It ranks play scenes based on factors like importance, timing, and replays. The approach offers two summary types: compressed video clips and organized video posters. Video clips allow flexible adjustment of summary length. Video posters arrange keyframes for easy navigation. Experimental results on baseball videos show promising performance compared to TV broadcasted summaries.

	Process Steps	Advantage	Disadvantage (Limitation)
1	Step 1: Assess Play Scene Significance using Metadata	Efficient Content Retrieval: The proposed method enables users to quickly access important play scenes in large sports video archives, saving time compared to manually searching through entire videos.	Dependency on Metadata: The effectiveness of the method relies on the availability and accuracy of metadata. If metadata is incomplete or inaccurate, it may lead to sub optimal summaries.
2	Step 2: Rank and Select Play Scenes based on Significance	Customizable Summaries: The system allows users to generate video summaries of varying lengths, providing flexibility to tailor the content to their preferences or time constraints.	Limited to Sports Videos: The current system is designed specifically for sports videos, which limits its applicability to other types of video content with different structures or characteristics.

3	Step 3: Generate Summary (Video Clip or Poster)		
4	Step 4: User Interaction and Evaluation (Viewing, Annotations, Metrics)		

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
Recall and precision rates	Metadata-based Significance Measures, User-Specified Parameters		

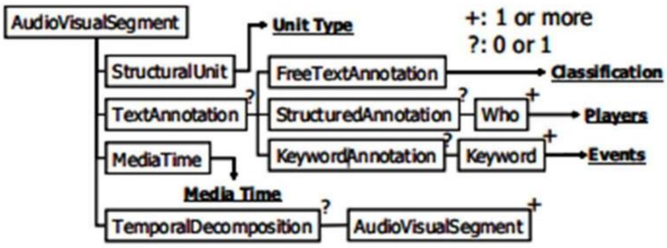
Relationship Among The Above 4 Variables in This article
The effectiveness of video summaries (dependent variable) is influenced by metadata-based significance measures and user-specified parameters (independent variables) in the play scene selection process.

Input and Output	Feature of This Solution	Contribution & The Value of This Work
Input: Video **Output**: Two types of video summaries: video clips and video posters	Efficient Sports Video Summarization	Good to have this knowledge from this paper as we reviewing of ideologies for developing cricket sports summarization using neural networks.

Input	Output
Video	Two types of video summaries: video clips and video posters

Positive Impact of this Solution in This Project Domain	Negative Impact of this Solution in This Project Domain
Enhanced accessibility and retrieval of important sports video content.	Possible reduction in revenue for platforms providing full-length sports broadcasts.

Analyse This Work By Critical Thinking	The Tools That Assessed this Work	What is the Structure of this Paper
This work introduces an innovative approach to video summarization for large sports	Metadata, MPEG-7 Standard, Video Summarization,	Abstract I. Introduction

archives, prioritizing play scenes based on metadata. The proposed methods, including video clips and posters, offer flexible options for users. Experimental results demonstrate promising effectiveness. However, the study is limited to baseball videos, and generalization to other sports may require adjustments.	Video Clips, Video Posters, Z- Score, Tree Structures, Greedy Method, Play-Cut Method, Evaluation Metrics	II. Metadata for Sports Videos III. Video Summarization IV. Experiments V. Conclusion
Diagram/Flowchart		
 		

---End of Paper 3---

4		
Reference in APA format	M. Z. Khan, S. Jabeen, S. ul Hassan, M. A. Hassan, and M. U. G. Khan, "Video Summarization using CNN and Bidirectional LSTM by Utilizing Scene Boundary Detection," 2019 International Conference on Applied and Engineering Mathematics (ICAEM), Taxila, Pakistan, 2019, pp. 197-202, doi: 10.1109/ICAEM.2019.8853663.	
URL of the Reference	**Authors Names and Emails**	**Keywords in this Reference**
[https://ieeexplore.ieee.org/document/885 3663](https://ieeexplore.ieee.org/document/8853663)	Muhammad Zeeshan Khan, Saira Jabeen, Saleet ul Hassan, M.A Hassan, Muhammad Usman Ghani Khan	Video Summarization, CNN, Bidirectional LSTM, Scene Boundary Detection, Multimedia Data, Deep Learning, Motion Features, TVSUM50 Dataset, F Measure Score, Video Retrieval
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	**The Goal (Objective) of this Solution & What is the problem that need to be solved**	**What are the components of it?**

Video Summarization using CNN and Bidirectional LSTM by Utilizing Scene Boundary Detection	Develop a video summarization technique using CNN and Bidirectional LSTM with scene boundary detection to efficiently generate concise and informative summaries from multimedia data.	Components: Scene boundary detection using motion features, CNN for frame-level importance, bidirectional LSTM for redundancy removal, leveraging deep learning.

The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process

The proposed method begins with scene boundary detection using motion features. The CNN analyzes frames for importance in each scene. Bidirectional LSTM is employed to eliminate redundant frames. The approach aims to generate video summaries by capturing significant content and improving efficiency compared to traditional methods, achieving better F measure scores.

	Process Steps	Advantage	Disadvantage (Limitation)
1	Scene Boundary Detection: Identify scene changes using motion features.	Improved Content Relevance: The system, using scene detection, CNN, and LSTM, enhances content selection for a more relevant video summary.	Computational Complexity: The multi-step process, including motion features, CNN, and LSTM, may lead to longer processing times, posing computational challenges.
2	CNN Analysis: Assess frame importance in each scene using Convolutional Neural Network.	Temporal Dependency Handling: Bidirectional LSTM improves coherence by addressing temporal dependencies and reducing redundancy in the video summary.	Training Data Dependency: The system's effectiveness hinges on the quality and diversity of training data, impacting performance across varied video content types.
3	Frame Selection: Utilize Bidirectional LSTM to remove redundant frames.		

4	Summary Generation: Combine selected frames to generate efficient video summaries, outperforming traditional methods in terms of F measure scores.		

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
F-measure Score	Dataset Characteristics, CNN and LSTM Architecture		

Relationship Among The Above 4 Variables in This article
The F-measure score is dependent on the successful combination of scene boundary detection, CNN-based frame importance, and Bidirectional LSTM -driven redundancy removal, collectively determining the model's overall performance in video summarization.

Input and Output	Feature of This Solution	Contribution & The Value of This Work
<table><tr><th>Input</th><th>Output</th></tr><tr><td>Video data containing diverse scenes.</td><td>Summarized video with reduced redundancy</td></tr></table>	Efficient video summarization using motion features, CNN, and bidirectional LSTM for accurate content extraction and reduced redundancy.	Good to have this knowledge from this paper as we reviewing of ideologies for developing cricket sports summarization using neural networks.

Positive Impact of this Solution in This Project Domain	Negative Impact of this Solution in This Project Domain
Enhanced video summarization efficiency, benefiting content retrieval, indexing, and real-time applications.	Increased computational complexity and dependency on training data quality, potentially affecting processing time and performance across diverse video content.

Analyse This Work By Critical Thinking	The Tools That Assessed this Work	What is the Structure of this Paper
Innovative video summarization approach with motion-based scene detection, CNN, and LSTM, improving content relevance, but potential challenges in efficiency and generalization.	TVSUM50 Dataset, F - Measure Score	Abstract I. Introduction II. Literature Survey III. Proposed System IV. Dataset V. Results and Discussions

Diagram/Flowchart

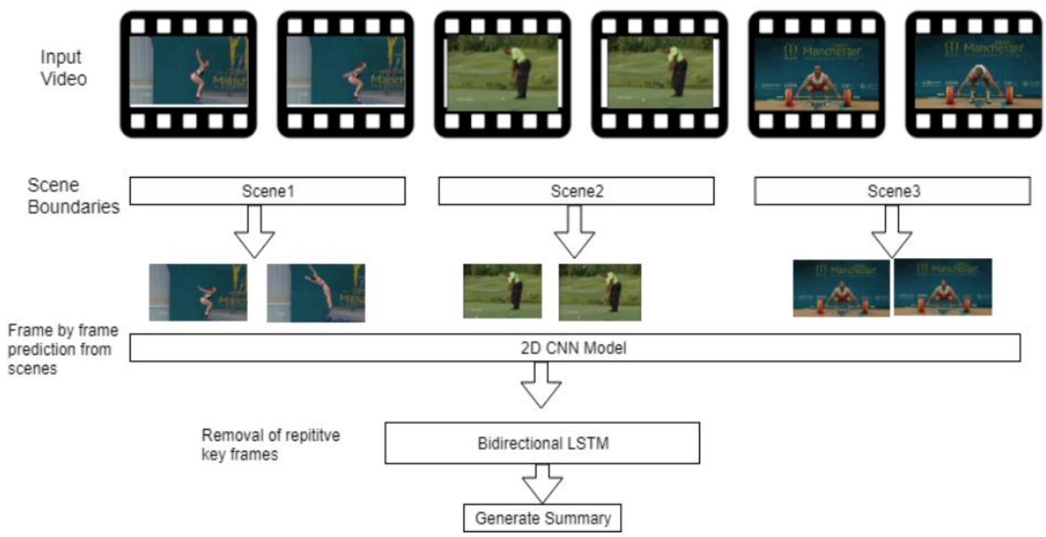


Fig. 1. Flow Diagram of Proposed Network

---End of Paper 4---

5

Reference in APA format	M. B. Andra and T. Usagawa, "Automatic Lecture Video Content Summarizationwith Attention-based Recurrent Neural Network," 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), Yogyakarta, Indonesia, 2019, pp. 54-59, doi: 10.1109/ICAIIT.2019.8834514.	
URL of the Reference	**Authors Names and Emails**	**Keywords in this Reference**
https://ieeexplore.ieee.org/document/8834514https://ieeexplore.ieee.org/document/8834514	Muhammad Bagus Andra, Tsuyoshi Usagawa	Summarization, Recurrent Neural Network (RNN), Attention-based, Lecture Video, Segmentation, Linguistic feature, ROUGE, Data-driven, Seq2seq model, NLP (Natural Language Processing).
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	**The Goal (Objective) of this Solution & What is the problem that need to be solved**	**What are the components of it?**
Automatic Lecture Video Content Summarization with Attention-based Recurrent Neural Network	The aim is to improve the quality of lecture video summaries, making them more informative and efficient for learners.	Preprocessing Module, Transcript Segmentation, Attention-Based RNN, Encoder-Decoder Architecture, LSTM Units, Attention Mechanism, Softmax Layer, Training (Epoch-based), Evaluation Metric (ROUGE), Cross-Entropy Loss Calculation

The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process
The system processes lecture video transcripts by cleaning noise, segmenting into coherent parts, and summarizing using an attention-based Recurrent Neural Network. This RNN captures essential content, leveraging linguistic features, yielding improved summaries compared to baseline models, validated through ROUGE evaluation.

	Process Steps	Advantage	Disadvantage (Limitation)
1	Preprocessing: Clean up the lecture text, removing unnecessary stuff and noting key features like word importance.	It helps you quickly understand lectures by summarizing them, so you don't have to go through the whole video.	It works best when the lecture transcripts are well-done. If they're messy or not structured, the system might not perform as well.
2	Segmentation: Break the text into logical parts using a method that considers phrases and word features.	It pays attention to what really matters in the lecture, creating summaries that make sense.	Doing the segmentation and summarization, especially with language features can use a lot of computer resources. This might make it hard to use on really big scales.
3	Summarization: Use a special type of neural network that pays attention to important words and structures to create a condensed summary		During training, the system might get too used to the training data. If not careful, it might not perform as well on new data; like learning a script too well but struggling with improve.
4	Evaluation: Check how well the summary matches with what humans would make, using a scoring system.		

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
ROUGE scores (ROUGE-1, ROUGE-2, and ROUGE-L)	Preprocessing techniques, transcript segmentation (PowerSeg method), attention-based RNN architecture.		

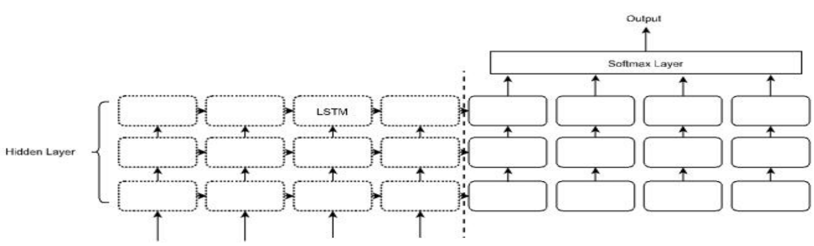
Relationship Among The Above 4 Variables in This article
The independent variables (Preprocessing, segmentation, attention-based RNN) influence the dependent variable (ROUGE scores), demonstrating how system components impact the quality of automatic lecture video summarization.

<table>
<tr><th>Input and Output</th><th>Feature of This Solution</th><th>Contribution & The Value of This Work</th></tr>
<tr><td><table><tr><th>Input</th><th>Output</th></tr><tr><td>Lecture video</td><td>Text-based summary of the lecture</td></tr></table></td><td>Automated lecture summarization with attention-based RNN, linguistic features, and segmentation improves content accessibility, context understanding, but may require careful transcript quality.</td><td>Good to have this knowledge from this paper as we reviewing of ideologies for developing cricket sports summarization using neural networks.</td></tr>
</table>

Positive Impact of this Solution in This Project Domain	Negative Impact of this Solution in This Project Domain
Enhances learning by providing efficient access to crucial lecture insights, supporting students in grasping key topics more effectively.	Resource-intensive processing may limit scalability in diverse educational settings.

Analyse This Work By Critical Thinking	The Tools That Assessed this Work	What is the Structure of this Paper
The work innovatively combines RNN and segmentation for lecture summarization, but challenges include transcript quality dependency and resource intensity, affecting practicality.	The tools used to evaluate this work include the ROUGE framework, which assesses the quality of text summarization through N-Gram recall.	Abstract I. Introduction II. Related Works III. Proposed Model IV. Experimental Setup V. Result and Discussion

Diagram/Flowchart



---End of Paper 5---

6		
Reference in APA format	S. H. Emon, A. H. M. Annur, A. H. Xian, K. M. Sultana and S. M. Shahriar, "Automatic Video Summarization from Cricket Videos Using Deep Learning," 2020 23rd International Conference on Computer and Information Technology (ICCIT), DHAKA, Bangladesh, 2020, pp. 1-6, doi: 10.1109/ICCIT51783.2020.9392707	
URL of the Reference	**Authors Names and Emails**	**Keywords in this Reference**
https://ieeexplore.ieee.org/document/9392707	Solayman Hossain Emon, A.H.M Annur, Abir Hossain Xian, Kazi Mahia Sultana, Shoeb Mohammad Shahriar	Video summarization, Deep Cricket Summarization Network (DCSN), Deep Reinforcement Learning, LSTM, Convolutional Neural Networks, Recurrent Neural Networks, Reward Function, Diversity Reward, Supervision Signal, Representativeness Reward, Maximum Likelihood Estimation
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	**The Goal (Objective) of this Solution & What is the problem that need to be solved**	**What are the components of it?**
Deep Cricket Summarization Network (DCSN)	Aim is to develop a summarization network that can automatically extract and select the most important moments from a cricket match video for creating concise summaries that capture the key moments.	Author used deep learning models DCSN, CricSum Dataset, CNN for feature extraction, supervision signals to help train the summarization network, Reinforcement Learning to optimize the summarization process by diversity and representative reward functions, F1-score and Mean Opinion Score for objective and subjective evaluation metrics.
The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process		
The proposed system Deep Cricket Summarization Network (DCSN) is an encoder-decoder architecture that predicts frame-level probabilities for video summarization. It uses		

CNN, LSTM, and reward functions to optimise the summary identifying key moments. Even though this author compared various results upon validating the test data and trained data using machine learning with all supervised, unsupervised and deep reinforcement learning algorithms.

	Process Steps	**Advantage**	**Disadvantage (Limitation)**
1	Data collection and training with CricSum dataset by Reinforcement Learning combining Diversity-representative reward functions	Reduces redundancy, recognizes key events, and improves summary quality. The supervision signals guide the model in understanding what content is essential in the videos.	With limited learning capacity, the model's ability to learn complex features, behaviours might be restricted and may results limited diversity and representativeness.
2	The frame visual features from video are extracted using CNN.	Extracts features, captures complex patterns, Understands the content of the video.	Computationally expensive.
3	The encoder-decoder network DCSN uses frame level features and supervision signals.	The supervision signals guide the model in understanding what content is essential in the videos.	.
4	Diversity Representative reward functions are used to evaluate the quality of generated video summaries by selectin of non-redundant frames	Provides quantitative measures of summary quality. MOS scores reflect human perception of summary quality.	Objective metrics may not fully capture the quality of the summaries.

Major Impact Factors in this Work

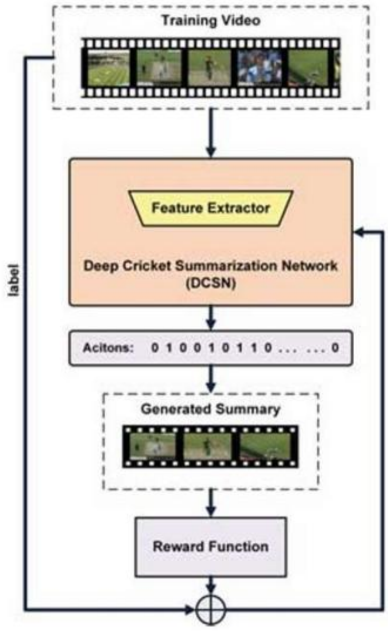
Dependent Variable	**Independent Variable**	**Moderating variable**	**Mediating (Intervening) variable**
Key moments extraction, Frame level probabilities, Video Summary.	Videos, human annotations	Video quality, Match context	Diversity Reward Function, Representativeness Reward, Supervision signal, CNN, Bi-LSTM.

Relationship Among The Above 4 Variables in This article
The independent variables (visual features) directly influence the quality of video summarization (dependent variable). The moderating variables influence the strength of relationship of dependent and independent variables. And while mediating variables how certain features affect the summarization outcome.

Input and Output	Feature of This Solution	Contribution & The Value of This Work
<table><tr><th>Input</th><th>Output</th></tr><tr><td>Cricket Video</td><td>Video Summary</td></tr></table>	Developing diversity-representative functions that can help in generate summaries include wide range of content while still capturing key moments. And developing supervision signals help in selecting relevant frames.	Good to have this knowledge from this paper as we reviewing of ideologies for developing cricket sports summarization using neural networks

Positive Impact of this Solution in This Project Domain	Negative Impact of this Solution in This Project Domain
Deep learning algorithms are big challenging in the current research. Reward functions and supervision signals improved time and resource efficiency.	Since this is a performance improvement algorithm, not much to project on negative side as all the things used are defined in advance.

Analyse This Work By Critical Thinking	The Tools That Assessed this Work	What is the Structure of this Paper
This work is good, as they tried improving the performance of DCSN using reward functions and supervision signals.	Tensorflow, keras, CNN, LSTM, numpy, pandas	Abstract I. Introduction II. Related Works III. Proposed Method IV. Experimental Setup V. Experimental Results and Evaluation VI. Conclusion VII. Future work

Diagram/Flowchart


---End of Paper 6---

7

Reference in APA format	Shingrakhia, Hansa, and Hetal Patel. "SGRNN-AM and HRF-DBN: a hybrid machine learning model for cricket video summarization." The Visual Computer 38, no. 7 (2022): 2285-2301.	
URL of the Reference	**Authors Names and Emails**	**Keywords in this Reference**
Video summarization based on SGRNNAM	Hansa Shingrakhia, Hetal Patel	Summarization, Key Events, Stacked Gated Recurrent Neural Network Attention Module (SGRNN-AM), Hybrid Rotation Forest Deep Belief Network (HRF-DBN).
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	**The Goal (Objective) of this Solution & What is the problem that need to be solved**	**What are the components of it?**
SGRNN-AM+HRF-DBN +OCR	To create an automated and efficient method for summarizing cricket videos. The problems need to be solved are: Lengthy	Audio-based Excitement Detection, Speech to Text Framework, Cumulative Key Frame Estimation, Key Frame Extraction, HRF-DBN Classifier, Scorecard Region

	duration, Complexity of content, Identification of key events, Heterogeneous Data Sources	Detection, Action Recognition Model, SGRNN-AM, Temporal Feature Extraction.

The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process

The proposed cricket video summarization model operates by initially detecting excitement through audio analysis and speech to text frameworks. It then classifies shots using hue histogram differences, employs OCR for scoreboard analysis, recognizes umpire gestures via joint-based and temporal dependent features, and integrates these extracted features into a SGRNN-AM for summarization, capturing key events like fours, sixes, and wickets for a concise representation of the match.

	Process Steps	**Advantage**	**Disadvantage (Limitation)**
1	Exciting clip is extracted using audio cues and speech to text frameworks to detect excitement.	Provides auditory context to thrilling moments.	Distinguishing between genuine excitement and background noise.
2	Shot boundary detection and classification based on hue histogram differences.	Separates different shot types (long, close). Provides visual cues for changing scenes.	Might fail in certain lighting conditions.
3	Extracting score and wickets information from scorecards using OCR. And recognizes umpire gestures by pose estimation by Deeper Cut approach.	Extracts critical information on frame.	May struggle with unclear visuals.
4	Integrating extracted features (audio excitement, shot classifications and scorecard data) to create video summary.	Captures both auditory and visual cues for video summarization.	Sensitive to clarity for accurate extraction and classification.

Major Impact Factors in this Work

The factors in this work contributes in audio and visual feature integration, umpire gesture recognition, leveraging cricket domain knowledge enhancing the accuracy of video summarization.

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
Excitement Segmentation, Shot Classification, key events, video summarization.	Audio Features, Visual Features, Textual Features.	Match Context, Video Quality and Clarity, Crowd Engagement level.	Feature extraction techniques (Hue Histogram Differences, OCR, Deep Cut approach), attention model in SGRNN-AM.

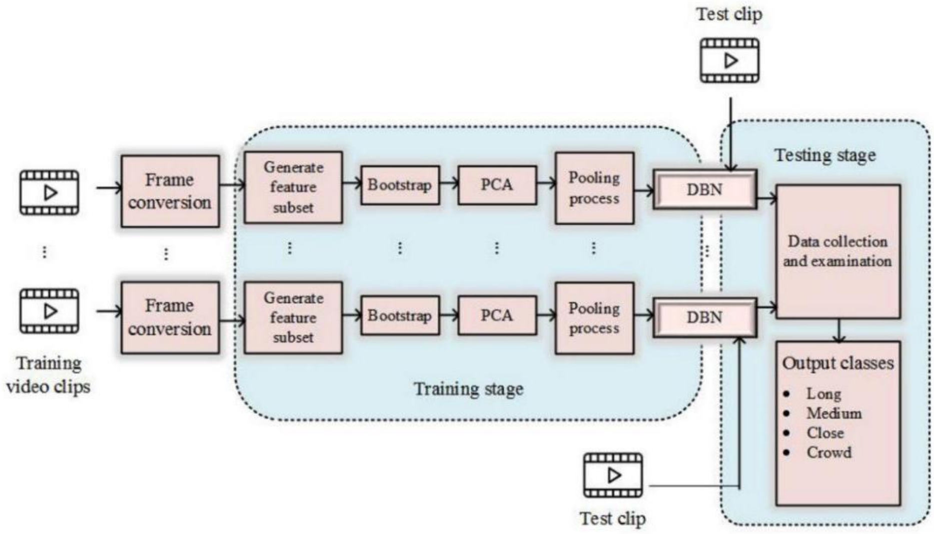
Relationship Among the Above 4 Variables in This article
The independent variables (audio, visual, textual features) directly influence the quality of video summarization (dependent variable). The moderating variables influence the strength of relationship of dependent and independent variables. And while mediating variables how certain features affect the summarization outcome.

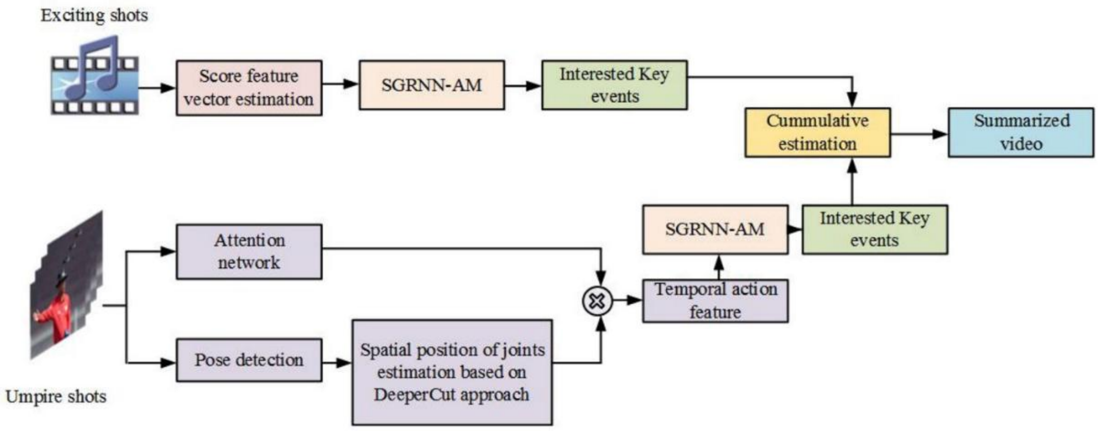
Input and Output	Feature of This Solution	Contribution & The Value of This Work
<table><tr><th>Input</th><th>Output</th></tr><tr><td>Cricket video</td><td>Summarized Video</td></tr></table>	Audio analysis, visual feature extraction using OCR, Umpire Gesture Detection, Shot Boundary detection.	This work contribution advances the field of cricket video summarization by leveraging hybrid approach, integrating various data sources and advanced models to provide and comprehensive cricket video summary.

Positive Impact of this Solution in This Project Domain	Negative Impact of this Solution in This Project Domain
The positive impact of this solution is utilizing the video content efficiently which improves the video analysis.	Delay in computations, could not predict the key events accurately. But using CNN instead of computer vision techniques has advantages in enhanced visual feature extraction, identify complex patterns, improved classification accuracy.

Analyse This Work By Critical Thinking	The Tools That Assessed this Work	What is the Structure of this Paper
The proposed approach shows its attempt to integrate multiple modalities for video summarization.	TensorFlow, Keras, Scikit, Numpy, Pandas, Opencv, NLTK, audio processing tools, OCR, GRU.	Abstract I. Introduction II. Related Work III. Proposed Method IV. Experiment Results V. Conclusion

Diagram/Flowchart





---End of Paper 7---

8		
Reference in APA format	Besta Srikanth, Sagarla Aravind, Mopuri Veera Narayana, Narayana Satya Narayana, “Sports Match Video to Text Summarization Using Neural Network.”, 2023 INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY (IJIRT).	
URL of the Reference	**Authors Names and Emails**	**Keywords in this Reference**
IJRTI2305007.pdf	Besta Srikanth, Mopuri Veera Narayana,	3-D CNN, LSTM, Residual Network, Neural Network,

	Narayana Satya Narayana, Sagarla Aravind	feature selection, ResNet34, Mean Opinion Score.
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	**The Goal (Objective) of this Solution & What is the problem that need to be solved**	**What are the components of it?**
Combination of a 3D-ResNet34 and a LSTM Neural Network Model	Converting Soccer video to text summarization by selecting keyframes.	Manual Annotation, Feature Extraction, LSTM Network, Evaluation.
The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process		

	Process Steps	Advantage	Disadvantage (Limitation)
1	Data collection and training with 744 football clips.	.	Low video quality effects the model training.
2	Feature Extraction using 3D-ResNet34 CNN to identify objects, key elements relevant to football activities.	Captures Spatial features and extracts hierarchical features which are essential for identifying football elements.	Computationally intensive.
3	LSTM captures sequence of highlights events and helps in summarizing the video content over time.	Captures the sequential patterns and temporal dependencies from video over time.	Computationally intensive.
4	Highlight ranking network ranks the highlight videos according to a highlight similarity score against a truth.	Relevance assessment and object evaluation.	Choosing an appropriate truth might results biases and subjectivity.
5	Combines the highest ranked highlights.	Emphasizes the most relevant and impactful moments, improving the final summary.	Balancing diversity of highlights and quality without redundancy is a challenge.

6	Event recognition and concise video summary generation by combining CNN and LSTM.		Important context of gameplay may be omitted in the summary.

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
Identification of key highlights, Concise Video Summary Quality.	Videos, text data	Dataset Characteristics such as size, diversity, quality and bias.	CNN ResNet34, LSTM, Highlight Ranking Network.

Relationship Among The Above 4 Variables in This article
The quality of the feature extraction, temporal understanding through LSTM, relevance assessment by the ranking network, along with influence of moderating and mediating variables, collectively shape the effectiveness and quality of the generated video summary

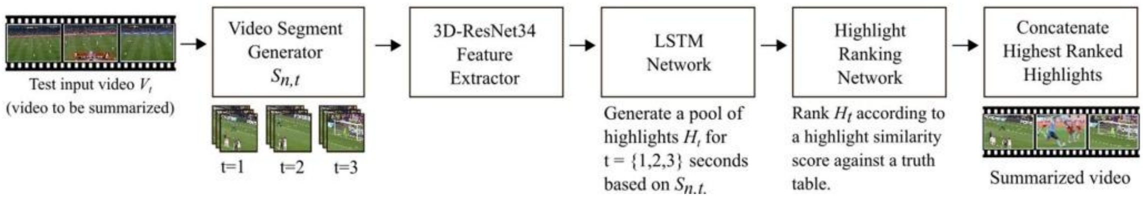
Input and Output	Feature of This Solution	Contribution & The Value of This Work
<table><tr><th>Input</th><th>Output</th></tr><tr><td>Soccer Match Videos</td><td>Text Summarization</td></tr></table>	Can be derivable to other domains as well.	To the extent this work is a simple approach for Soccer Sports Video to Text Summarization using Neural Networks.

Positive Impact of this Solution in This Project Domain	Negative Impact of this Solution in This Project Domain
Generates highlights of soccer match	Since this is a general approach for implementation, not much to project on negative side. But the proposed methodology is not evaluated.

Analyse This Work By Critical Thinking	The Tools That Assessed this Work	What is the Structure of this Paper
This work is simple as there is no optimization and evaluation for the proposed methodology.	TensorFlow, keras, 3D-ResNet34 CNN, LSTM.	Abstract I. Introduction II. Literature Review III. Existing System

		IV. Proposed Methodology V. References

Diagram/Flowchart



---End of Paper 8---

9		
Reference in APA format	Guntuboina C, Porwal A, Jain P, Shingrakhia H. Deep learning based automated sports video summarization using YOLO. ELCVIA Electronic Letters on Computer Vision and Image Analysis. 2021 May 27;20(1):99-116.	
URL of the Reference	**Authors Names and Emails**	**Keywords in this Reference**
https://doi.org/10.5565/rev/elcvia.1286	Chakradhar Guntunboina, Aditya Porwal, Preet Jain, Hansa Shingrakhia	Sports video, Image detection, Image Processing, Optical Character Recognizer (OCR), YOLO, Intersection Over Union, Region of Image, Mean Average Precision, Key-events.
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	**The Goal (Objective) of this Solution & What is the problem that need to be solved**	**What are the components of it?**
YOLO+OCR	The identification of scores is accomplished by using YOLO and customized CNN.	YOLO, Image Processing techniques, OCR, F1-Score.
The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process		

	Process Steps	Advantage	Disadvantage (Limitation)
1	A sports video is given as input to pretrained YOLO model to detect whether scoreboard is present.	Efficient object detection.	Requires intensive computational resources. It might struggle with variations in scoreboard designs.
2	If scoreboard region is present, it is detected and cropped and extracted.	Allows for precise extraction of scoreboard using OCR.	
3	Text information (score) is extracted from scoreboard using Optical Character Recognition (OCR) consisting of 1 CNN layer.	Extracts textual information and OCR can adapt to various font styles and sizes.	OCR can be time consuming, especially with noisy images.
4	The scores are recorded at timestamps, if the scores differ then considered as key event.	Temporal association of scores with timestamps are extracted. Event detection is identified based on score differences.	Requires efficient storage and management.

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
Scoreboard Detection and extraction, OCR accuracy Key event identification	Scoreboard, Textual information on frame.	Scoreboard variation, video quality	OCR performance, relationship between scoreboard region extraction and accurate score extraction.

Relationship Among The Above 4 Variables in This article
In this model, successful extraction of the scoreboard region (independent variable) is influenced by accurate YOLO- Based scoreboard detection, subsequently impacting precision of OCR score extraction. The accuracy of OCR (independent variable) directly affects the identification of score differences, which, in turn, influences the detection of key events (dependent variable), within the video timeline.

<table>
<tr><th colspan="2">Input and Output</th><th colspan="2">Feature of This Solution</th><th>Contribution & The Value of This Work</th></tr>
<tr><td colspan="2"><table><tr><th>Input</th><th>Output</th></tr><tr><td>Sports Video</td><td>Scores with timestamps</td></tr></table></td><td colspan="2">Key events like scores are identified using scoreboard.</td><td>The work and its approach provide efficient results in extracting key events of score which can be further used for match analysis.</td></tr>
<tr><th colspan="3">Positive Impact of this Solution in This Project Domain</th><th colspan="2">Negative Impact of this Solution in This Project Domain</th></tr>
<tr><td colspan="3">This solution can be adapted to various sports like Soccer, Kabaddi.</td><td colspan="2">Since this is a simple algorithm, not much to project on negative side as all the things used are defined in advance.</td></tr>
<tr><th colspan="2">Analyse This Work By Critical Thinking</th><th colspan="2">The Tools That Assessed this Work</th><th>What is the Structure of this Paper</th></tr>
<tr><td colspan="2">This work is a promising deep learning application in video analysis for capturing key moments, providing potential value also in other domains.</td><td colspan="2">YOLO, OpenCV, Python Image Library, Tesseract OCR.</td><td>Abstract
I. Introduction
II. Literature
III. YOLO
IV. Methodology
V. Results
VI. Conclusion</td></tr>
<tr><th colspan="5">Diagram/Flowchart</th></tr>
<tr><td colspan="5">

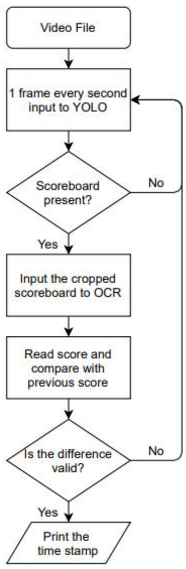


Figure 1: Flow Chart of the Algorithm

</td></tr>
</table>

---End of Paper 9---

10		
Reference in APA format	Dilawari, Aniqa and Muhammad Usman Ghani Khan. "ASoVS: Abstractive Summarization of Video Sequences." IEEE Access 7 (2019): 29253-29263.	
URL of the Reference	**Authors Names and Emails**	**Keywords in this Reference**
https://ieeexplore.ieee.org/abstract/document/8664480	Aniqua Dilawari, Muhammad Usman Ghani Khan	Abstractive summarization, attention, human evaluation, LSTM, METEOR, multi-line video captioning, multi-task feature learning, VGG-16.
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	**The Goal (Objective) of this Solution & What is the problem that need to be solved**	**What are the components of it?**
Abstractive Summarization of Video Sequences	It aims to automatically understand the semantics embedded within videos and convert visual information into text information. The problem it addresses is the need to quicker access to relevant information from vast amounts of videos.	CNN, LSTM, LSTM Encoder-Decoder, Attention Mechanisms

The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process

The ASoVS model uses CNN for feature attraction, LSTM for sequential understanding, attention mechanisms for context focus, pointer-generators for vocabulary handling, and human evaluation for subjective assessment.

	Process Steps	Advantage	Disadvantage (Limitation)
1	Visual Features are extracted using CNN.	Model can extract hierarchical features from visual data.	Requires extensive computations resources
2	Multi-Line textual descriptions of videos from features are generated using LSTM.	LSTM captures sequential data effectively.	Increased complexity with multiple layers.

3	Abstractive Text Summarization (ATS) model uses LSTM with attention mechanisms and pointer-generation networks to focus on relevant content and handle out-of-vocabulary words.	Improves summary quality, handles vocabulary gaps.	Complexity in training with the attention mechanisms.
4	Generated text summaries are evaluated by human assessment.		Subjective nature prone to bias, Time consuming.

Major Impact Factors in this Work

The major impact factors in this work include the utilization of Deep Neural Networks like CNN, LSTM along with attention mechanisms and pointer-generator networks for context-based summarization

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
Evaluation based on metrics like METEOR scores, ROGUE scores and Human evaluations.	Videos, transcripts.	Complexity of videos, and the human evaluator’s subjectivity in assessing the generated summaries.	CNN, LSTM, Attention mechanisms, pointer-generation network.

Relationship Among The Above 4 Variables in This article

Videos and transcripts are the independents, driving the model to produce descriptive text. The model serves as a mediator, learning from these inputs to generate the textual output.

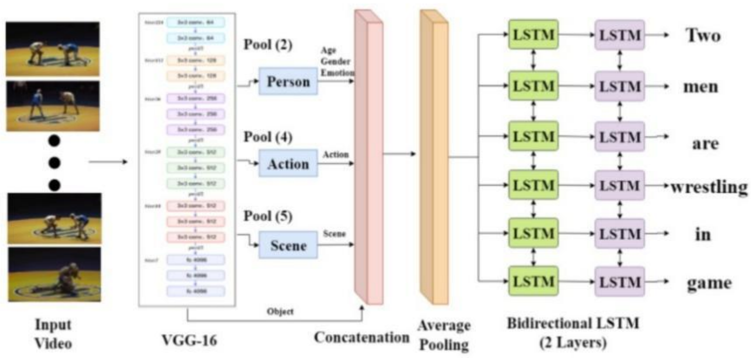
Input and Output	Feature of This Solution	Contribution & The Value of This Work
Input: Videos, Transcripts. **Output**: Descriptive textual summary	This work features an effective approach for generating abstract text summary from videos using deep neural networks CNN, LSTM along with attention mechanisms and pointer generation network.	The work contributes an effective approach for comprehensive video understanding and summary generation by bridging the visual data and textual data for efficient generation of concise summaries.

Input	Output
Videos, Transcripts.	Descriptive textual summary

Positive Impact of this Solution in This Project Domain	Negative Impact of this Solution in This Project Domain
This work approach impacts streamlines content understanding and facilitates faster data processing in multimedia contexts. This approach is applicable for a specific domain video to text summarization.	Improper training of model leads to subjective text summarizations.

Analyse This Work By Critical Thinking	The Tools That Assessed this Work	What is the Structure of this Paper
This work displays an innovative approach by combining video understanding with text summarization, leveraging deep learning techniques.	Tensorflow, keras, VGG-16, LSTM, METEOR, ROGUE and human evaluations.	Abstract I. Introduction II. Literature Review III. Methodology IV. Experimental Settings V. Implementation Details VI. Results VII. Human Evaluation VIII. Conclusion

Diagram/Flowchart



---End of Paper 10---

11	
Reference in APA format	Abhishek Yadav, Anjali Vishwakarma, Shyama Panickar, Prof. Satish Kuchiwale, "Real Time Video to Text Summarization using Neural Network", 2020 International Research Journal of Engineering and Technology (IRJET).

URL of the Reference	Authors Names and Emails	Keywords in this Reference
https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Real+Time+Video+to+Text+Summarization+using+Neural+Network&btnG=#d=gs_qabs&t=1698414040959&u=%23p%3DoWhZ14OzbyUJ	Abhishek Yadav Anjali Vishwakarma Shyama Panickar Satish Kuchiwale	Video captioning, Caption signals, Semantic representations, Video summarization, Convolutional Neural Network, Recurrent Neural Networks, Long short-term memory network, Tensorflow and Keras, Visual Geometry Group (VGG-16), BLEU score.
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	**The Goal (Objective) of this Solution & What is the problem that need to be solved**	**What are the components of it?**
Real Time Video to Text Summarization using Neural Network	The goal is to develop a model for automatically identifying key frames from a real time video and annotating the video with captions to enable a rich and more concise summarization of the video.	Author used deep learning technologies and tools, including CNNs, RNNs (LSTM), TensorFlow, Keras, and specific datasets like Flickr 8K. These technologies and tools are employed to develop and train the model, preprocess images, generate captions, and evaluate the results.

The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process

It consists of several steps like firstly, Convolutional Neural Networks (CNNs) like Inception v3 and VGG-16 are used for feature extraction from images, which effectively capture image content. Then, Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, generate captions for the images, benefiting from their sequence-to-sequence capabilities. Data processing involves tokenization, simplifying text data for the model. The model is trained using TensorFlow and Keras on a dataset like Flickr 8K. Finally, the BLEU score is employed as an evaluation metric, providing a quantitative measure of caption quality. This comprehensive process presents a powerful approach to video summarization, but deals with computational intensity and data quality.

	Process Steps	Advantage	Disadvantage (Limitation)
1	The proposed system uses Convolutional Neural Networks (CNNs), such as Inception v3 and	CNNs are highly effective at extracting visual features from images, which is crucial for video summarization	Pre-trained CNN models might have been trained on a diverse dataset that does not precisely match the domain of the video

	VGG- 16, to extract features from images. These features represent important visual information.	and captioning.	content you are summarizing or captioning. Fine-tuning the models may be necessary to adapt them to your specific domain.
2	Recurrent Neural Networks (RNNs), in particular Long Short-Term Memory (LSTM) networks, are utilized for generating textual captions for the keyframes.	LSTMs, with their ability to maintain memory over longer sequences, can better understand and remember the context of the video, ensuring that the generated captions are contextually accurate.	
3	The model is initialized with an image and one word as input, and it generates the subsequent words in the caption based on its learned associations between images and text.	This approach allows for the incremental generation of captions, which means that the system can start generating captions as soon as the first frame becomes available.	Incremental captioning may also face challenges with rare or domain-specific words. If the model's vocabulary is not comprehensive or if it has not been fine- tuned for a specific domain, it may struggle to produce accurate captions for certain terms.
4	Performance is evaluated using metrics BLEU score for caption quality and content coverage metrics for video summarization.	These metrics provide quantifiable feedback on the system's performance, allowing developers and researchers to track improvements over time and identify areas for enhancement.	These metrics do not consider the user's perspective or experience. A high BLEU score or content coverage metric does not guarantee that viewers find the summaries or captions useful, engaging, or contextually relevant.

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
BLEU Score	Key Frames		

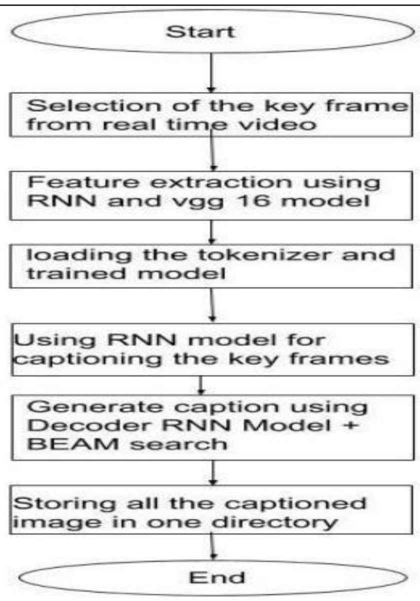
Relationship Among The Above 4 Variables in This article

Input and Output	Feature of This Solution	Contribution & The Value of This Work
Input: Video Output: Text Summary	The model that has been established is essential for producing video summaries that cover a wide range of content. This guarantees that the summaries produced are complete and do not limit themselves to specific parts of the video.	Good to have this knowledge from this paper as we reviewing of ideologies for developing cricket sports summarization using neural networks.

Positive Impact of this Solution in This Project Domain	Negative Impact of this Solution in This Project Domain
This solution significantly enhances the efficiency of video analysis, improves data accessibility, and generates informative video summaries in the project domain of video summarization and captioning.	Ethical concerns and the potential for misuse, especially in surveillance scenarios, can lead to privacy issues and public apprehension.

Analyse This Work By Critical Thinking	The Tools That Assessed this Work	What is the Structure of this Paper
The solution leverages deep learning techniques to automatically summarize video content, making it more accessible and manageable for users.	Flicker 8K Dataset Tensorflow Keras	Abstract I. Introduction II. Related Works III. Proposed Methodology IV. Experimental Results and Evaluation V. Conclusion

Diagram/Flowchart



---End of Paper 11---

12		
Reference in APA format	Joys Princia A, Ms. J Sangeetha Priya, Kalai Selvi J, Rithi Afra J, Rukshana S, "Video and Text Summarization Using VDAN and RNN",2021 INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY (IJIRT).	
URL of the Reference	**Authors Names and Emails**	**Keywords in this Reference**
https://ijirt.org/master/publishedpaper/IJIRT152248_PAPER.pdf	Joys Princia A Ms. J Sangeetha Priya Kalai Selvi J Rithi Afra J Rukshana S	Video Summarization, Text Summarization, VDAN (Visually Guided Document Attention Network), RNN (Recurrent Neural Network), Deep Learning, CNN (Convolutional Neural Network), LSTM (Long Short-Term Memory), Multimedia Analysis, Content Summarization, Sequence Data, Precision, Recall, F1 Score, Natural Language Processing, Information Retrieval.
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/	**The Goal (Objective) of this Solution & What is the problem**	**What are the components of it?**

Model/ Tool/ Framework/ ... etc)	that need to be solved	
Video and Text Summarization Using VDAN and RNN	The goal (objective) of the solution presented in the research paper "Video and Text Summarization Using VDAN and RNN" is to address the problem of efficiently summarizing both video and text content to provide users with a concise and informative representation of the original material.	Critical components: VDAN combines visual and textual information for summarization, CNN extracts visual features, RNN with LSTM handles text summarization, and actions (accelerate, decelerate, do nothing) control the process. Text preprocessing cleans and tokenizes data. Attention mechanism enhances focus on key details.

The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process
The process of solving the problem in "Video and Text Summarization Using VDAN and RNN" involves data preprocessing, extracting visual features with CNN, text summarization with RNN, and determining actions (accelerate, decelerate, do nothing) based on features. Advantages include improved data quality, complex visual understanding, human-like text summaries, and adaptability. However, disadvantages include potential data loss in preprocessing, resource-intense CNN, challenges with long-range text dependencies, and computational complexity with the attention mechanism.

	Process Steps	**Advantage**	**Disadvantage (Limitation)**
1	Clean and tokenize the text data, removing stopwords, and prepare it for summarization.	Preprocessing ensures that the summarization model operates on meaningful words and phrases. Consequently, it leads to higher- quality summaries that capture the essence of the text.	
2	Extract visual features from the video content using Convolutional Neural Networks (CNN).	CNNs excel at object recognition and can identify objects and actions within video frames, contributing to the selection of keyframes for the summary.	Fine tuning hyperparameters is crutial for the model to perform well in different domains.

3	Use Recurrent Neural Networks (RNN) for text summarization. RNN employs an encoder- decoder architecture and LSTM to generate coherent text summaries.	RNN-based models can be trained for various languages and domains, making them versatile for text summarization tasks across different types of content.	
4	Use an attention mechanism to select relevant information while discarding irrelevant content.	By focusing on relevant information, attention mechanisms reduce noise in the text and improve the quality of the summary.	It can lead to overfitting if not properly regularized. Overfit models may perform well on the training data but generalize poorly to unseen data.

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
F1 Score, Precision	Training Data, Input Data		

Relationship Among The Above 4 Variables in This article

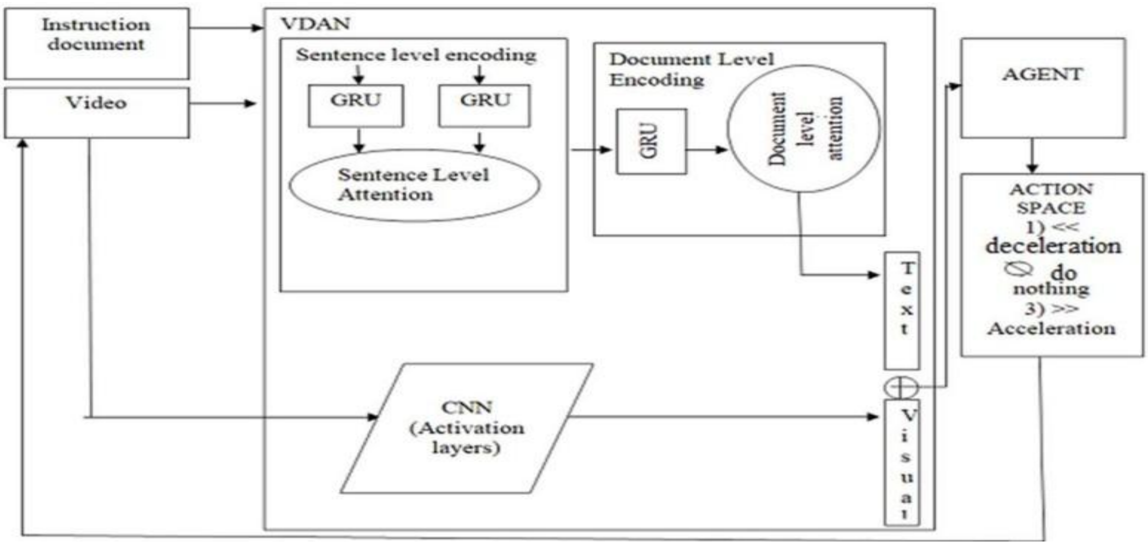
Input and Output	Feature of This Solution	Contribution & The Value of This Work
<table><tr><th>Input</th><th>Output</th></tr><tr><td>Video content, Textual content, Preproc-essed text</td><td>Summari-zed video, Summari-zed text</td></tr></table>	A highlighting feature of the solution is its adaptability through the use of action decisions based on extracted features. This feature allows the system to dynamically adjust the summarization process by choosing actions like accelerating, decelerating, or doing nothing. It enhances the system's ability for a wide range of multimedia content and	Good to have this knowledge from this paper as we reviewing of ideologies for developing cricket sports summarization using neural networks.

	user preferences.	

Positive Impact of this Solution in This Project Domain	Negative Impact of this Solution in This Project Domain
Positive impacts within the project domain include efficient content access, adaptability, deep learning advancements, and user-friendly text summaries, enhancing multimedia content understanding and knowledge acquisition.	Nothing new in terms of core logic. Used two algorithms which are alreadydefined.

Analyse This Work By Critical Thinking	The Tools That Assessed this Work	What is the Structure of this Paper
Innovative system for multimedia summarization, but resource demands and data loss risks should be considered for practical application.	VDAN Keras Tensorflow Matplotlib PyTorch	Abstract I. Introduction II. Literature Review III. Summarization System IV. Experiment Results V. Conclusion VI. Future Research

Diagram/Flowchart



---End of Paper 12---

13

Reference in APA format	Hansaraj Wankhede, Rachana Chawke, R Bharathi Kumar, Sushant Kawade, & Ashish Ramtekkar. (2023).

	AI-based Video Summarization using FFmpeg and NLP. International Journal of Innovative Science and Research Technology, 8(4), 1140–1145. https://doi.org/10.5281/zenodo.7888972	
URL of the Reference	**Authors Names and Emails**	**Keywords in this Reference**
https://ijisrt.com/assets/upload/files/IJISRT23APR1549.pdf	Hansaraj Wankhede R Bharathi Kumar Sushant Kawade Ashish Ramtekkar Rachana Chawke	Video Summarization, AI-based, FFmpeg, Natural Language Processing (NLP), AssemblyAI, Static Features, Motion Features, SumMe Dataset, Accuracy, Comprehensiveness, User Satisfaction, Data Split, Benchmarking, Video Content, Deep Learning, Ablation Study, NLP Fine-Tuning, Automation, Video Processing, Multimedia Summarization.
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	**The Goal (Objective) of this Solution & What is the problem that need to be solved**	**What are the components of it?**
AI-based Video Summarization usingFFmpeg and NLP	The goal of this solution is to create an efficient and accurate video summarization system using AI-based techniques, including FFmpeg, NLP, and AssemblyAI. The problem this solution aims to address is the huge amount of video content available, making it challenging for users to quickly and comprehensively understand the content without watching the entire video.	The components of the proposed video summarization solution include FFmpeg for video processing, Natural Language Processing (NLP) techniques for text generation, AssemblyAI for transcription, and the integration of motion and static features with an attention mechanism for video analysis.
The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process		
The proposed video summarization process comprises three key steps. First, it involves using FFmpeg to extract audio and frame data from the input video. Then, the extracted data is utilized by AssemblyAI to generate a preliminary text-based summary. Finally,		

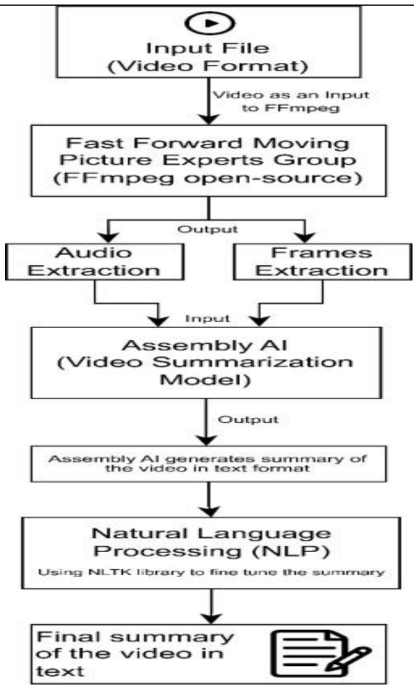
Natural Language Processing (NLP) techniques are employed to fine-tune the text summary, enhancing its accuracy and coherence. The advantages include efficient data extraction and automated text summarization, while potential disadvantages encompass inaccuracies in transcription and variations in NLP fine-tuning quality, impacting the overall quality of the video summary.

	Process Steps	Advantage	Disadvantage (Limitation)
1	FFmpeg is used to extract audio and frame data from the input video.	FFmpeg is a widely used and powerful tool for video and audio processing. It can handle a wide range of video codecs and formats, making it versatile for various types of videos.	
2	AssemblyAI generates a text-based preliminary summary using the extracted data.	Automated transcription ensures consistency in the summary generation process which can minimize errors.	The effectiveness of summarization can vary based on video content complexity.
3	Natural Language Processing (NLP) techniques are applied to enhance the accuracy and coherence of the initial summary.	It can maintain a consistent tone and style throughout the summary, making it more professional and easier to follow.	In some cases, NLP fine-tuning can lead to over-processing, resulting in summaries that are excessively verbose.
4	The process results in an efficient video summary, but potential disadvantages include inaccuracies in transcription and variations in NLP fine-tuning quality, which can affect the overall summary quality.		

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
Accuracy	Audio and Visua Features		

<table>
<tr><th colspan="4">Relationship Among The Above 4 Variables in This article</th></tr>
<tr><td colspan="4"></td></tr>
<tr><th>Input and Output</th><th colspan="2">Feature of This Solution</th><th>Contribution & The Value of This Work</th></tr>
<tr><td><table><tr><th>Input</th><th>Output</th></tr><tr><td>video content</td><td>text-based video, summary</td></tr></table></td><td colspan="2">A key feature of this solution is the integration of FFmpeg, NLP techniques, and AssemblyAI to efficiently generate accurate video summaries by combining audio and visual data analysis.</td><td></td></tr>
<tr><th colspan="2">Positive Impact of this Solution in This Project Domain</th><th colspan="2">Negative Impact of this Solution in This Project Domain</th></tr>
<tr><td colspan="2">This solution effectively summarizes videos, streamlining content analysis and enhancing user interaction through improved accuracy and efficiency.</td><td colspan="2">Negative impacts include variable NLP fine-tuning quality, content diversity challenges, and resource-intensive computational requirements.</td></tr>
<tr><th>Analyse This Work By Critical Thinking</th><th colspan="2">The Tools That Assessed this Work</th><th>What is the Structure of this Paper</th></tr>
<tr><td>The approach utilizes FFmpeg, NLP, and AI effectively for video summarization, yet more research is needed to address the nuances of different video types and enhance scalability.</td><td colspan="2">AssemblyAI</td><td>Abstract
I. Introduction
II. Related Work
III. Working on Video Summarization
IV. Experiment Results
V. Summary
VI. Conclusion</td></tr>
<tr><th colspan="4">Diagram/Flowchart</th></tr>
</table>



---End of Paper 13---

14

Reference in APA format	J. Mun, L. Yang, Z. Ren, N. Xu and B. Han, "Streamlined Dense Video Captioning," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 6581-6590, doi: 10.1109/CVPR.2019.00675	
URL of the Reference	**Authors Names and Emails**	**Keywords in this Reference**
https://ieeexplore.ieee.org/document/8953594	Jonghwan Mun Linjie Yang Zhou Ren Ning Xu Bohyung Han	Temporal Dependency Modeling, Event Proposal Network (EPN), Event Sequence Generation Network (ESGN), Sequential Captioning Network (SCN), Reinforcement Learning (RL), ActivityNet Captions Dataset, METEOR, CIDEr, BLEU (Evaluation Metrics), Event Detection and Caption Generation, Visual and Linguistic Context Modeling, Deep Neural Network Architecture.
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	**The Goal (Objective) of this Solution & What is the problem that need to be solved**	**What are the components of it?**

Streamlined Dense Video Captioning	The primary objective is to generate coherent and comprehensive captions for dense video content by effectively selecting event sequences, understanding temporal dependencies, and generating captions that form a cohesive storyline.	Three interconnected networks, Event Proposal Network (EPN) identifies potential event segments in a video, Event Sequence Generation Network (ESGN) arranges these events into a storyline, and Sequential Captioning Network (SCN) generates captions based on the events, considering both visual and linguistic contexts.

The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process

The proposed process comprises three components. First, the Event Proposal Network (EPN) adeptly identifies potential events in the data. Second, the Event Sequence Generation Network (ESGN) efficiently filters these proposals, yet faces challenges in sorting them based on temporal aspects. Lastly, the Sequential Captioning Network (SCN) leverages hierarchical RNNs to generate sequential captions based on the selected event sequences. Overall, while this method progressively enhances event selection and caption coherence in dense video data, it struggles with challenges related to proposal abundance and sequence sorting.

	Process Steps	Advantage	Disadvantage (Limitation)
1	Event Proposal Network (EPN): Identifies candidate event proposals in the video by selecting relevant segments and generating proposals based on visual representations.	Efficiently identifies event candidates and provides visual representations for further analysis.	May generate many proposals, leading to redundancy and increased computationalload.
2	Event Sequence Generation Network (ESGN): Sequentially selects a series of events from the candidate proposals to form an ordered sequence that represents the storyline of the video.	Adaptive selection of a sequence of events to form an episode, reducing the number of proposals while maintaining contextual relevance.	
3	Sequential Captioning Network (SCN): Utilizes a hierarchical recurrent neural network to generate	Hierarchical RNNs generate captions based on detected event sequences, enabling context- aware and	Relies heavily on the accuracy of the event sequence selection, potentially leading to errors in caption

	captions for the selected event proposals, considering both the visual context and linguistic context across the events in the sequence.	sequential caption generation.	generation.

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
BLEU Score	Extracted Features		

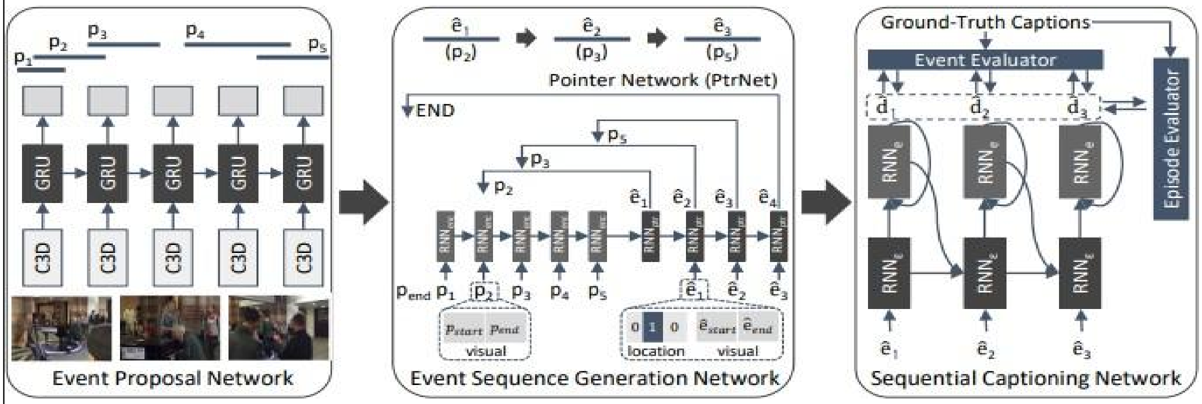
Relationship Among The Above 4 Variables in This article

Input and Output	Feature of This Solution	Contribution & The Value of This Work
Input: video; Output: Captions for each frame.	Employs event sequences, sequential captioning, and reinforcement learning to enhance dense video captioning, ensuring narrative coherence and contextual optimization.	Good to have this knowledge from this paper as we reviewing of ideologies for developing cricket sports summarization using neural networks.

Input	Output
video	Captions for each frame.

Positive Impact of this Solution in This Project Domain	Negative Impact of this Solution in This Project Domain
The model is further trained with reinforcement learning using two-level rewards (episode and event levels), improving the coherence and quality of captions generated.	It could potentially generate numerous suggestions, resulting in repetition and an increase in computational load.

Analyse This Work By Critical Thinking	The Tools That Assessed this Work	What is the Structure of this Paper
The proposed system leverages temporal dependency modeling and contextual captioning, yet faces challenges in scalability and complexity for broader adoption in real-world application.	ActivityNet Dataset Keras Tensorflow	Abstract I. Introduction II. Related Work III. Proposed System IV. Training V. Experiment Results VI. Conclusion

Diagram/Flowchart


---End of Paper 14---

15		
Reference in APA format	V.Vijayakumar and R.Nedunchezhian, "A Novel Method for Super Imposed Text Extraction in a Sports Video", International Journal of Computer Applications 15(1):1–6, February 2011.	
URL of the Reference	**Authors Names and Emails**	**Keywords in this Reference**
https://www.ijcaonline.org/volume15/number1/pxc3872553.pdf	V.Vijayakumar R.Nedunchezhian	Video Retrieval, Text Extraction, Superimposed Text, Sports Videos, Key Frame Extraction, OCR (Optical Character Recognition), Image Processing, Video Annotation, Edge Detection, Video Indexing
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	**The Goal (Objective) of this Solution & What is the problem that need to be solved**	**What are the components of it?**
A Novel Method for Super Imposed Text Extraction in a Sports Video	The goal of the proposed solution is to develop a method for effectively extracting superimposed text from sports videos. The	The method involves identifying important frames, converting them to grayscale for efficient processing, cropping out text regions like player details and scores,

	problem it addresses is the need to automatically detect, isolate, and extract textual information, such as player details and scores, which are typically added as overlays in sports video broadcasts.	detecting their edges, and using Optical Character Recognition to convert them into readable text for verification and indexing.
The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process		

The process involves key frame selection to reduce processing, grayscale conversion for efficient text detection, cropping text regions for focus, edge detection for precise boundaries, and Optical Character Recognition (OCR) for text conversion. Advantages include faster processing, accurate text detection, reduced data analysis, precise boundary identification, and automated text extraction. However, limitations include potential loss of color- related information, exclusion of relevant data outside cropped regions, sensitivity to noise in edge detection, and potential accuracy issues in OCR due to text quality or image resolution.

	Process Steps	**Advantage**	**Disadvantage (Limitation)**
1	Selecting key frames reduces computational load by focusing on essential frames.	Reduces processing time and resource requirements by analyzing only critical frames.	May miss details present in non-key frames that could be relevant.
2	Enhances text detection efficiency by converting frames to grayscale.	Simplifies image processing, making text detection more accurate and faster.	Potential loss of color-related information, which could be relevant in certain contexts
3	Focuses completely on regions where text, like scores and player details, is expected to appear. Accurately identifies the boundaries of the text regions using edge detection algorithms.	Reduces unnecessary data processing and simlifies analysis.	May exclude relevant information located outside the cropped regions.
4	Converts detected text regions into readable ASCII text for verification and indexing.	Enables automated text extraction and indexing, facilitating easy access to specific video content.	Accuracy may be affected by text quality or image resolution, impacting OCR performance.

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
Precision	Key Frames,Grayscale convertion		

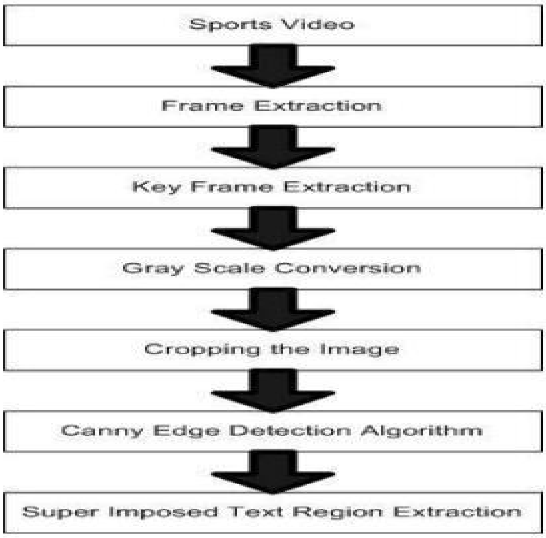
Relationship Among The Above 4 Variables in This article

Input and Output	Feature of This Solution	Contribution & The Value of This Work
<table><tr><th>Input</th><th>Output</th></tr><tr><td>video</td><td>Extracted Text</td></tr></table>	Efficiently processes key frames, precisely identifies text boundaries with edge detection, automates text extraction using OCR, and focuses on relevant player details and scores in sports videos for easier retrieval.	Good to have this knowledge from this paper as we reviewing of ideologies for developing cricket sports summarization using neural networks.

Positive Impact of this Solution in This Project Domain	Negative Impact of this Solution in This Project Domain
The positive impact lies in efficient data extraction, enabling quick access to crucial game details in sports videos, and facilitating faster retrieval of specific information for analysis and content summarization.	Potential negative impacts might include the loss of nuanced details in the process, such as color-related information due to grayscale conversion, and the exclusion of potentially relevant data outside the cropped text regions.

Analyse This Work By Critical Thinking	The Tools That Assessed this Work	What is the Structure of this Paper
This method efficiently extracts text from sports videos but might lose some detailed information due to grayscale conversion. While it focuses on relevant sections, excluding data outside those areas could limit the comprehensive understanding of the content.	Java OpenCV	Abstract I. Introduction II. Background III. Methodology IV. Experiment Results V. Conclusion

Diagram/Flowchart



---End of Paper 15---

16

Reference in APA format	Deep hierarchical LSTM networks with attention for video summarization Lin J., Zhong S.-H., Fares A.(2022) Computers and Electrical Engineering, 97,art.no. 107618	
URL of the Reference	**Authors Names and Emails**	**Keywords in this Reference**
https://doi.org/10.1016/j.compeleceng.2021.107618	Jingxu Lin , Sheng-hua Zhong , Ahmed Fares	Video summarization, Cost-sensitive learning, LSTM, Attention mechanism,
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	**The Goal (Objective) of this Solution & What is the problem that need to be solved**	**What are the components of it?**
Deep hierarchical LSTM networks with attention for video summarization(DHAVS)	The paper introduces DHAVS, a video summarization framework aiming to compress videos effectively. It employs a 3D ResNeXt-101 model for spatio-temporal feature extraction and an attention-based hierarchical LSTM module. DHAVS is evaluated using F-score and correlation coefficients, outperforming existing methods in summarizing videos.	The approach has a multi-faceted strategy for effective video summarization. Leveraging the power of a pre-trained 3D ResNeXt-101 model, it captures spatio-temporal features. The introduction of an attention-based hierarchical LSTM module enhances semantic understanding and temporal dependencies. To combat imbalanced class distribution, a cost-sensitive loss function is employed. The summarization process involves scene change detection through

		Kernel Temporal Segmentation (KTS), shot-level scoring, and a dynamic programming-based solution to the 0–1 Knapsack problem, ensuring both accuracy and computational efficiency in generating video summaries.

The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process

The proposed video summarization system, DHAVS, introduces a novel approach by incorporating a pre-trained 3D ResNeXt-101 model for spatio-temporal feature extraction and an attention-based hierarchical LSTM module to enhance semantic understanding. The system addresses imbalanced class distribution with a cost-sensitive loss function. Leveraging Kernel Temporal Segmentation (KTS) for scene change detection and a dynamic programming-based solution for summarization, DHAVS achieves competitive results. However, challenges include potential computational complexity due to the dynamic programming approach and sensitivity to certain hyperparameters, such as the misclassification cost (λ). Despite these, DHAVS offers a comprehensive solution to video summarization tasks

	Process Steps	**Advantage**	**Disadvantage (Limitation)**
1	Feature Extraction with 3D ResNeXt-101: Employ a pre-trained 3D ResNeXt-101 model to extract spatio-temporal features from the segmented video clips	Captures essential spatio-temporal features vital for video comprehension, providing richer representation than standard 2D CNNs, enhancing understanding of video content.	3D CNNs demand more computation, resulting in longer processing times due to their intricate architecture. This complexity can lead to overfitting and necessitate extensive hyperparameter tuning.
2	Attention-based Hierarchical LSTM: Implement a hierarchical LSTM module with attention mechanisms to capture semantic information and temporal dependencies.	Selective Focus enhances summary quality by concentrating on pertinent details. Meanwhile, addressing Long-range Dependencies surpasses LSTM in capturing intricate temporal relationships effectively.	Attention mechanisms elevate architectural complexity, complicating interpretation and adjustment. Improperly managed, they pose overfitting risks without adequate regularization or validation.
3	Model Training: Train the model using PyTorch with specific parameters (learning rate, dropout, weight decay) until the set maximum epochs are	Tailored models enable customization to suit summarization tasks, resulting in enhanced performance compared to generic models due to their fine-tuning for	Demands Resources: Needs substantial computational power and time for training, notably with big datasets or intricate models. Risk of Overfitting: Complex

	reached.	specific task characteristics.	models may overfit training data without proper regularization or validation.
4	Evaluation: Assess the model's performance using F-score, Kendall's τ, and Spearman's ρ correlation coefficients on SumMe and TVSum datasets, including an ablation study for further analysis.	Quantitative assessment delivers numerical metrics, creating clear benchmarks for performance comparison. Objective measurement reduces subjectivity by providing standardized metrics for summarization quality assessment	F-scores might miss the full video summary quality, lacking in comprehensiveness. Without subjective evaluation, crucial nuances essential for human understanding might be disregarded.

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
Performance Metrics , Model Performance , Time Efficiency , Correlation Coefficients	Video Features , Model Architecture , Training Parameters , Length of Final Summary (L)		

Relationship Among The Above 4 Variables in This article

Input and Output	Feature of This Solution	Contribution & The Value of This Work
Input: Summe dataset from "Creating summaries from user videos," **Output**: Video Summary	This solution introduces a video summarization method leveraging 3D ResNeXt-101 and attention-based LSTM models. It optimizes feature extraction, utilizes specific training strategies, and evaluates using F-score and correlation coefficients. The approach adeptly captures spatial and temporal details, resulting in succinct yet comprehensive video summaries.	This solution presents an advanced video summarization approach that efficiently incorporates spatio-temporal features. Its contribution lies in employing a 3D ResNeXt-101 model and attention-based hierarchical LSTM, providing precise video summarization, which is pivotal for various applications like surveillance, meeting summaries, and video retrieval

<table>
<tr><th colspan="2">Positive Impact of this Solution in This Project Domain</th><th colspan="2">Negative Impact of this Solution in This Project Domain</th></tr>
<tr><td colspan="2">This solution's positive impacts lie in its ability to generate precise video summaries through the integration of spatio-temporal features. It enhances applications in surveillance, meeting summaries, and video retrieval, enabling efficient content comprehension, retrieval, and analysis in these domains.</td><td colspan="2">The negative impact of this solution might encompass potential challenges in handling longer videos effectively due to summarization limitations. Additionally, dependence on pre-trained models could restrict adaptability to diverse datasets, potentially limiting its performance across varying content domains.</td></tr>
<tr><th>Analyse This Work By Critical Thinking</th><th colspan="2">The Tools That Assessed this Work</th><th>What is the Structure of this Paper</th></tr>
<tr><td>The study introduces a robust video summarization technique by combining a 3D ResNeXt-101 model and an attentionbased hierarchical LSTM module. It excels in leveraging cutting-edge architectures for feature extraction and capturing temporal dependencies. Yet, it relies on pre-trained models, lacks scalability for longer videos, and requires adaptation to diverse datasets. Despite these limitations, it marks a notable advancement in video summarization methods.</td><td colspan="2">PyTorch
NVIDIA Tesla V100</td><td>Title
I. Abstract
II. Introduction
III. Related Work
IV. Methodology/Approach
V. Experimental Setup
VI. Results and Discussion
VII. Conclusion
VIII. References</td></tr>
<tr><th colspan="4">Diagram/Flowchart</th></tr>
</table>

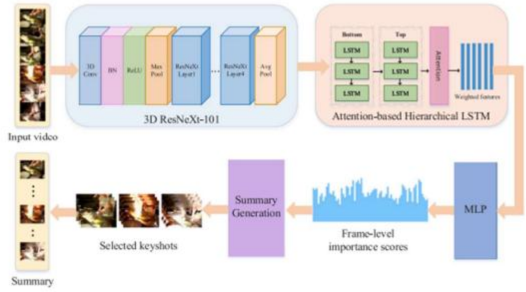


Fig. 1. An overview of our proposed DHAVS.

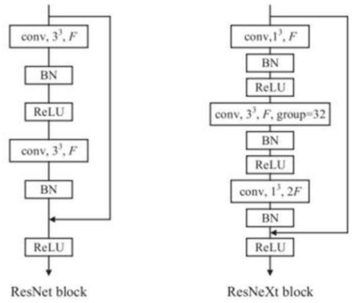


Fig. 2. **Left:** ResNet block. **Right:** ResNeXt block with 32 groups.

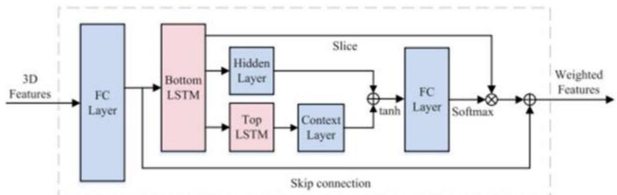


Fig. 3. Illustration of attention-based hierarchical LSTM module.

---End of Paper 16---

17

Reference in APA format	Video Summarization using Deep Semantic Features Mayu Otani, Yuta Nakashima, Esa Rahtu, Janne Heikkilä, Naokazu Yokoya ,16 pages, the 13th Asian Conference on Computer Vision (ACCV'16)	
URL of the Reference	**Authors Names and Emails**	**Keywords in this Reference**
https://doi.org/10.48550/arXiv.1609.08758	Mayu Otani , Yuta Nakashima , Esa Rahtu , Janne Heikkil , and Naokazu Yokoya	• Video Summary • Deep Features • Common Semantic Space • Summer Dataset • Joint Training
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	**The Goal (Objective) of this Solution & What is the problem that need to be solved**	**What are the components of it?**
Video Summarization using Deep Semantic Features	The paper aims to improve video summarization using deep semantic features from videos for better, more meaningful	The approach has a multi-faceted strategy for effective video summarization. Leveraging the power of a pre-trained 3D ResNeXt-101 model, it captures spatio-

	summaries. It addresses condensing lengthy videos into concise yet informative summaries, crucial for tasks like browsing collections or efficient retrieval. Manual summarization is time-consuming and subjective, prompting the need for automatic identification of relevant video segments, a complex challenge this solution tackles.	temporal features. The introduction of an attention-based hierarchical LSTM module enhances semantic understanding and temporal dependencies. To combat imbalanced class distribution, a cost-sensitive loss function is employed. The summarization process involves scene change detection through Kernel Temporal Segmentation (KTS), shot-level scoring, and a dynamic programming-based solution to the 0–1 Knapsack problem, ensuring both accuracy and computational efficiency in generating video summaries.

The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process

The study employs deep learning, extracting intricate video features via CNNs. It automates segment selection for summarization but faces challenges in relevancy determination. Algorithmic summarization aids automation, yet subjectivity remains. Objective metrics evaluate summary quality, applied in usercentric applications for enhanced browsing despite potential information loss.

	Process Steps	**Advantage**	**Disadvantage (Limitation)**
1	Deep Feature Extraction:Utilize Convolutional Neural Networks (CNNs) to extract intricate semantic features from videos	Captures complex visual information; enables understanding of video content at a deeper level	High computational cost; might require extensive data for effective feature learning.
2	Segment Identification: Automatically identify relevant video segments using extracted deep features.	Streamlines selection process; reduces manual effort and time in segment identification	Complexity in determining relevancy; potential oversight of important segments.
3	Summarization Algorithm:Process: Employ algorithms to generate video summaries based on	Automates summary creation; enhances efficiency and scalability.	May oversimplify or omit nuanced details; subjective aspects might affect the summary's accuracy.

	the identified segments		
4	Evaluation: Create objective metrics to evaluate video summaries accurately. Implement these summaries in user-friendly applications, enhancing efficient video browsing and retrieval experiences.	Provides measurable standards for assessment; facilitates comparison and improvement.	Might not encompass all subjective aspects; difficulty in capturing qualitative elements.

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
Quality of Generated Video SummariesF-measure (Evaluation Metric)	Video Features (Deep Semantic Features), Video Segments, Different Video Summarization Techniques, Hyperparameters and Training Techniques		

Relationship Among The Above 4 Variables in This article

Input and Output	Feature of This Solution	Contribution & The Value of This Work
<table><tr><th>Input</th><th>Output</th></tr><tr><td>Summe dataset from “Creating summaries from user videos,”</td><td>Video Summary</td></tr></table>	The solution features deep semantic feature extraction, automated segment identification, and algorithmic summarization. It employs objective metrics for quality assessment and integrates summaries into user-centric applications, enhancing user experience and enabling efficient video	This work presents a novel video summarization approach using deep semantic features learned from videos and descriptions. By employing these features, it generates more accurate video summaries, surpassing traditional methods. The approach provides insights into unsupervised learning for improved video content representation and summarization.

<table>
<tr><td></td><td colspan="2">browsing and retrieval.</td><td></td></tr>
<tr><th colspan="2">Positive Impact of this Solution in This Project Domain</th><th colspan="2">Negative Impact of this Solution in This Project Domain</th></tr>
<tr><td colspan="2">This solution's positive impact lies in revolutionizing video summarization by leveraging deep semantic features, surpassing conventional methods. It enhances summarization accuracy by effectively representing video content. Its unsupervised approach and successful incorporation of semantic features promise advancements in diverse applications, from content retrieval to automated video understanding, enriching various industries relying on video data.</td><td colspan="2">The solution's limitations include potential inaccuracies in summarizing shorter videos due to segment extraction constraints, leading to reduced performance. Moreover, it might struggle with videos containing extended unimportant sections or varied content, impacting the summarization quality. Dependency on unsupervised methods may occasionally hinder precise identification of crucial segments, affecting overall effectiveness.</td></tr>
<tr><th>Analyse This Work By Critical Thinking</th><th colspan="2">The Tools That Assessed this Work</th><th>What is the Structure of this Paper</th></tr>
<tr><td>The work introduces a novel video summarization method reliant on deep semantic features acquired via a contrastive loss-trained DNN. Although promising, its unsupervised nature and fixed-segment approach pose scalability and accuracy concerns. While innovative, its efficacy requires further validation across diverse datasets to ensure robustness and applicability in varied video scenarios.</td><td colspan="2">• t-SNE (t-distributed Stochastic Neighbor Embedding) for dimensionality reduction
• Deep Neural Networks (DNNs)
• Modified version of VGG (Visual Geometry Group) network</td><td>Title
I. Abstract
II. Introduction
III. Related Work
IV. Approach
V. Implementation Detail
VI. Experiment
VII. Conclusion</td></tr>
<tr><th colspan="4">Diagram/Flowchart</th></tr>
</table>

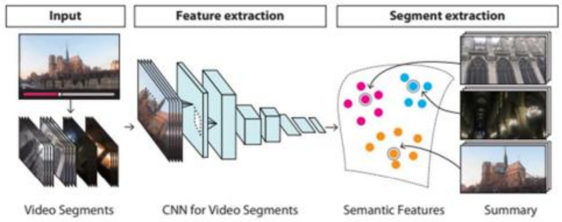


Fig. 2. Our approach for video summarization using deep semantic features. We extract uniform length video segments from an input video. The segments are fed to a CNN for feature extraction and mapped to points in a semantic space. We generate a video summary by sampling video segments that correspond to cluster centers in the semantic space.

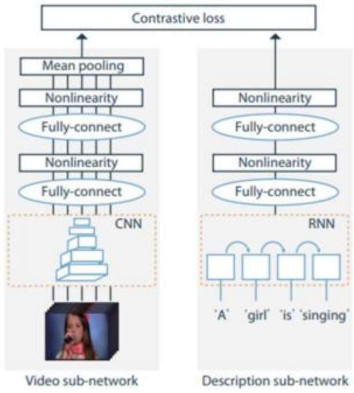


Fig. 4. The network architecture. Video segments and descriptions are encoded into vectors in the same size. Both sub-network for videos and descriptions are trained jointly by minimizing the contrastive loss.

---End of Paper 17---

18		
Reference in APA format	Self-Attention Based Generative Adversarial Networks For Unsupervised Video Summarization Maria Nektaria Minaidi, Charilaos Papaioannou, Alexandros Potamianos	
URL of the Reference	**Authors Names and Emails**	**Keywords in this Reference**
https://doi.org/10.48550/arXiv.2307.08145	Maria Nektaria Minaidi, Charilaos Papaioannou, Alexandros Potamianos	• Training, • Gallium nitride, • Decoding, • Feature extraction, • Machine learning, • Benchmark testing, • Neural networks
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	**The Goal (Objective) of this Solution & What is the problem that need to be solved**	**What are the components of it?**

Self-Attention Based Generative Adversarial Networks For Unsupervised Video Summarization	The objective is to enhance unsupervised video summarization through advanced GAN-based architectures, addressing the challenge of condensing extensive video content. By integrating attention mechanisms and transformers, the goal is to capture complex temporal dependencies and create accurate, concise video summaries for efficient content comprehension.	The solution utilizes a Generative Adversarial Network (GAN) comprising attention mechanisms, LSTM units, and a Variational AutoEncoder (VAE). This framework employs self-attention, transformers, and LSTM modules for encoding, decoding, and capturing long-term temporal dependencies, enhancing unsupervised video summarization by creating concise summaries from extensive video content.

The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process

The mechanism integrates a Generative Adversarial Network (GAN) with attention mechanisms (like self-attention and transformers), LSTMs, and a Variational Auto-Encoder (VAE). This approach leverages attention to capture long-term dependencies, while combining LSTM and transformer models to encode, decode, and select frames for generating accurate and concise video summaries in an unsupervised manner.

	Process Steps	**Advantage**	**Disadvantage (Limitation)**
1	Frame Selection:Use an attention-based mechanism (like self-attention) to identify crucial frames for video summarization	Self-attention aids in capturing long-term dependencies, highlighting crucial video segments, while accurate key frame selection ensures precise identification vital for summarization.	Attention weight management increases computational demands. Balancing computational efficiency and precise frame selection poses optimization challenges.
2	Encoder & Decoder: Utilize Long Short-Term Memory (LSTM) networks for encoding and decoding temporal relationships among frames.	LSTMs excel in capturing temporal patterns for sequence modeling, enabling robust encoding of complex temporal relationships among frames in sequential information handling.	LSTMs struggle with extensive dependencies, while transformers excel. Complex video content in LSTM encoding may induce overfitting due to sequence diversity.

3	GAN Training: Employ a Generative Adversarial Network (GAN) to train a summarizer and a discriminator simultaneous ly.	GANs enable joint training of summarizer and discriminator, enhancing summarization quality. The discriminator distinguishes original videos from generated summaries, improving fidelity in summarization.	GANs encounter convergence, mode collapse, and training instability. Balancing summarizer and discriminator training in GANs poses significant computational demands.
4	Variational Auto-Encoder (VAE): Incorporate a Variational Auto-Encoder (VAE) to generate underlying video representations and additional contextual information.	VAE improves training via representation learning, enhancing summarization quality. Contextual information supplements frame scores, improving accuracy in summary generation.	VAEs' inclusion may escalate computational burdens, elongating training. Balancing their impact amid hyperparameter tuning could demand significant effort due to sensitivity.

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable

Relationship Among The Above 4 Variables in This article

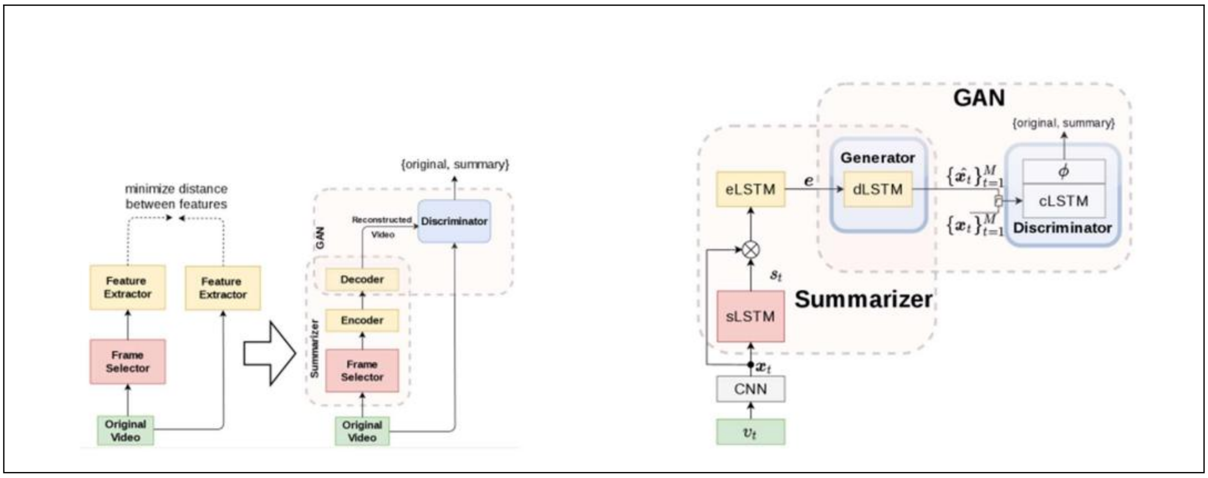
Input and Output	Feature of This Solution	Contribution & The Value of This Work
<table><tr><th>Input</th><th>Output</th></tr><tr><td>High Quality Video</td><td>Video Summary</td></tr></table>	The solution utilizes VAEs for extracting complex video features, emphasizing efficient computational handling. It balances VAE influence via adaptive hyperparameter tuning, employs objective metrics for quality evaluation, and integrates the	This work's contributions lie in leveraging VAEs for advanced video feature extraction, ensuring efficiency, and enhancing summarization quality through hyperparameter tuning. It introduces objective metrics and facilitates improved user interaction within applications, advancing video summarization capabilities for diverse real-world uses.

	summarization model into user applications for enhanced accessibility and interaction.	

Positive Impact of this Solution in This Project Domain	**Negative Impact of this Solution in This Project Domain**
The proposed solution for unsupervised video summarization through attention mechanisms and adversarial learning offers several positive impacts. It enhances the efficiency of video content consumption by autonomously extracting crucial frames, aiding in quicker comprehension of extensive video datasets. Additionally, it facilitates improved accessibility to relevant information within videos, benefiting various fields like media, education, and content curation, fostering streamlined information retrieval and knowledge extraction.	While the solution enhances video summarization, it poses potential drawbacks. These include computational demands due to attention mechanisms and adversarial training, potentially requiring substantial processing power and time. Moreover, the complexity of neural network architectures might lead to challenges in fine-tuning hyperparameters, making the implementation and optimization of the model intricate and resource-intensive.

Analyse This Work By Critical Thinking	**The Tools That Assessed this Work**	**What is the Structure of this Paper**
This work pioneers unsupervised video summarization by integrating attention mechanisms, LSTM, Transformer architectures, and GANs. While achieving state-of-the-art performance, the models' complexity may raise computational demands. Ensuring generalizability beyond benchmark datasets and enhancing interpretability are pivotal for practical deployment and broader applicability in diverse video contexts.	• PyTorch • GoogleNet • GitHub • Python • SLP Framework	Title I. Abstract II. Introduction III. Related Work IV. Proposed Method V. Experiments VI. Discussion VII. Conclusion VIII. References IX. Acknowledgments

Diagram/Flowchart



---End of Paper 18--

19		
Reference in APA format	Unsupervised video summarization framework using keyframe extraction and video skimming Shruti Jadon, Mahmood Jasim 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA) 10 November 2020	
URL of the Reference	**Authors Names and Emails**	**Keywords in this Reference**
https://doi.org/10.1109/ICCCA49541.2020.9250764	Shruti Jadon; Mahmood Jasim	Video Summarization, Vision, Deep Learning, Clusterring, Image processing
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	**The Goal (Objective) of this Solution & What is the problem that need to be solved**	**What are the components of it?**
Unsupervised video summarization framework using keyframe extraction and video skimming	The aim of the paper's solution is to improve video summarization by developing an algorithm that generates user-friendly video summaries, closely matching human preferences. The method involves testing different keyframe extraction and clustering techniques and	First,our approach involves diverse keyframe selection techniques, including uniform sampling, image histograms, SIFT, and deep learning-based ResNet16 on ImageNet. Clustering methods like Kmeans and Gaussian clustering categorize keyframes into interesting and uninteresting frames, emphasizing relevant content. Video summaries are created by selecting keyframes and

	adapting to diverse video types while acknowledging the inherent challenges in video summarization.	incorporating short skims for fluidity. Evaluation employs F-scores, comparing algorithm-generated summaries to human ones using the SumMe dataset. These components collectively improve video summarization, prioritizing user-friendliness and efficiency.

The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process

	Process Steps	Advantage	Disadvantage (Limitation)
1	Keyframe Selection: Various techniques are employed to select keyframes from the video frames, including uniform sampling, image histograms, Scale Invariant Feature Transform (SIFT), and deep learning-based feature extraction using ResNet16 on ImageNet.	Utilizing a range of keyframe extraction techniques enhances adaptability to diverse video content. Deep learning-based methods excel with intricate visuals, while uniform sampling serves as a reliable baseline. This flexibility empowers the system to handle videos with varying characteristics and complexities effectively.	Keyframe extraction techniques, particularly deep learning-based ones, can demand significant computational resources. Choosing the right method often involves time-consuming experimentation and tuning.
2	Clustering: The selected keyframes are then categorized into interesting and uninteresting frames using clustering methods, such as K-means and Gaussian clustering. This step helps filter out the most relevant content.	Clustering aids in filtering relevant content by grouping keyframes with important information, enabling customization of the summarization process through parameter and criteria adjustments.	The method selection challenge in clustering involves choosing the right method and parameters, with poor choices potentially leading to subpar results. Furthermore, clustering may lead to information loss if it fails to capture subtle nuances in video content.
3	Video Summarization: Video summaries are created by choosing keyframes that show important content. To ensure a smooth and	Video summarization enhances userfriendliness by using keyframes and short video skims, resulting in concise representations	Content loss can occur when keyframes don't capture essential information in video summarization. Striking a balance between brevity

	continuous summary, short video skims are added around these keyframes.	of video content, aiding quick comprehension of main points.	and content coverage is a challenge, as summary length varies with keyframe selection.
4	Evaluation: The effectiveness of the algorithm is assessed using F-scores. These scores measure how closely the algorithmgenerated video summaries align with those created by human evaluators using the SumMe dataset	. F-scores offer a quantitative measure to assess system performance compared to human summaries, providing an objective evaluation based on human judgments.	F-scores in summarization evaluations may not capture all quality aspects, like coherency and storytelling. Human-generated summaries are subjective and can vary among annotators, leading to variable evaluation results.

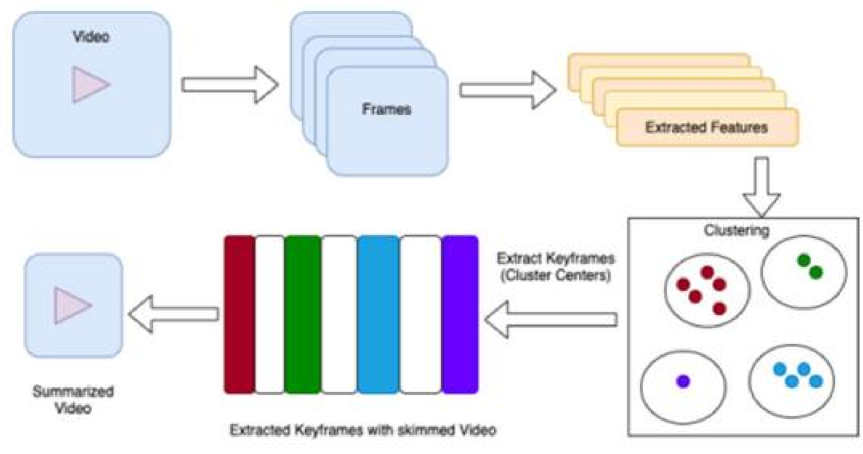
Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable

Relationship Among The Above 4 Variables in This article

Input and Output	Feature of This Solution	Contribution & The Value of This Work
<table><tr><th>Input</th><th>Output</th></tr><tr><td>Summe dataset from “Creating summaries from user videos,”</td><td>Video Summary</td></tr></table>	This solution takes videos and chooses the most important moments. It then groups these moments into interesting and less interesting parts. Finally, it makes a shorter, easier-towatch summary video. It's like picking the best parts of a movie. The choice of techniques matters, as some are faster, while others take more time.	This solution offers a structured method for creating concise video summaries, accommodating various video types. It enhances user-friendliness by applying adaptable keyframe extraction and clustering techniques. The approach ensures efficient content presentation, improving video accessibility. However, the selection of techniques must match video characteristics for optimal results.

<table>
<tr><th colspan="2">Positive Impact of this Solution in This Project Domain</th><th>Negative Impact of this Solution in This Project Domain</th></tr>
<tr><td colspan="2">This solution enhances video consumption by condensing lengthy content into succinct summaries. It benefits content creators, allowing them to engage viewers more effectively. For consumers, it saves time and provides quick access to essential information. Moreover, its adaptability across different video types is a significant advantage.</td><td>One challenge lies in the selection of appropriate techniques, which can be time-consuming and computationally intensive. The risk of losing critical content in the summarization process is another concern. Additionally, evaluation using F-scores may not fully capture qualitative aspects like coherency and storytelling. The subjectivity in human-generated summaries can introduce variability in the evaluation results, making it challenging to achieve a universally accepted standard for video summarization.</td></tr>
<tr><th>Analyse This Work By Critical Thinking</th><th>The Tools That Assessed this Work</th><th>What is the Structure of this Paper</th></tr>
<tr><td>The solution this work provides valuable insights into the complex domain of video summarization. It acknowledges the multifaceted challenges and explores a range of techniques, thereby contributing to the field of video content management. The critical thinking here involves recognizing the need for adaptability in summarization techniques and the importance of aligning algorithms with the specific characteristics of the video content. It also highlights the importance of bridging the gap between automated summarization and human perception, a crucial consideration in this context.</td><td>Machine learning
F-Score
SumMe Dataset
Clustering Algorithms</td><td>Title
I. Abstract
II. Introduction
III. Related Research
IV. Approaches
V. Experiments and Results
VI. Conclusion
VII. References</td></tr>
<tr><th colspan="3">Diagram/Flowchart</th></tr>
</table>



---End of Paper 19---

20		
Reference in APA format	Zawbaa, H.M., El-Bendary, N., Hassanien, A.E., Kim, Th. (2011). Machine Learning-Based Soccer Video Summarization System. In: Kim, Th., et al. Multimedia, Computer Graphics and Broadcasting. MulGraB 2011. Communications in Computer and Information Science, vol 263. Springer, Berlin, Heidelberg.	
URL of the Reference	**Authors Names and Emails**	**Keywords in this Reference**
https://doi.org/10.1007/978-3-642-27186-1_3	Hossam M. Zawbaa, Nashwa El-Bendary, Aboul Ella Hassanien, and Tai-hoon Kim	Support Vector Machine Detection Phase Video Shot Video Summarization Sport Video
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	**The Goal (Objective) of this Solution & What is the problem that need to be solved**	**What are the components of it?**
Machine Learning-Based Soccer Video Summarization System	The aim of the proposed solution is to develop a machine learning-based system for the automatic summarization of soccer match videos. The primary goal is to extract and highlight key events, such as	The authors develop a soccer video summarization system. This system automatically extracts and highlights pivotal events, such as goals and attacks, using SVM and NN for classification. It also includes score board detection and utilizes K-means clustering, Hough transform,

	goals, attacks, and other exciting moments, from the videos to create concise and engaging summaries. This solution seeks to enhance the viewing experience for soccer fans by providing them with a time-efficient way to relive the most important moments of the game	Gabor filters, and audio analysis for excitement event detection. This comprehensive approach condenses soccer matches into engaging summaries, enhancing the viewer experience.

The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process

The proposed system is a sophisticated soccer video summarization solution. It leverages machine learning techniques, including Support Vector Machine (SVM) and Neural Network (NN), to automatically detect and highlight key events in soccer matches. These events encompass goals, attacks, and various other exciting moments. Utilizing image processing and audio analysis, the system identifies and extracts relevant information, such as logos and score boards. It combines multiple algorithms, including K-means clustering, Hough transform, and Gabor filters, to ensure event detection accuracy. The result is a concise and engaging summary that enhances the viewing experience for soccer fans. Helps identify distinct parts of the video. Reduces redundancy and retains important visual content. The choice of segmentation criteria may affect the quality of the summary. Keyframe selection criteria can impac

	Process Steps	Advantage	Disadvantage (Limitation)
1	Video Processing Phase: -Segment the video into smaller shots based on dominant colors and Classify video shots into different types and identify play and break sequences. It also has replay detection.	Helps identify distinct parts of the video. Reduces redundancy and retains important visual content.	The choice of segmentation criteria may affect the quality of the summary. Keyframe selection criteria can impact the quality of the summary.
2	Event detection: Use SVM to locate and extract score board information and Identify exciting moments near the goal-mouth area using	Effective score record and enhances the viewer's experience by detecting goals and penalties	Sensitive to noise complex, algorithm may be needed, reliance on specific visual cues

	techniques like K-means clustering, Hough transform, and Gabor filters.		
3	Other Event detection and Summarization Phase: The proposed system highlights the most important events during the soccer match, such as goals and goal attempts, in order to save the viewer's time and introduce the technology of computer-based summarization into sports field.	Provides a concise representation of the video's content.	The choice of the summary format (clip or poster) may not suit all user preferences.
4	Evaluation & Tuning involves assessing the summary using recall and precision metrics, comparing it to reference summaries. Parameter tuning is essential to optimize the summarization process for diverse video types, ensuring it functions effectively.	Quantifies the quality of the summary. Allows for adaptability to varying video content.	Evaluation metrics may not capture all aspects of summary quality. Requires expertise and time to determine optimal parameter settings.

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable

Relationship Among The Above 4 Variables in This article

Input and Output	Feature of This Solution	Contribution & The Value of This Work

<table>
<tr>
<td>
<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>Sports match video</td><td>Video Summary</td></tr>
</table>
</td>
<td>Developing diversity-representative functions that can help in generate summaries include wide range of content while still capturing key moments. And developing supervision signals help in selecting relevant frames.</td>
<td>Good to have this knowledge from this paper as we reviewing of ideologies for developing cricket sports summarization using neural networks.</td>
</tr>
</table>

<table>
<tr><th>Positive Impact of this Solution in This Project Domain</th><th>Negative Impact of this Solution in This Project Domain</th></tr>
<tr><td>Considers the user intrests and has enhanced event detection accuracy. Generates engaging highlights and improves accessibility</td><td>Sensitive to content and broadcast variations, inaccuracies may occur in non-stranded broadcasts</td></tr>
</table>

<table>
<tr><th>Analyse This Work By Critical Thinking</th><th>The Tools That Assessed this Work</th><th>What is the Structure of this Paper</th></tr>
<tr><td>The solution offers advanced soccer video summarization but faces complexities in event detection due to its reliance on visual cues. Striking a balance between precision and recall is crucial. Scalability and user experience need further evaluation, and human verification may be necessary in complex scenarios to ensure accuracy and viewer satisfaction.</td><td>Machine learning</td><td>Abstract
I. 1. Introduction
II. 2. Machine Learning (ML): A Brief
Background
III. 3. The Proposed Soccer Video
Summarization System
- Pre-processing Phase
- Shot Processing Phase
- Replay Detection Phase
- Score Board Detection Phase
- Excitement Event Detection Phase
- Event Detection and Summarization
Phase
IV. 4. Experimental Results
V. 5. Conclusions and Future Works</td></tr>
<tr><th colspan="3">Diagram/Flowchart</th></tr>
<tr><td colspan="3"></td></tr>
</table>

---End of Paper 20---

2.2 COMPARISON TABLE:

Author	Year	Approach	Description
R. Agyeman, R. Muhammad and G. S. Choi	2019	Deep learning with ResNet-based 3D CNN and LSTM for soccer video summarization	The paper proposes a method combining CNN and LSTM to summarize soccer videos efficiently
C. Lin and Y. Chen	2019	Utilize deep learning with 3D convolution neural networks for MLB video summarization.	Employing deep learning to intelligently summarize Major League Baseball videos for enhanced content retention.
Y. Takahashi, N. Nitta and N. Babaguchi	2005	Utilize metadata and normalized play scene significance for sports video summarization.	Combining the temporal compression and spatial image keyframes to summarize sports videos effectively using metadata.
M. Z. Khan, S. Jabeen, S. ul Hassan, M. A. Hassan, and M. U. G. Khan	2019	Utilizes motion based scene boundary detection, CNN for frame importance, and Bidirectional LSTM for redundancy removal.	Proposes a method to enhance multimedia content understanding, providing an efficient video summary by removing redundant frames.
M. B. Andra and T. Usagawa	2019	Combine linguistic segmentation and attention-based RNN for lecture video summarization.	Employing RNN and linguistic segmentation to enhance lecture video summarization.
Solayman Hossain Emon	2020	Encoder-Decoder architecture employing CNN, LSTM, and reward functions to optimize the video summary identifying key moments.	Proposes a method to optimize the video summarization process by reward functions and supervision signals.
Hansa Shingrakhia, Hetal Patel.	2020	Employs speech-to-text frameworks to detect excitement.	Utilizes many forms of data from video for video summarization.
Besta Srikanth	2023	Employs CNN for visual feature extraction, LSTM for temporal event understanding.	Proposes a method for event highlight ranking.

Chakradhar Guntuboina	2021	Employs YOLO for scoreboard detection, OCR for digit recognition and extraction.	Proposes a method for detecting the score from scoreboard.
Aniqa Dilawari, Muhammad Ghani Khan	2019	Employs CNN for visual feature extraction, sequence-to-sequence for model, Bi-Directional LSTM for text generation.	Proposes a method to generate abstract text summaries for videos using CNN and RNN.
Abhishek Yadav	2020	Utilizes InceptionV3 CNN model to extract visual feature and LSTM for text generation.	Proposed a model for captioning key frames in a video using neural networks.
Joys Princia A	2021	Visually Guided Document Attention Network (VDAN), CNN and LSTM.	Proposed a model for both video and text summariza.
Hansaraj Wankhede	2023	Uses Ffmpeg to extract audio and visual features, Assembly AI for summarization and NLP library to fine tune the summary.	Proposed a model to summarize videos using Assembly AI and NLP.
Jonghwan Mun	2019	3D CNN model, GRU and RNN.	Proposed a model for captioning a video using Deep Learning and Reinforcement learning to improve accuracy.
V.Vijayakumar	2011	Gray image conversion and canny edge detection.	The OCR tool extracts the text information from sports videos.
Jingxu Lin , Sheng-hua Zhong , Ahmed Fares	2021	a 3D CNN (3D ResNeXt-101), hierarchical LSTM network	The proposed system uses a novel supervised approach that combines a 3D CNN and hierarchical LSTM network with attention to capture long-term dependencies addressing the imbalanced class distribution
Mayu Otani , Yuta Nakashima , Esa Rahtu , Janne Heikkil , and Naokazu Yokoya	2016	Video Summary, Deep Feature, Common Semantic Space, Summer Dataset, Joint Training	The paper presents an innovative video summarization approach using deep semantic features. It leverages a pre-trained 3D ResNeXt-101 model and hierarchical LSTM modules for accurate and efficient summarization.

Maria Nektaria Minaidi, Charilaos Papaioannou, Alexandros Potamianos	2023	Feature extraction, Machine learning, Benchmark testing, Neural networks	Enhance video summarization using advanced GAN-based architecture and attention mechanisms.
Shruti Jadon; Mahmood Jasim	2020	Video Summarization, Vision, Deep Learning, Clusterring, Image processing	Algorithm enhances video summarization, aligning with human preferences efficiently.
Hossam M. Zawbaa, Nashwa El-Bendary, Aboul Ella Hassanien, and Tai-hoon Kim	2011	Support Vector Machine Detection Phase Video Shot Video Summarization Sport Video	Machine learning system condenses soccer matches into engaging summaries efficiently.

2.3 WORK EVALUATION TABLE:

Author Name and Year	Work Goal	System's Components	System's Mechanism	Features /Characteristics	Performance	Advantages	Results
Rockson Agyeman, Rafiq Muhammad, Gyu Sang Choi 2019	Develop a soccer video summarization system using 3D-CNN and LSTM for action recognition and concatenation.	ResNet-based 3D CNN and LSTM work together to extract features and model temporal evolution for soccer highlight detection.	The system processes soccer videos, extracting spatiotemporal features with 3D-CNN and using LSTM to identify and concatenate highlights.	Features include ResNet-based 3D CNN, manual annotation of 744 clips, and a summarization approach	The proposed system achieves high accuracy in action recognition, outperforming existing benchmarks on UCF101.	effective soccer action recognition, flexibility for diverse sports application with minimal modification, and a heuristic-based summarization approach.	Summarized soccer videos receive a collective 4 out of 5 Mean Opinion Score (MOS) in evaluations, indicating good overall performance.
ChingShun Lin, YuChing Chen 2019	Use deep learning to summarize sports videos by addressing the semantic gap between low-level features and human perception.	3D-CNN and 2D Convolutional LSTM employed for video summarization.	Utilizes spatiotemporal features to extract keyframes, bridging the semantic gap.	Integrates 3D-CNN and 2D Convolutional LSTM for effective video summarization.	Evaluated using recall rate, precision rate, and F1-score, demonstrating efficient summarization.	Provides an effective, efficient, and robust solution for MLB video summarization.	Achieves a high recall rate, precision rate, and F1-score in summarizing baseball events, proving effectiveness.

Yoshimasa Takahashi, Naoko Nitta, Noboru Babaguchi 2005	Automatic content-based video summarization for large sports video archives.	Play scene significance, summarization methods, metadata usage, play scene selection, visualization techniques, evaluation metrics, experimental results.	Play scenes are ranked based on significance using metadata, and summaries are generated by selecting scenes according to user-specified length.	Semantic content-based summarization, flexibility in summary length, two visualization systems (video clip and video poster), hierarchical access to scenes.	Experimental results show 66% recall and 83% precision compared to TV broadcasted summaries.	Efficiently condenses significant play scenes, handles large sports video archives, adaptable to different video types with parameter adjustments.	Successful generation of video summaries with high recall and precision rates.
Muhammad Zeeshan Khan, Saira Jabeen, Saleet ul Hassan, M.A Hassan, Muhammad Usman Ghani Khan 2019	Develop a video summarization technique using CNN and Bidirectional LSTM for efficient multimedia content overview.	Scene Boundary Detection, Convolutional Neural Network (CNN), Bidirectional Long Short-Term Memory (LSTM).	Detects scene boundaries, assigns frame importance with CNN, removes redundancy with Bidirectional LSTM, and generates a concise video summary.	Utilizes motion features for scene detection, CNN for frame importance, and Bidirectional LSTM for redundancy elimination.	Outperforms traditional feature-based approaches, demonstrated by a higher F-measure score on the TVSUM50 Dataset.	Efficiently captures significant video content, providing a more accurate and compact summary.	Achieves a superior F-measure score of 0.84% compared to other state-of-the-art methods on the TVSUM50 Dataset

Muhammad Bagus Andra, Tsuyoshi Usagawa 2019	Develop an automatic lecture video summarization system using attention-based RNN for improved content accessibility.	Preprocessing module, Transcript Segmentation (PowerSeg method), and Attention-Based Recurrent Neural Network (seq2seq with attention).	Preprocesses transcripts, segments based on linguistic features, and employs an attention-based RNN for summarization.	Utilizes noise removal, linguistic feature extraction, and attention mechanism for effective lecture summarization.	The proposed system achieves high accuracy in action recognition, outperforming existing benchmarks on UCF101.	effective soccer action recognition, flexibility for diverse sports application with minimal modification, and a heuristic-based summarization approach.	Significant improvements in ROUGE scores, demonstrating the proposed model's effectiveness in capturing key content compared to baseline methods.
Solayman Hossain Emon, A.H.M Annur, Abir Hossain Xian 2020	Developed a summarization network that can automatically extract and select the most important moments from cricket match video for creating concise summaries that capture the key moments.	Deep Cricket Summarization Network, CricSum, Convolution Neural Netwok, Bidirectional Long short-term memory.	DCSN is an encoder-decoder architecture that predicts frame-level probabilities for video summarization. It uses CNN, LSTM to reward functions to optimize the summary identifying key moments	Diversity - representative functions that can help in generate summaries include wide range of content while still capturing key moments.	The performance was evaluated by F1-score for ground truth summary and mean opinion score to measure the degree of human judgement.	The system provides improve time and resource efficiency.	The models perform the task with 60.6% accuracy.

			.				
Hansa Shingrakhia, Hetal Patel **2021**	Developed a summarizing cricket video. Lengthy duration, content complexity, Identification of key events, Heterogeneous Data Sources.	Speech to Text Framework, Key frame extraction, HRF-DBN Classifier, Scorecard Region Detection, Action Recognition model, SGRNN-AM, Temporal Feature Extraction.	Detects key moments through audio excitement analysis. Classifying shots using hue histogram differences. OCR for scoreboard analysis, Umpire gesture recognition using joint based features.	Audio analysis, OCR for scoreboard analysis, Umpire Gesture Detection, shot boundary detection.	Performance was evaluated by f1-score accuracy, error rate.	Utilizes multi-modal data and textual data efficiently	The model performs the task with 96.32% accuracy.

Besta Srikanth, Mopuri Veera Narayana, Narayana Satya, Sagarla Aravind 2023	Converting Soccer video to text summmarization by selecting keyframes.	Manual Annotaion, Feature Extarction, LSTM Network	CNN to extract visual features and LSTM to generate text summarization using the visual features. And Ranking mechanism to rank highlights.	The model can generate text summaries from soccer videos.	The performance is evaluated using Mean Score.	Can be derivable to other sport domains as well.	By human assessment the accuracy for MOS received 80%(4/5) rating.
Chakradhar Guntunboina, Aditya Porwal, Preet Jain, Hansa Shigrakhia 2021	The identification of scores is accomplished by using YOLO and customised CNN.	YOLO, Image Processing techniques, OCR, F1-score	YOLO detects the scoreboard region and using OCR textual info of score is extracted.	The model can generate scores with timestamps.	The performance is evaluated using F1-Score.	Can be applicable to multiple sports.	The models perform the task with 98% accuracy.
Aniqua Dilawari, Muhammad Usman Ghani Khan 2019	To understand the semantics embedded within the videos and convert visual information into	CNN, LSTM encoder-decoder, Attention mechanisms.	This model uses CNN to extract visual features and Seq2Seq model for machine translation and	An effective approach for generating abstract text summary from videos using deep neural networks	The performance is evaluated by Human assessment, meteor, rogue.	Generates text summarization of videos effectively.	The rating by human assessment achieved is 3.6.

	text information.		LSTM for multiline text generation. And pointer generation network	CNN, LSTM along with attention mechanism and pointer-generation network.			
Abhishek Yadav, Anjali Vishwakarma, Shyama Panickar, Satish Kuchiwale	The work goal is to develop a neural network-based system for automatically summarizing live video content into concise text captions for enhanced understanding and navigation.	The system comprises components such as feature extraction using CNNs, caption generation viaRNNs (LSTM), data preparation, model training, evaluation metrics, system design visualization, and result analysis.	The system mechanism includes CNN-based image feature extraction, LSTM-driven caption generation, data preparation, model training, BLEU score evaluation, and result presentation through system design and analysis.	The system's characteristics include real-time summarization, neural network integration for image features and caption generation, semantic understanding, data processing, performance evaluation using BLEU scores, and comprehensive visualization and analysis.	Performance is assessed by evaluating caption quality using BLEU scores, indicating the system's accuracy in summarizing video content through generated captions.	The system's advantage is its ability to automatically condense live video into concise text summaries, facilitating efficient navigation and comprehension of extensive video datasets.	The results highlight the system's efficacy in producing accurate text summaries from live video frames, validated through evaluations and detailed analysis.

Joys Princia A, Ms. J Sangeetha Priya, Kalai Selvi J, Rithi Afra J, Rukshana S	The primary goal is to develop efficient algorithms using deep learning to summarize videos and text, condensing content while retaining essential information for quick comprehension.	The system comprises components such as VDAN for feature extraction, CNN for visual data, LSTM for sequential understanding,and RNN for text summarization, aiming to condense video and text while retaining key information.	The system integrates VDAN, CNN, LSTM, and RNN to extract features, comprehend sequences, and summarize content from videos and text, while preservingkey elements.	Uses VDAN for extraction textual features.	The system's performance is evaluated through precision, recall, and F1 score metrics in video summarization, ensuring accuracy , while text summarization focuses on coherence in generated summaries.	The system's advantages lie in its ability to efficiently concentrate on video and text content, facilitating quick comprehension, time-saving, and accessibility to essential information amongst lengthy data.	The system demonstrates successful video summarization, achieving high precision, recall, and F1 scores, while text summarization yields coherent and concise summaries, showcasing the effectiveness of the implemented techniques.
Hansaraj Wankhede, R Bharathi Kumar, Sushant Kawade, Ashish Ramtekkar, Rachana Chawke	The research aims to develop an AI-powered Video summarization system using FFmpeg, NLP, and Assembl	The system comprises three primary components: an Ffmpeg-based video Processing module, an NLP-based	The system combines FFmpeg for video processing, AssemblyAI for automated transcription, NLP	Instead of LSTM it uses NLP techniques for summarization.	The AI model Consistently achieves high accuracy and efficiency in summarizing diverse content, with continuous enhancements in precision and speed through iterative learning	-	The project aims to generate concise video summaries using Ffmpeg and NLP, evaluating accurac

	yAI to Automatically extract key information from videos.	Information extraction module, and an AssemblyAI-powered speech-to-text module for audio analysis.	techniques, and a Multi-Source Visual Attention model to generate concise and accurate video summaries through a multi-step mechanism.		processes.		y, efficiency, and user satisfaction for various applications.
Jonghwan Mun, Linjie Yang, Zhou Ren, Ning Xu, Bohyung Han	The goal is to develop a framework for dense video captioning that captures temporal dependencies between events, ensuring coherent and context-aware Caption Generation for improved video understanding	The system consists of an Event Proposal Network (EPN), an Event Sequence Generation Network (ESGN), and a Sequential Captioning Network (SCN) With Reinforcement	The system initially selects event proposals via EPN, uses ESGN to detect adaptive event sequences, and then employs SCN, guided by reinforcement learning, to generate coherent captions by leveragi	The system uses Reinforcement Learning to improve the accuracy of the summary.	The system showcases leading performance on the ActivityNet Captions dataset by effectively capturing temporal dependences and context, elevating accuracy in dense video captioning metrics like METEOR, CIDEr, and BLEU.	Generating efficient summaries by leveraging Reinforcement Learning.	The results display the system's superiority in dense video captioning metrics, highlighting its ability to capture temporal dependencies and generate context-aware captions, surpassing existing methods on the Activity Net

	.	learning for dense video captioning incorporating temporal dependencies and context awareness.	ng temporal dependencies and context in dense video captioning.				Captions dataset.
V.Vijayakumar, R.Nedunchezhian	The goal of the proposed solution is to develop a method for effectively extracting superimposedtext from sports videos. The problem it addresses isthe need to automatically detect, isolate, and extract textual	The system Comprises video frame extraction, key frame identification, grayscale conversion, image cropping, Canny edge detection, text region retrieval, and OCR for text transformation, Enabling Systematic extraction of textual data from	The mechanism involves frame extraction, key frame selection, grayscale conversion, region isolation, Canny edge detection, and OCR for systematic extraction of textual data from sports	In this approach only the areas where text is present are targeted and the unnecessary information is removed.	The system's Performance in text extraction from sports videos is calculated using metrics like Recall -Precision andAccuracy, Showcasing Effectiveness with room for further optimization	As it focuses only on useful information the computation alcost is reduced.	The results the system exhibit promising accuracy in extracting text from sports videos, as demonstrated by metrics like Recall-Precision and Accuracy, its potential for effective indexing and

	information.	sports videos.	videos				retrieval purposes with room for further refinement.
Jingxu Lin , Sheng-hua Zhong , Ahmed Fares	The goal of DHAVS (Deep Hierarchical LSTM Networks with Attention for Video Summarization) is to provide a framework for compressing videos effectively using a multi-faceted strategy for video summarization.	DHAVS uses a pre-trained 3D ResNeXt-101 model for spatio-temporal feature extraction, an attention-based hierarchical LSTM module for capturing semantic information and temporal dependencies, and a cost-sensitive loss function for addressing	DHAVS uses a multi-stage approach involving scene change detection through KTS, shot-level scoring, and a dynamic programming-based solution to the 0-1 Knapsack problem. It captures spatio-temporal features through a pre-trained 3D ResNeXt-101	DHAVS uses a multi-stage approach involving scene change detection through KTS, shot-level scoring, and a dynamic programming-based solution to the 0-1 Knapsack problem. It captures spatio-temporal features through a pre-trained 3D ResNeXt-101 model, and enhances	DHAVS is evaluated using F-score and correlation coefficients, and it is shown to outperform existing methods in summarizing videos.	DHAVS offers a comprehensive solution to video summarization tasks through a combination of features and mechanisms that enhance its performance.	The DHAVS system achieves competitive results in video summarization tasks, outperforming existing methods that use deep learning and machine learning techniques.

		imbalanced class distribution. It also leverages Kernel Temporal Segmentation (KTS) for scene change detection and a dynamic programming-based solution for summarization.	model, and enhances semantic understanding and temporal dependencies using an attention-based hierarchical LSTM module.	semantic understanding and temporal dependencies using an attention-based hierarchical LSTM module.			
Mayu Otani , Yuta Nakashima , Esa Rahtu , Janne Heikkil , and Naokazu Yokoya	The paper aims to improve video summarization using deep semantic features from videos for better, more meaningful summaries. It addresses condensing lengthy	The approach has a multi-faceted strategy for effective video summarization. Leveraging the power of a pre-trained 3D ResNeXt-101 model, it captures spatio-temporal features. The	The proposed video summarization system DHAVS introduces a hierarchical attention network based on deep semantic features for summarization of raw videos, combined with temporal	Multi-faceted approach for video summarization Leverages pretrained 3D ResNeXt-101 model to capture spatio-temporal features. Attention-based hierarchical LSTM module enhances semantic	The paper reports performance results on two datasets, SumMe and TVSum. On the SumMe dataset, the proposed system outperformed the state-of-the-art methods in terms of F1 scores. On the TVSum dataset, the proposed system achieved comparative or better performance	The proposed system is multi-faceted, accurate, and computationally efficient in generating video summaries. The cost-sensitive loss function incorporated in the system enhances its performance in dealing with imbalanced	The proposed system achieved state-of-the-art performance in terms of F1 scores and comparative or better performance on the SumMe and TVSum datasets, respecti

	videos into concise yet informative summaries, crucial for tasks like browsing collections or efficient retrieval. Manual summarization is time-consuming and subjective, prompting the need for automatic identification of relevant video segments, a complex challenge this solution tackles.	introduction of an attention-based hierarchical LSTM module enhances semantic understanding and temporal dependencies. To combat imbalanced class distribution, a cost-sensitive loss function is employed. The summarization process involves scene change detection through Kernel Temporal Segmentation (KTS), shot-level scoring, and a dynamic programming-based	techniques and a novel cost-sensitive loss function.	understanding and temporal dependencies. Summarization process involves scene change detection through Kernel Temporal Segmentation (KTS), shot-level scoring, and a dynamic programming g-based solution to the 0–1 Knapsack problem. Ensures both accuracy and computational efficiency in generating video summaries	compared to state-of-the-art methods.	class distribution, ensuring better results.	vely. Specifically, for the SumMe dataset, the proposed system attained an F1 score of 0.512, which was better than the performance of all state-of-the-art unsupervised video summarization methods. For the TVSum dataset, the proposed system scored 0.378 and 0.500 on two of the videos, respectively, outperforming the state-of-the-art

		solution to the 0–1 Knapsack problem, ensuring both accuracy and computational efficiency in generating video summaries.					methods on these videos.
Maria Nektaria Minaidi, Charilaos Papaioannou, Alexandros Potamianos	The goal of this paper is to improve unsupervised video summarization through advanced GAN-based architectures, addressing the challenge of condensing extensive video content. By integrating attention mechanisms and	The proposed solution utilizes a Generative Adversarial Network (GAN) comprising attention mechanisms, LSTM units, and a Variational Autoencoder (VAE). This framework employs self-attention, transformers,	The proposed approach leverages attention mechanisms and transformers to capture long-term temporal dependencies, while combining LSTM and transformer models to encode, decode, and select frames for	Advanced GAN-based architectures for video summarization. Attention mechanisms, LSTM units, and a Variational Autoencoder (VAE) employed for video encoding and decoding. Self-attention and transformers used to capture long-term	The proposed system achieved state-of-the-art performance in terms of F1 scores and comparative or better performance on the SumMe and TVSum datasets, respectively. Specifically, for the SumMe dataset, the proposed system attained an F1 score of 0.538, which was better than the performance of all state-of-the-art methods. For	The proposed system is advanced and effective in generating accurate, concise video summaries, outperforming existing state-of-the-art approaches. It leverages advanced GAN-based architectures, attention mechanisms, and transformers to capture and represent video content accurately.	The proposed self-attention based Generative Adversarial Network (SAGAN) achieved state-of-the-art performance in terms of F1 scores and comparative or better performance on the SumMe and

	transformers, the goal is to capture complex temporal dependencies and create accurate, concise video summaries for efficient content comprehension.	and LSTM modules for encoding, decoding, and capturing long-term temporal dependencies, enhancing unsupervised video summarization by creating concise summaries from extensive video content.	generating accurate and concise video summaries in an unsupervised manner. The Generative Adversarial Network enables joint training of summarizer and discriminator, enhancing summarization quality.	dependencies. Unsupervised video summarization for efficient content comprehension	the TVSum dataset, the proposed system scored 0.55 and 0.61 on two of the videos, respectively, outperforming the state-of-the-art methods on these videos.		TVSum datasets, respectively. Specifically, for the SumMe dataset, the proposed SAGAN system attained an F1 score of 0.538, which was better than the performance of all state-of-the-art methods. For the TVSum dataset, the proposed system scored 0.55 and 0.61 on two of the videos, respectively, outperforming the state-of-the-art methods on these

							videos. The proposed system was also comparable or better than existing approaches on the most extended video in the datasets.
Shruti Jadon; Mahmood Jasim	The goal of this work is to present and evaluate an unsupervised approach to summarize unstructured videos by combining techniques of keyframe extraction and video skimming.	The proposed system consists of three main components: 1) keyframe extraction using various image features such as color histograms and shot boundaries detection, 2) clustering of the keyframes using the k-means algorith	The system extracts keyframes using various image features and clustering methods, then applies video skimming to select critical keyframes and stitch them together into a summary. The proposed method uses boundar	- Unsupervised approach to summarize unstructured videos - Keyframe extraction using various image features such as color histograms and shot boundaries detection - Clustering of the keyframes using	The proposed system outperformed multiple unsupervised video summarization algorithms on the SumMe dataset, resulting in a higher F-score performance with shorter summary length.	The proposed framework is computationally efficient and able to summarize long videos (up to 2 hours) with high accuracy. The system uses boundary detection to enhance coherence and improve summarization quality.	The proposed system achieved an F1 score of 0.404 on the SumMe dataset and compared favorably on the TVSum dataset with state-of-the-art unsupervised video summarization techniques.

		m to identify significant frames 3) video skimming to create a summary from the selected keyframes with boundary detection and stitching keyframes based on content similarities.	y detection to maintain coherence between frames, which enhances the summarization quality of video sequences.	the k-means algorithm to identify significant frames - Video skimming to create a summary from the selected keyframes - Boundary detection and content-based stitching techniques to enhance summarization quality			
Hossam M. Zawbaa, Nashwa El-Bendary, Aboul Ella Hassanien, and Tai-hoon Kim	The goal of this work is to develop a machine learning-based soccer video summarization system that can automatically extract and	: The proposed soccer video summarization system consists of a pre-processing phase, shot processing phase, replay detection phase, score board	The proposed system uses K-means clustering, Hough transform, Gabor filters, and audio analysis to detect excitement events. It	The machine learning-based soccer video summarization system is designed to detect and highlight pivotal events such as goals and attacks. It utilizes	The proposed system uses K-means clustering, Hough transform, Gabor filters, and audio analysis to detect excitement events. It also supports vector machines (SVMs) and neural networks (NNs) for	The proposed system provides a comprehensive approach to summarizing soccer matches into engaging summaries, improving the viewer experience for soccer fans. The use of K-means	The proposed soccer video summarization system achieved an F-measure of 91.9% on the soccer video dataset. It uses a comprehensive

	highlight pivotal events, such as goals and attacks, from soccer matches, providing soccer fans with a time-efficient way to relive the most important moments of the game and enhancing their viewing experience.	detection phase, excitement event detection phase, and event detection and summarization phase. The system utilizes K-means clustering, Hough transform, Gabor filters, and audio analysis to detect excitement events, supports vector machines (SVMs) and neural networks (NNs) for classification, and includes score board detection.	also supports vector machines (SVMs) and neural networks (NNs) for classification. It includes score board detection and utilizes a comprehensive approach to detect and summarize the pivotal events in soccer matches.	various techniques including K-means clustering, Hough transform, Gabor filters, and audio analysis for excitement event detection. Additionally, the system employs support vector machines (SVMs) and neural networks (NNs) for classification purposes. One of its key features is scoreboard detection, which enhances its ability to provide comprehensive summaries of soccer matches.	classification. It includes score board detection and utilizes a comprehensive approach to detect and summarize the pivotal events in soccer matches.	clustering, Hough transform, Gabor filters, and audio analysis for excitement event detection, as well as SVMs and NNs for classification, enhances summarization accuracy	approach to detect and summarize pivotal soccer events, including excitement event detection and score board detection. The system improves the viewing experience for soccer fans.

CHAPTER 3

PROPOSED SYSTEM

3.1 PROPOSED SYSTEM

The proposed system aims to revolutionize cricket video summarization by leveraging advanced deep learning techniques and computer vision algorithms. It seeks to automate the process of extracting key insights and generating concise textual summaries from cricket match footage. By integrating cutting-edge technologies such as Convolutional neural networks (CNNs), optical character recognition (OCR), and recurrent neural networks (RNNs), the system endeavors to provide accurate and unbiased summaries of cricket matches, catering to the needs of coaches, players, researchers, and enthusiasts. Through seamless integration of various components, the system promises to offer actionable insights and comprehensive analyses, thereby enhancing the user experience and facilitating a deeper understanding of the game.

3.2 ADVANTAGES OF PROPOSED SYSTEM

The proposed system has the following advantages:

- Automation streamlines the summarization process, reducing time and effort.
- Leveraging advanced deep learning techniques ensures accuracy and reliability.
- Eliminates biases introduced by human interpretation for more objective summaries.
- Enables quick access to crucial match information through concise textual summaries.
- Facilitates informed decision-making and deeper analysis of gameplay dynamics.
- Enhances efficiency, accuracy, and accessibility in cricket match analysis.
- Improves the user experience and contributes to a better understanding of the game.

3.3 SYSTEM REQUIREMENTS

The system requirements for our project encompass both development and deployment aspects. These requirements are essential to ensuring smooth progress in building the application and its successful deployment to various platforms. Adequate computing resources and compatibility with target platforms are key considerations to enable efficient development and seamless functionality across devices. Additionally, reliable internet connectivity may be necessary for accessing external resources or cloud-based services during the development and deployment phases. Overall, careful attention to system requirements will facilitate the smooth execution and usability of our project.

3.3.1 SOFTWARE REQUIREMENTS

Below are the software requirements for application development:

1. Operating System: Compatible with Windows 10, macOS Mojave (10.14) or later, or popular Linux distributions such as Ubuntu 18.04 LTS or newer versions.
2. Python Environment: Python 3.x is installed with essential libraries such as TensorFlow, Keras, scikit-learn, NumPy, NLTK, and OpenCV for machine learning, natural language processing, and image processing tasks.
3. Development Tools: Integrated Development Environments (IDEs) such as PyCharm, Jupyter Notebook, or VSCode for coding, debugging, and experimentation.
4. Version Control: Git installed for version control management, facilitating collaboration and tracking changes in code and project files.
5. External Libraries and Models: Installation of additional libraries and models such as Paddle OCR, BART (Bidirectional and Auto-Regressive Transformers), and pre-trained models like VGG16 for image processing and text summarization tasks.

6. Internet Connectivity: High-speed internet connection for accessing online resources, downloading additional datasets, and cricket match footage.

3.3.2 HARDWARE REQUIREMENTS

Hardware requirements for application development are as follows:

1. CPU: A modern multi-core processor (Intel Core i5 or equivalent) to handle computational tasks efficiently.
2. GPU: A dedicated graphics processing unit (NVIDIA GeForce GTX 1060 or equivalent) with CUDA support for accelerated deep learning computations, especially for training large neural network models.
3. RAM: A minimum of 8GB of RAM (16GB recommended) to ensure smooth processing of large datasets and model training operations.
4. Storage: Adequate storage space (at least 500GB HDD or SSD) for storing video datasets, image frames, trained models, and intermediate data files.

3.3.3 IMPLEMENTATION TECHNOLOGIES

Python: The primary programming language used for implementing the cricket video-to-text summarization tool due to its extensive libraries and frameworks support for machine learning, deep learning, and natural language processing tasks.

OpenCV: Utilized for video processing tasks such as frame extraction, grayscale conversion, and object detection, enabling efficient handling of cricket match footage.

TensorFlow and Keras: Deep learning frameworks employed for building and training Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) models, facilitating tasks like object detection, image feature extraction, and text generation.

YOLO (You Only Look Once): Specifically, YOLOv8 model utilized for object detection, enabling accurate identification of cricket scoreboard regions within video frames, crucial for extracting textual details.

Paddle OCR: Leveraged for optical character recognition (OCR) capabilities, enabling the extraction of textual information from scoreboard images, player statistics, and other textual elements within cricket match footage.

NLTK (Natural Language Toolkit): Utilized for natural language processing tasks such as tokenization, enabling the parsing and analysis of textual data extracted from cricket videos for further processing and summarization.

BART (Bidirectional and Auto-Regressive Transformers): Specifically, the distilbart-cnn-12-6 model utilized for text summarization, enabling the generation of concise textual summaries encapsulating key highlights of cricket matches.

Git: Version control system employed for managing project codebase, facilitating collaboration, tracking changes, and ensuring code integrity throughout the development process.

IDEs (Integrated Development Environments): Development environments such as PyCharm, Jupyter Notebook, or VSCode used for coding, debugging, and experimentation, providing a conducive workspace for implementing and testing algorithms and models.

NVIDIA CUDA: Utilized for GPU acceleration, enhancing the performance of deep learning computations, especially during training phases, by leveraging the parallel processing capabilities of compatible NVIDIA GPUs.

CHAPTER 4

SYSTEM DESIGN

4.1 PROPOSED SYSTEM ARCHITECTURE

The proposed system involves the development of a web application that can be used to generate textual summaries of any cricket match. The application has been named CrikyWiki, and this refers to the application developed hereafter.

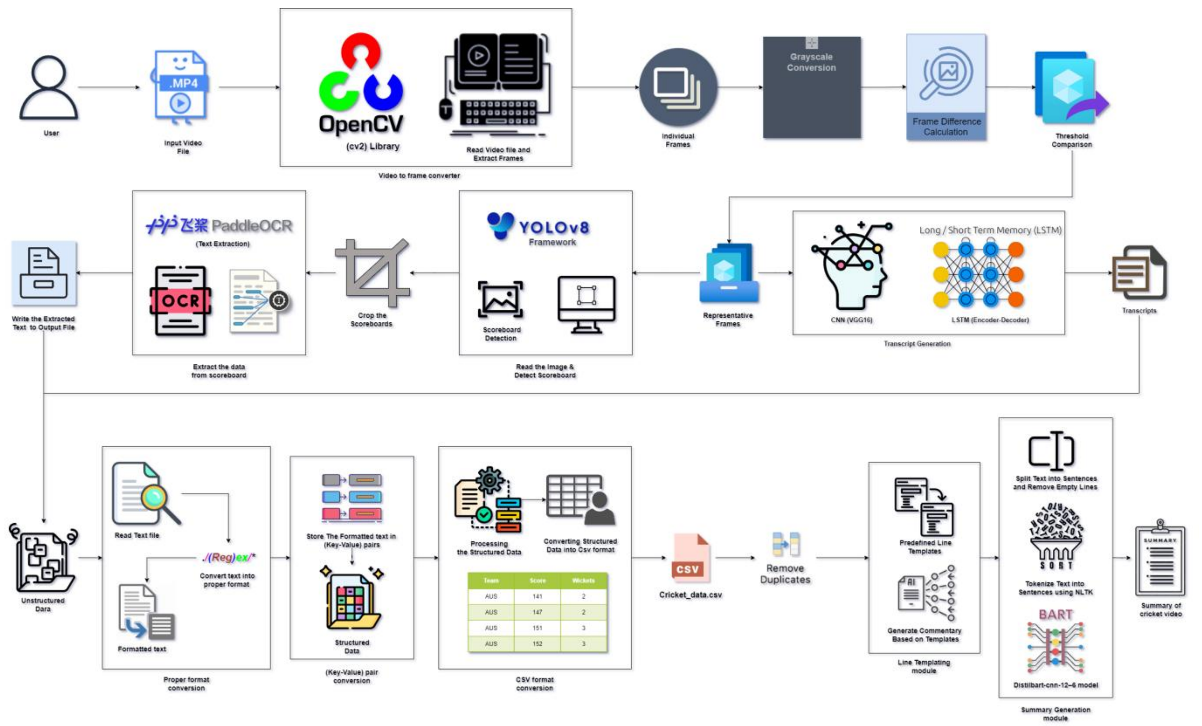


Figure 1: Proposed Architecture

4.2 APPLICATION MODULES

The application comprises five primary modules, each designed to fulfill a distinct function. Firstly, the Video Frame Conversion Module transforms video content into individual frames, enabling further analysis. Next, the Scoreboard Detection and Data Extraction Module identifies scoreboards within frames and extracts relevant data. The Data Structuring Module converts unstructured data into a structured format, facilitating efficient processing.

Subsequently, the Transcript Generation Module creates textual transcripts for each frame, aiding in content analysis. Lastly, the Line Templating and Summarization Module organizes the extracted data and transcripts into predefined line templates, culminating in a comprehensive summary of the video content.

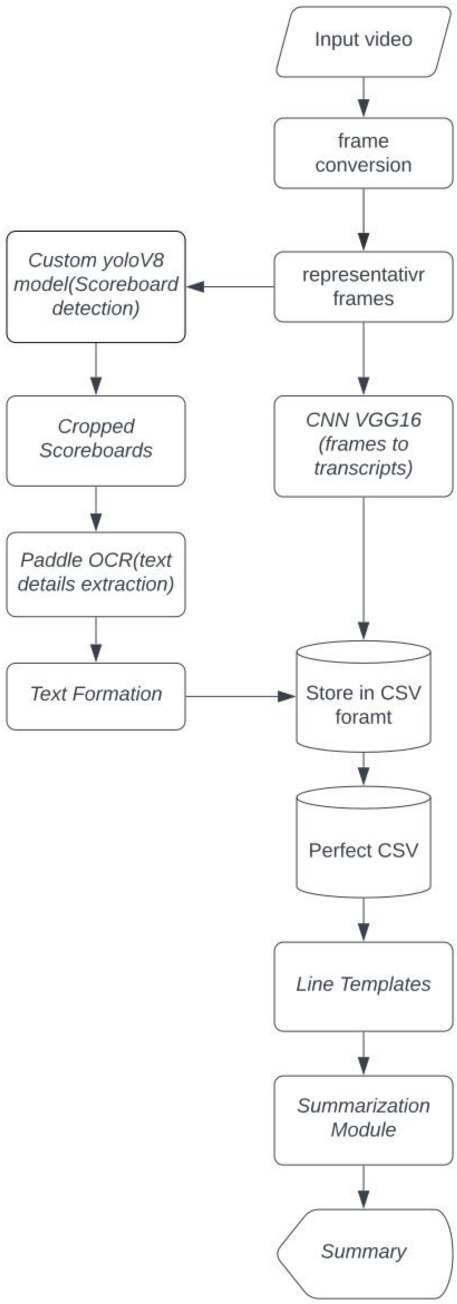


Figure 2: Workflow of the Proposed System

4.2.1 Video Frame Conversion Module:

The Video Frame Conversion Module is a crucial component of the application, facilitating the transformation of video content into a series of individual frames. This process is orchestrated through a systematic approach, beginning with the initialization of key parameters such as the input video path and the desired output directory. Upon setting these parameters, the module leverages the OpenCV (CV2) library to access and parse the input video file. Any failure to open the video file is meticulously handled, ensuring seamless execution. Subsequently, the module iterates through the video frames, reading each frame sequentially. As frames are read, they are saved to the designated output directory, with each frame meticulously labeled with a filename that reflects its position in the sequence. This systematic approach ensures the preservation of frame integrity and facilitates subsequent analysis. Simultaneously, the module maintains a vigilant watch for user input, enabling termination of the frame extraction process upon user command. Throughout the execution of this process, informative messages are provided to the user, offering clarity on the progression of frame extraction.

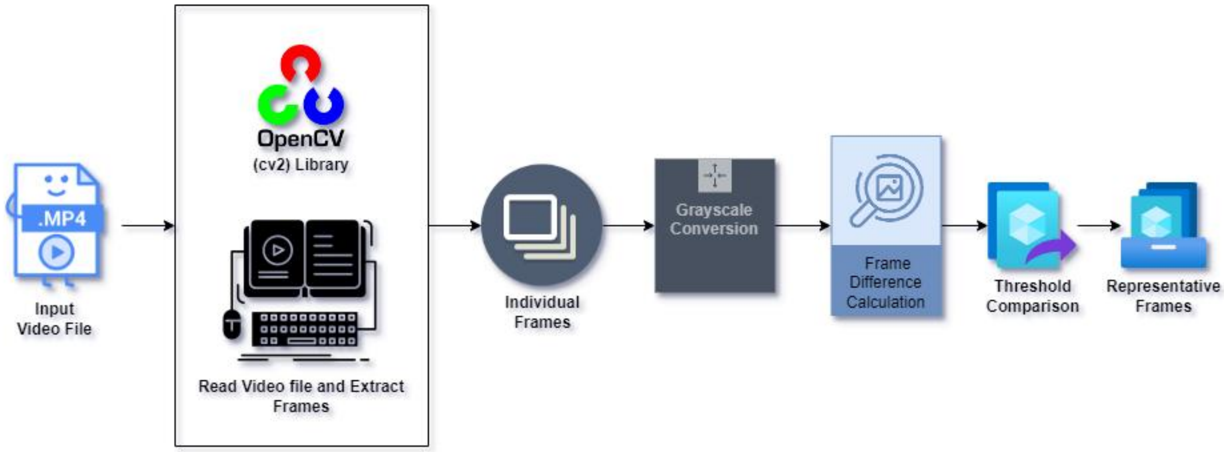


Figure 3: Workflow of the Architecture explaining the process of Video to Representative frames conversion.

Following the successful extraction of frames, the module meticulously identifies representative frames from the extracted set. This involves an additional process wherein the grayscale representation of each frame is evaluated to determine its uniqueness. By computing the absolute difference between consecutive frames and applying a configurable threshold, redundant frames are effectively filtered out. The remaining representative frames are then stored in a designated directory, ensuring efficient storage and accessibility for subsequent processing steps.

4.2.2 Scoreboard Detection and Data Extraction Module:

The Scoreboard Detection and Data Extraction Module plays a pivotal role in the application, focusing on two key processes: scoreboard detection using YOLOv8 and data extraction via Paddle OCR. Firstly, the module employs the YOLOv8 framework to accurately identify the cricket scoreboard region within the provided images. Leveraging a pre-trained model, the module swiftly analyzes each image within the designated directory, isolating the scoreboard through precise cropping. This process ensures that only the relevant region containing the scoreboard information is retained for subsequent analysis. The cropped scoreboard images are then stored in a dedicated directory, ready for further processing. Subsequently, the module utilizes Paddle OCR to extract textual information from the cropped scoreboard images. By configuring the OCR model with appropriate parameters, including language settings and GPU utilization preferences, the module ensures optimal performance. For each cropped image, the OCR engine diligently scans and interprets the textual content, extracting critical data such as player names, team scores, and individual player scores. The extracted information is then organized and compiled into a structured format, facilitating seamless integration with downstream processing modules.

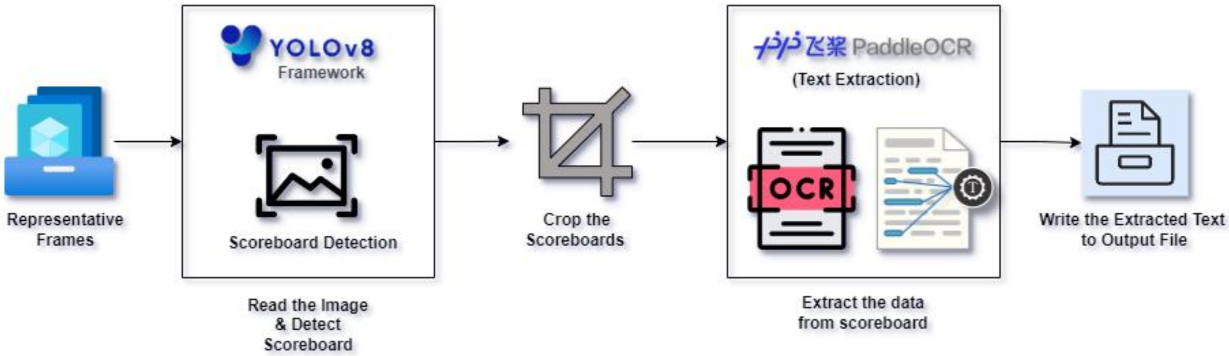


Figure 4: Workflow of the Architecture explaining the process of Scoreboard Detection and Data Extraction.

Throughout these processes, meticulous attention is paid to data integrity and accuracy. Stringent quality control measures are implemented to mitigate errors and ensure the fidelity of the extracted data. Additionally, robust error handling mechanisms are in place to address any unforeseen challenges encountered during execution, thereby enhancing the reliability and robustness of the module. The synergy between YOLOv8-based scoreboard detection and paddle OCR-based data extraction empowers the application to effectively glean insights from cricket match images. By automating the detection and extraction of scoreboard information, the module significantly streamlines the data analysis workflow, enabling users to derive actionable insights with minimal manual intervention.

4.2.3 Data Structuring Module:

The Data Structuring Module serves as a critical component within the application, focused on converting unstructured textual data into a structured format conducive to further analysis and processing. This module encompasses two key functions: Text Formation and Text to CSV File.

The Text Formation function operates by parsing textual information extracted from cricket match images, typically stored in a file named "Ocr_details.txt." Leveraging pre-defined regex patterns, this function systematically identifies key pieces of information such as frame numbers, team names, scores, player details, and bowling statistics. Each identified piece of information is then meticulously structured into a key-value pair format, ensuring consistency and accuracy across all extracted data points. The resulting structured data is subsequently written to an output file named "text_format.txt," facilitating easy access and reference for downstream processing tasks.

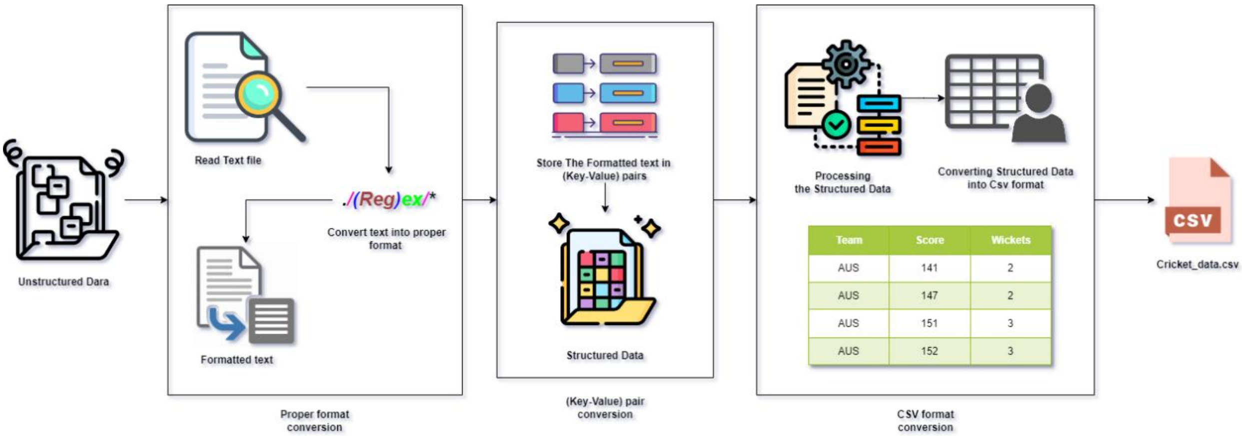


Figure 5: Workflow of the Architecture explaining the process of Data Structuring.

Subsequently, the Text to CSV File function further refines the structured textual data by converting it into a comma-separated value (CSV) format. By prompting the user to specify a file name, the module allows for customization of the output CSV file. Each line of the input text file is parsed, with key-value pairs separated by a colon-space delimiter. Special attention is given to specific categories of data, such as player details (e.g., striker, non-striker, bowler), which are converted into nested dictionaries for enhanced organization. Once all data within a frame has been processed, it is appended to a list representing the entirety of the cricket match data. Finally, this structured data is written to a CSV file named "cricket_data.csv," enabling seamless integration with external analysis tools and platforms. Through these processes, the Data Structuring Module effectively transforms raw textual data extracted from cricket match images into a structured format conducive to comprehensive analysis and interpretation. By leveraging regex patterns and systematic parsing techniques, this module ensures the integrity and reliability of the extracted data, empowering users to derive meaningful insights.

4.2.4 Transcript Generation Module:

The Transcript Generation Module is used for predicting actions within cricket video frames and generating descriptive transcripts. It begins by inputting frames from the video into a pre-trained CNN VGG16 model, which adeptly extracts visual features capturing various cricket elements such as batsman, bowler, wickets, fielder, ground, and audience, representing them as vectors. These vectors undergo further processing through an LSTM encoder-decoder architecture. The LSTM encoder handles the sequential input, capturing temporal relationships among the feature vectors. Subsequently, the decoder LSTM takes over, generating word integer tokens based on the encoded sequence. At each step, the decoder LSTM predicts the next token conditioned on prior tokens and the encoded sequence. These tokens denote different aspects of the cricket scene, including actions and contextual elements.

Following this, the word integer tokens are mapped back to their corresponding words using a vocabulary mapping, resulting in a sequence of words constituting a descriptive transcript. This transcript encapsulates the key elements and actions observed in the frame, providing a textual representation of the visual content. For instance, an input frame showing a batsman striking a cricket ball might yield a transcript such as "The batsman strikes the ball with power, aiming for a boundary." Through the fusion of CNN-based visual feature extraction and LSTM -based sequence generation, the Transcript Generation Module effectively bridges the visual and textual domains, facilitating a comprehensive understanding and interpretation of cricket video content.

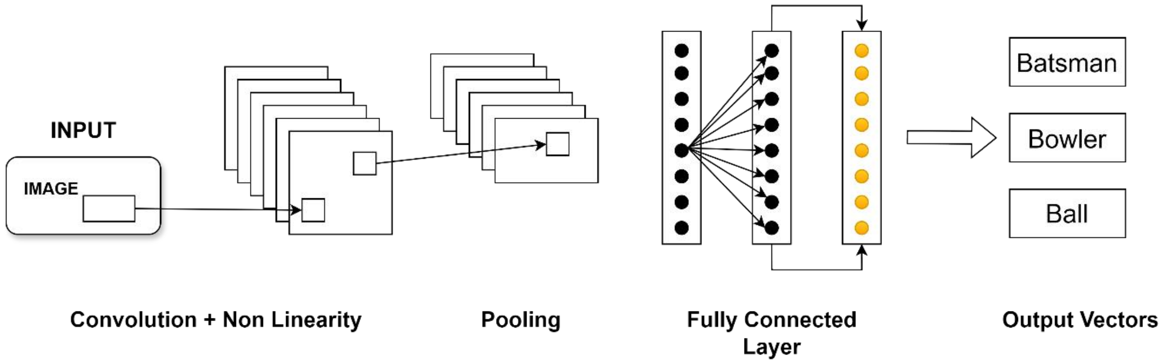


Figure 6: Workflow of the Vgg16 architecture explaining the process of Visual feature extraction

4.2.5 Line Templating and Summarization Module:

The Line Templating and Summarization Module is instrumental in generating coherent and concise textual summaries based on structured data extracted from cricket match analyses. This module encompasses three distinct functions: the perfect CSV function (data storing), line templates, and summarization. The perfect CSV function serves as a preparatory step, ensuring the integrity and efficiency of the data before proceeding to line Templating. It begins by reading data from "text_format.txt" and "transcripts.txt" and parsing and organizing it into a structured format. Notably, this function also addresses the potential issue of consecutive duplicate rows within the data, ensuring that each row represents unique and relevant information. Upon processing, the cleaned and refined data is stored in a CSV file named "Cricket_datanew.csv," ready for further analysis. Subsequently, the Line Templates function leverages the structured data stored in "Cricket_datanew.csv" to generate contextual commentary lines. These commentary lines are formulated based on predefined templates categorized into positive, negative, or neutral sentiments. Relevant fields from the input DataFrame, such as team names, scores, and wickets, are extracted and seamlessly integrated into the templates. The resulting commentary lines are then printed, providing insightful summaries of the cricket match dynamics.

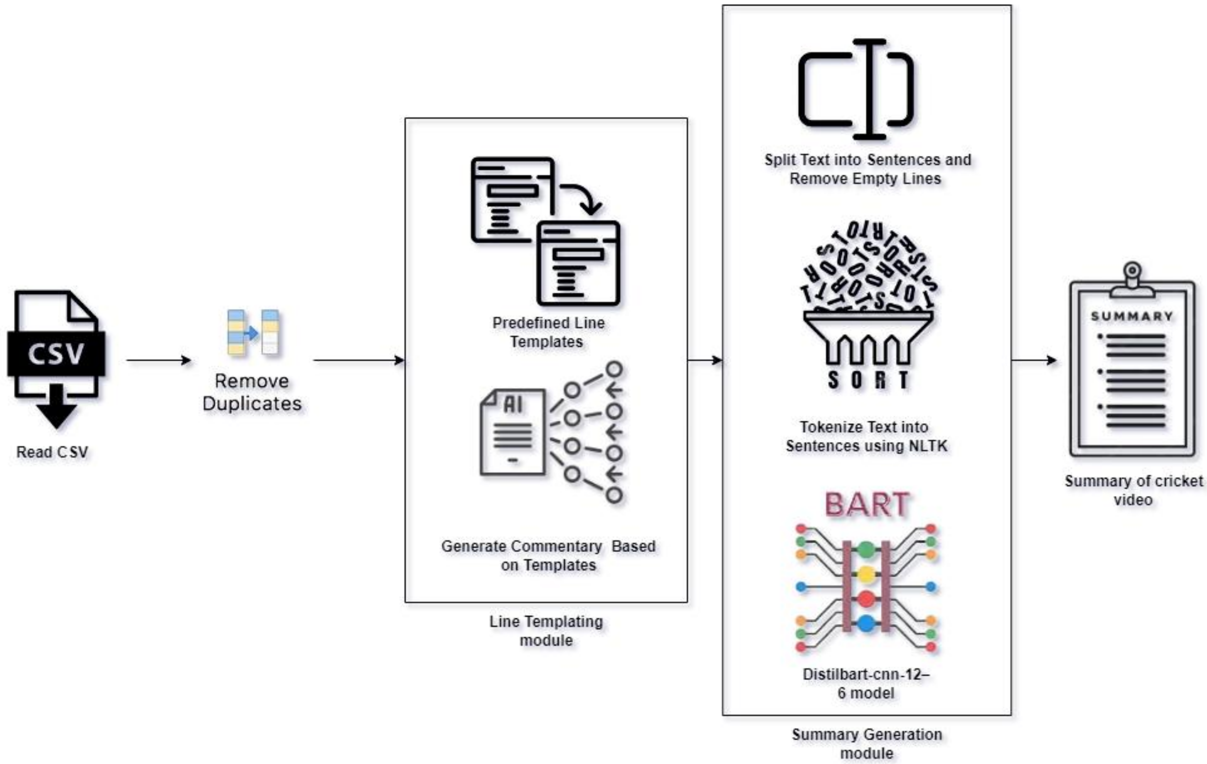


Figure 7: Workflow of the Architecture explaining the process of Line Templating and Summarization.

The summarization function operates on a separate textual input, specifically "gameplay_sentences.txt," to generate comprehensive summaries using state-of-the-art natural language processing techniques. Beginning with tokenization and sentence splitting, the function utilizes NLTK and BART model checkpoints to divide the text into manageable chunks. Each chunk is then summarized using the BART model, ensuring coherence and relevance in the generated summaries. Finally, the summarized text is written to the file "summary.txt," consolidating the insights gleaned from the original gameplay sentences. Together, these functions synergistically enable the generation of informative and digestible summaries encapsulating key aspects of cricket match analyses. By harnessing structured data and advanced natural language processing capabilities, the Line Templating and Summarization Module empowers users to derive actionable insights.

4.3 UML Diagrams

A UML (Unified Modeling Language) diagram is a visual representation used in software engineering to depict the structure and behavior of a system. It employs standardized symbols and notation to illustrate various aspects of the system's architecture, such as classes, objects, relationships, and interactions. UML diagrams aid in communication among stakeholders by providing a clear and concise overview of the system's design and functionality. They serve as blueprints for software development, facilitating the understanding, analysis, and design of complex systems, thereby enhancing the efficiency and effectiveness of the development process.

4.3.1 Use Case Diagram

The provided UML diagram is a use case diagram, depicting interactions between actors and system functionalities. An actor labeled "User" interacts with the system, represented as a rectangle named "Crikywiki." Within this system, the user can upload a cricket video ("Upload Cricket Video") and subsequently view the generated text summary ("View Summary"). This diagram outlines the core functionalities of a system designed to summarize cricket videos using neural networks. It succinctly illustrates the primary interactions between the user and the system, emphasizing the specific use case of video summarization within the context of cricket content.

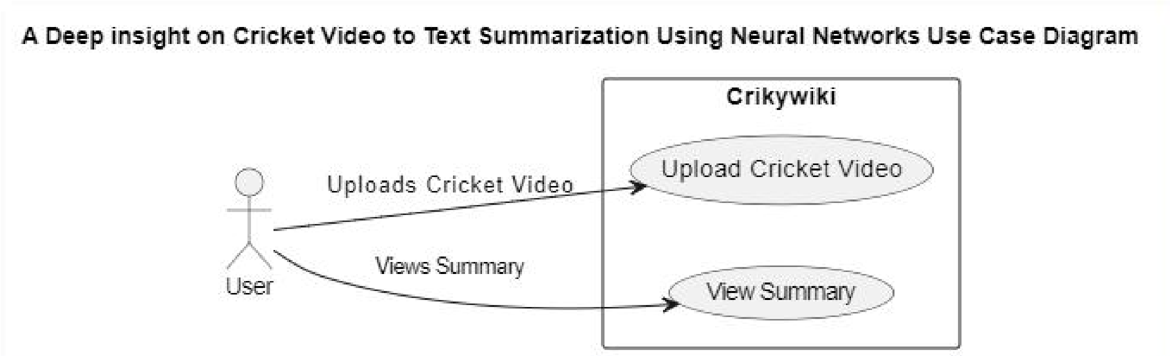


Figure 8: Use Case Diagram

4.3.2 Class Diagram

The provided UML diagram is a class diagram, delineating the structure and relationships among classes in a system. It illustrates the components and interactions within a system designed for cricket video summarization into text using neural networks. Key classes include VideoProcessor, FrameExtractor, TextExtractor, DataExporter, and Summarizer, each responsible for specific tasks such as video processing, frame extraction, text extraction, data exporting, and text summarization. Associations between classes denote dependencies and data flow, outlining how these components collaborate to achieve the system's objective of converting cricket videos into summarized text representations.

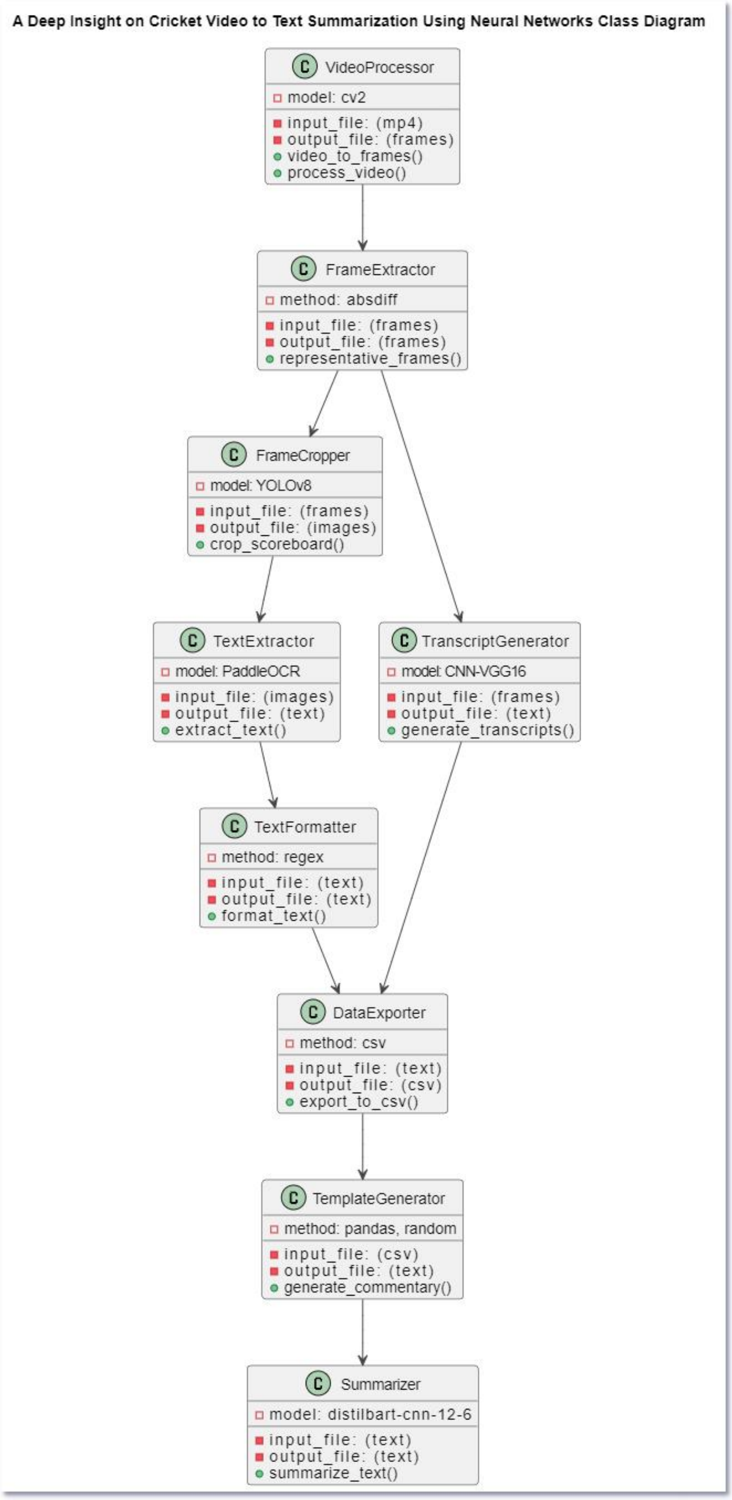


Figure 9: Class Diagram

4.3.3 Sequence Diagram

The provided UML diagram is a sequence diagram, outlining the chronological sequence of interactions between components within a system. It illustrates the process of converting cricket videos into summarized text using neural networks. The user initiates the video processing, triggering the activation of the video processor, which orchestrates subsequent actions. Frames are extracted, analyzed, and processed for text extraction and formatting. Data, including transcripts and scoreboard information, is exported. Commentary templates are generated before the final text summarization. Each step involves the activation and deactivation of relevant components, showcasing the systematic flow of operations within the system.

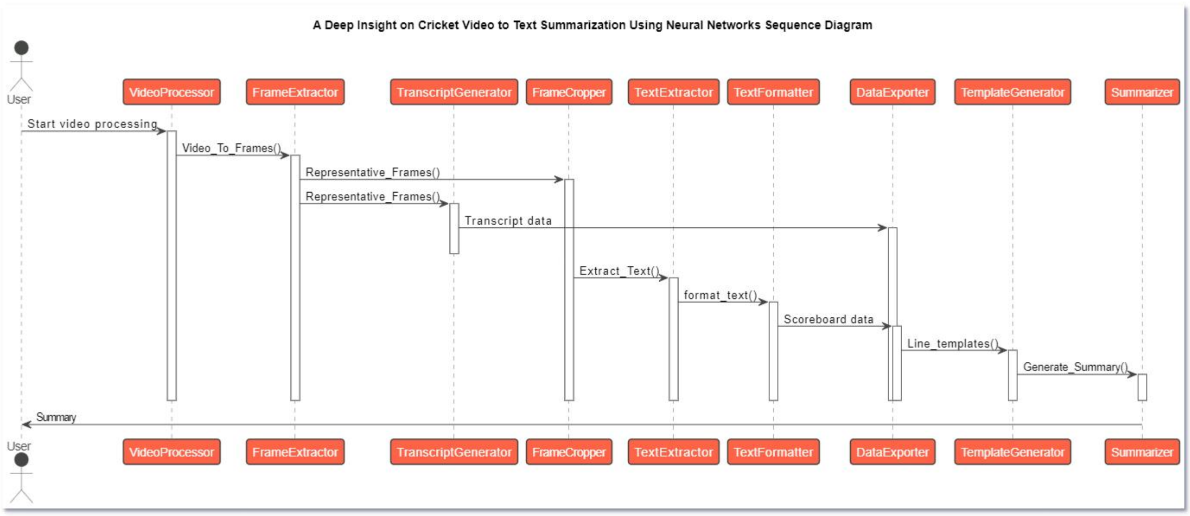


Figure 10: Sequence Diagram

4.3.4 Activity Diagram

The provided UML diagram is an activity diagram, illustrating the sequential flow of activities within a system. It outlines the process of converting cricket videos into text summaries using neural networks. Activities include parameter initialization, video file opening, frame extraction, scoreboard detection, transcript generation, and text extraction. Decision points determine successful execution paths, while error handling ensures graceful termination in case of failure. Activities such as data structuring, CSV conversion, and duplicate row removal optimize data processing. This diagram offers a detailed overview of the systematic steps involved in the cricket video summarization process, emphasizing activity sequencing and error management.

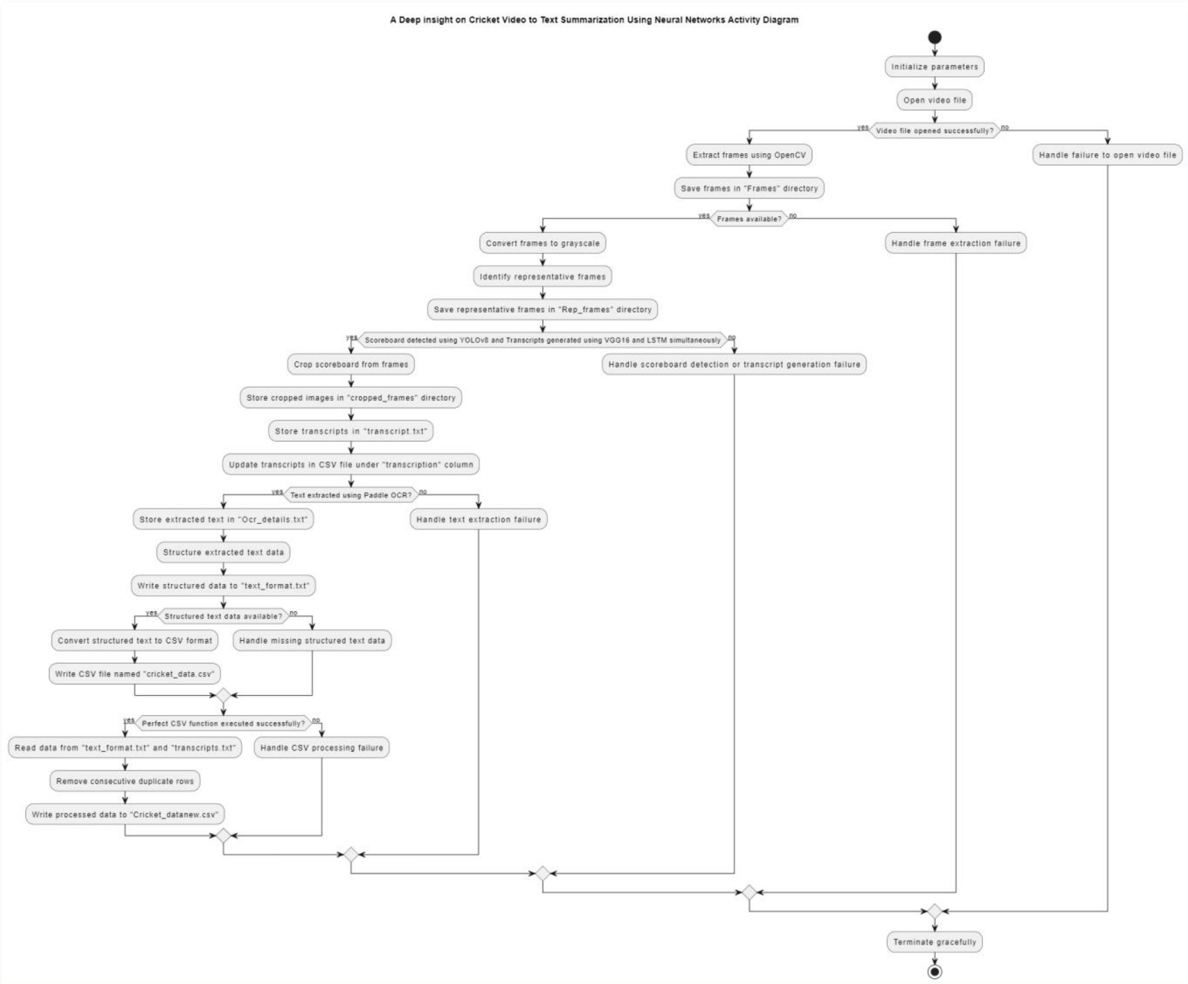


Figure 11: Activity Diagram

4.3.5 Deployment Diagram

The provided UML diagram is a deployment diagram, illustrating the physical deployment of system components across various servers. It outlines the architecture for the "Deep Insight on Cricket Video to Text Summarization Using Neural Networks" system. Servers include a user server, a web server, a video processing server, a data processing server, and a storage server. Components within each server represent specific functionalities, such as video processing, data processing, and storage. Arrows depict the flow of interactions between servers and components, illustrating how user requests traverse through the system for cricket video summarization. This diagram offers a clear depiction of the system's deployment structure, facilitating understanding of component distribution and interactions.

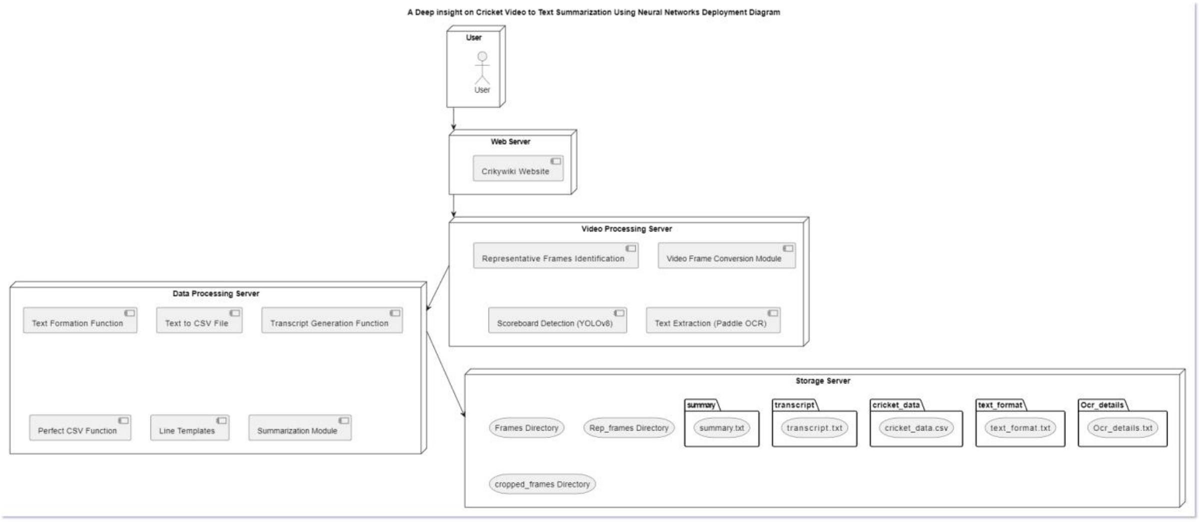


Figure 12: Deployment Diagram

4.3.6 Component Diagram

The provided UML diagram is a component diagram, depicting the system's modular structure and interconnections. It illustrates the "Deep Insight on Cricket Video to Text Summarization Using Neural Networks" system's architecture. Components within the "Video Processing System" package encompass functionalities such as frame extraction, conversion, representative frame identification, scoreboard detection, text extraction, formation, transcript generation, CSV processing, line template generation, summarization, and error handling. The "Data Storage" database stores relevant files and folders, while the "User Interface" frame represents components for user interaction. Arrows denote dependencies and interactions between components, outlining the sequential flow of data and processing steps within the system.

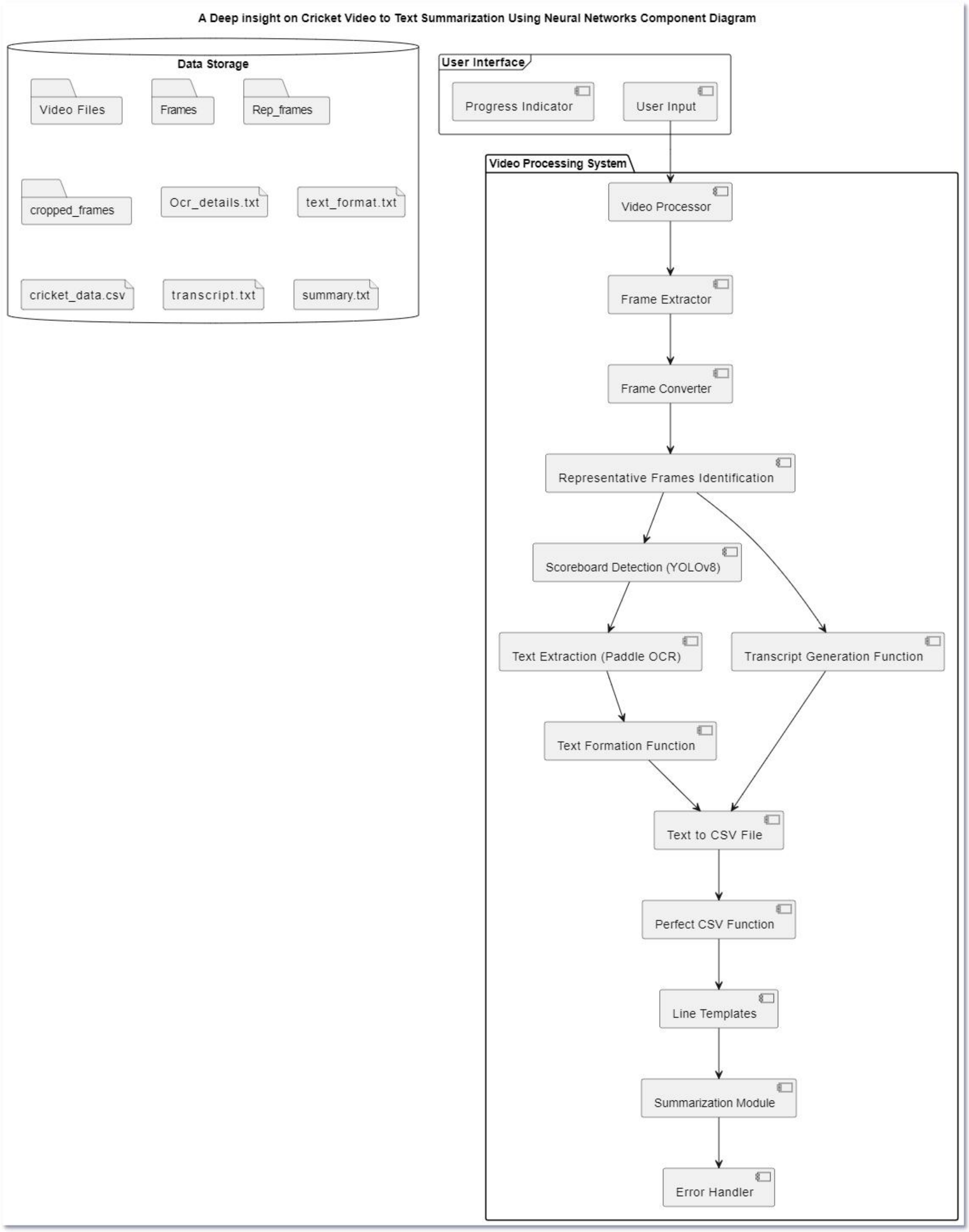


Figure 13: Component Diagram.

CHAPTER 5

IMPLEMENTATION

5.1 IMPLEMENTATION WITH HYPOTHETICAL SCENARIOS

This subsection evaluates the use of application in various scenarios involving different hypothetical situations.

S.No.	Scenario	Result
1	Scoreboard present in frame	Player names and scores, bowler’s information, team information (score and wickets) and predicted action.
2	Scoreboard not present in frame	Only the predicted action based on frame content.

Table 3: Possible combinations of data that can extracted from video frames

5.1.1 Scoreboard Present in Frame

When the scoreboard is detected in the frame, the system proceeds with extracting relevant textual information and predicting actions:

Text Extraction from Scoreboard: The system utilizes YOLOv8 to detect the scoreboard in the frame. Once detected, the region of interest (ROI) containing the scoreboard is extracted. Optical Character Recognition (OCR) is applied to the scoreboard region to extract textual information such as player names, scores, bowler's information, and team details (score and wickets).

Action Prediction: Concurrently, the system employs a CNN VGG16 model and LSTM encoder-decoder, fine-tuned on cricket action images. Using the fine-tuned model, it predicts the action happening in the frame, such as "batsman hits the ball" or "fielder catches the ball".

Summary Generation: The extracted textual information from the scoreboard, along with the predicted action are combined to form comprehensive gameplay sentences. These are given as input to Distilbart-CNN summary model to generate the final concise and contextual summary.

5.1.2 Scoreboard Not Present in Frame

When the scoreboard is not detected in the frame, the system relies solely on action prediction:

Action Prediction: Since the scoreboard is absent, the system skips the OCR step as there's no relevant textual information to extract. It directly utilizes the CNN model VGG16 and LSTM encoder-encoder to predict the action happening in the frame.

In both cases, the system adapts its processing based on the presence or absence of the scoreboard in the frame. If the scoreboard is detected, it provides a detailed summary including textual information from the scoreboard along with the predicted action. If the scoreboard is not present, it offers a summary solely based on the predicted action.

5.2 SOURCE CODE

index.html

```
<!DOCTYPE html>
<html>
<head>
            <title>CrikyWiki</title>
            <meta name="viewport" content="width=device-width, initial-scale=1.0">
            <link rel="icon" type="image/x-icon" href="{{url_for('static', filename='images/favicon.png') }}">
            <link rel="stylesheet" type="text/css" href="{{url_for('static',filename='css/style.css') }}">
</head>

<body>
   <header>
            <div class="logo">
                <img src="{{url_for('static', filename='images/iconlogo6.png') }}" alt="Berger Hut Logo">
    </div>
   <nav>
    <ul>
        <li><a href="#home">Home</a></li>
        <li><a href="#about">About</a></li>
        <li><a href="#services">Services</a></li>
                        <li><a href="#faq">FAQ</a></li>
                        <li><a href="#contact">Contact</a></li>
            </ul>
   </nav>
   </header>

   <section class="hero">
            <div class="hero-content">
                        <h1>Welcome to CrikyWiki!</h1>
                        <p>- Your Ultimate Match Summarization Tool</p>
                        <a href="#services" class="btn">Get Started</a>
            </div>
   </section>
   <section class="about dark-theme">
    <div class="about-content">
        <h2>About CrikyWiki</h2>
        <p>Welcome to CrikyWiki, where we revolutionize your cricket experience like            never       before!      With
CrikyWiki, you can transform every cricket match into an            immersive  journey  through  our  cutting-edge  match
summarization tool.
        </p>
        <p>Our platform empowers you to upload cricket match videos and instantly             receive  comprehensive  textual
summaries of all the thrilling action. Gone are              the days of sifting through hours of footage to relive the highlights.
CrikyWiki        condenses the excitement into concise, informative summaries that capture every            pivotal moment of
the game.
        </p>
        <p>Whether you're a passionate cricket enthusiast, a casual viewer, or a sports           analyst,  CrikyWiki  is  your
ultimate companion for staying updated on the latest              matches and reliving the excitement of past games. Join us
and discover a new              way to experience the world's most beloved sport – cricket – with CrikyWiki.
        </p>
    <br>
    <a href="#services" class="btn">Check out our new features</a>
    </div>
    <div class="about-image">
        <img src="{{url_for('static', filename='images/about-image.png') }}" alt="About Image">
    </div>
</section>

<section class="menu">
    <h2>How CrikyWiki Works</h2>
    <div class="menu-items">
          <div class="menu-item">
        <img src="{{url_for('static', filename='images/upload.png') }}" alt="Burger 1">
```

```
        <h3>Step-1</h3>
        <p>Upload your cricket match video. </p>
          </div>
          <div class="menu-item">
        <img src="{{url_for('static', filename='images/getsummary.png') }}" alt="Burger 2">
        <h3>Step-2</h3>
        <p>Click "Generate" to receive an instant summary. </p>
          </div>
          <div class="menu-item">
        <img src="{{url_for('static', filename='images/read.png') }}" alt="Burger 3">
        <h3>Step-3</h3>
        <p>A Dive into the key moments of the match effortlessly! </p>
          </div>
    </div>
</section>
<section class="upload-section">
    <h3>Upload Your Cricket Match Video</h3>
    <form action="/upload" method="post" enctype="multipart/form-data">
    <input type="file" name="file" accept=".mp4" id="uploadBtn" style="display:none;"  required>
    <label for="uploadBtn"><i class="fa-solid fa-upload"></i> Upload </label>
    <button id="submit-button" type="submit" class="fa-solid fa-upload"> Submit </button>
    </form>
</section>
<section class="summary-section">
    <div class="summary-container">
      <div class="summary-content-button">
       <h3>Get Summary</h3>
       <button id="summary-btn" onclick="generatesummary()">Generate</button>
      </div>
    <div id="summary-text" class="output-box"></div>
      <div id="error-message" style="color: #c0392b;">
      </div>
    </div>
</section>
<section class="testimonials">
    <h2>What Our Customers Say</h2>
    <div class="testimonial">
      <img src="{{url_for('static', filename='images/customer1.jpg')}}" alt="Customer 1">
      <p>"CrikyWiki is simply amazing! It has revolutionized how I experience cricket matches. Highly
recommended!"</p>
      <h4>- John Doe, Cricket Enthusiast</h4>
    </div>
    <div class="testimonial">
      <img src="{{url_for('static', filename='images/customer2.jpg') }}" alt="Customer 2">
      <p>"I never knew summarizing cricket matches could be this easy. Thanks, CrikyWiki!"
      </p>
      <h4>- Jane Smith, Sports Journalist</h4>
    </div>
</section>
<section class="gallery">
    <h2>New Features Coming Soon!</h2>
    <div class="image-grid">
      <div class="image-item">
         <img src="{{ url_for('static', filename='images/real-time-analysis.jpeg') }}" alt="Image 1">
       <p>Real-Time Match Analysis</p>
    </div>
    <div class="image-item">
      <img src="{{url_for('static', filename='images/personalised-summaries.jpeg') }}" alt="Image 2">
      <p>Personalized Match Summaries</p>
    </div>
    <div class="image-item">
      <img src="{{url_for('static', filename='images/lang.jpg') }}"    alt="Image 3">
      <p>Multilingual Summaries</p>
    </div>
    <div class="image-item">
      <img src="{{url_for('static', filename='images/sport.jpg') }}"    alt="Image 4">
```

```
        <p>Various Sports Summaries</p>
      </div>
      </div>
</section>

<section class="contact">
      <div class="contact-container">
        <h2>Contact Us</h2>
        <div class="contact-info">
        <div class="info-item">
         <i class="fas fa-map-marker-alt"></i>
         <p>123 Main Street, City, Country</p>
        </div>
      </div>
      <form class="contact-form">
         <input type="text" name="name" placeholder="Your Name" required>
         <input type="email" name="email" placeholder="Your Email" required><textarea name="message"
placeholder="Your Message" rows="3" required>
         </textarea>
         <button type="submit">Send Message</button>
      </form>
      </div>
</section>
<footer class="footer">
      <div class="footer-content">
        <div class="footer-logo">
         <img src="{{url_for('static', filename='images/iconlogo6.png') }}" alt="Logo">
        </div>
        <br><br>
        <nav class="footer-links">
         <a href="#">Home</a>
         <a href="#">About</a>
         <a href="#">Services</a>
         <a href="#">Testimonials</a>
         <a href="#">New Features</a>
         <a href="#">FAQ</a>
         <a href="#">Contact</a>
        </nav>
        <br>
        <div class="footer-social">
         <a href="#"><i class="fab fa-facebook"></i></a>
         <a href="#"><i class="fab fa-twitter"></i></a>
         <a href="#"><i class="fab fa-instagram"></i></a>
        </div>
      </div>
      <p class="footer-text">&copy; Major Project- Cricket Video to Text Summarization Using   Neural Networks</p>
</footer>
<script>
      document.getElementById('summary-btn').addEventListener('click', function() {
      // Reset error message
      document.getElementById("error-message").innerHTML = "";
      // Fetch the summary content from the server
      fetch('/get_summary')
      .then(response =>
      {
         if (!response.ok)
         {
                        throw new Error('Network response was not ok');
         }
                  return response.text();
      })
      .then(content =>
      {
                  // Display the summary content
                  document.getElementById('summary-text').innerHTML = content;
      })
```

```
    .catch(error =>
    {
                    // Display error message
                    console.error('There was a problem fetching the summary:', error);
                    document.getElementById('summary-text').innerText = "Error fetching summary. Please try again later.";
    });
   });
</script>
</body>
</html>
```

styles.css

```
body {
    font-family: 'Noto Sans', sans-serif;
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

/* Header */
header {
    background-color: #141414;
    padding: 20px;
    display: flex;
    align-items: center;
    justify-content: space-between;
}

.logo img {
    height: 80px;
    width: 300px;
}

nav ul {
    list-style: none;
    margin: 0;
    padding: 0;
}

nav ul li {
    display: inline-block;
    font-size: 22px;
    margin-right: 15px;
}

nav ul li a {
    text-decoration: none;
    color: white;
    font-weight: bold;
}

nav ul li a:hover {
    color: #ff0000;
}

/* Hero */

.hero {
    background-image: linear-gradient(rgba(0,0,0,0.4),rgba(0, 0, 0, 0.4)), url("{{url_for('static', filename='images/hero-
background15.jpg') }}");
    background-size: cover;
    background-position: center;
    height: 100vh;
    display: flex;
```

```
    justify-content: center;
    align-items: center;
    text-align: center;
    color: #ffffff;
}

.hero-content {
    max-width: 600px;
}

.hero-content h1 {
    font-size: 48px;
    font-style: thin;
    margin-bottom: 10px;
    max-width: 600px;
}

.hero-content p {
    font-size: 24px;
    font-style: italic;
    margin-bottom: 20px;
}

.btn {
    display: inline-block;
    background-color: #ff0000;
    color: #ffffff;
    padding: 20px 30px;
    border-radius: 7px;
    text-decoration: none;
    font-weight: bold;
    transition: background-color 0.3s ease;
}

.btn:hover {
    background-color: #e60000;
}

/* About */
.about {
    display: flex;
    justify-content: space-between;
    padding: 80px 20px;
    background-color: #141414;
}

.about-content {
    flex: 1;
    max-width: 600px;
    color: #ffffff;
}

.about h2 {
    font-size: 60px;
    font-style: times new roman;
    margin-bottom: 20px;
}

.about-image {
    flex: 0.1;
    flex-direction: row;
    text-align: left; /* Align the content (image) to the left */
}

/* Upload Section */
.upload-section {
```

```
    padding: 80px 20px;
    font-style: calibri;
    text-align: center;
    background-color: #2b2b2b;
}

.upload-section h3 {
    font-size: 36px;
    margin-bottom: 20px;
    color: #ffffff;
}

/* Summary Section */
.summary-section {
    padding: 80px 20px;
    background-color: #141414;
}

.summary-container {
    max-width: 1500px;
    margin: 0 auto;
    background-color: #2b2b2b;
    padding: 30px;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);
}

.summary-container h3 {
    color: #fff;
    font-size: 36px;
    font-style: calibri;
    margin-bottom: 20px;
}

.output-box {
    background-color: #fff;
    color: #000;
    padding: 30px;
    border-radius: 5px;
    margin-top: 20px;
    font-size: 18px;
}

/* Testimonials */
.testimonials {
    padding: 80px 20px;
    text-align: center;
    background-color: #1a1a1a;
}

.testimonials h2 {
    font-size: 56px;
    margin-bottom: 40px;
    color: #fff;
}

.testimonial {
    max-width: 600px;
    margin: 0 auto 40px;
    text-align: left;
}

.testimonial img {
    display: block;
    width: 80px;
    height: 80px;
```

```
    border-radius: 50%;
    margin: 0 auto 20px;
}

.testimonial p {
    font-size: 24px;
    font-style: italic;
    max-width: 600px;
    margin-bottom: 20px;
    color: #fff;
}

.testimonial h4 {
    font-size: 18px;
    font-weight: bold;
    color: #fff;
}

/* Gallery */
.gallery {
    padding: 80px 20px;
    text-align: center;
    background-color: #2c2c2c;
}

.gallery h2 {
    font-size: 56px;
    margin-bottom: 40px;
    color: #fff;
}

.image-grid {
    display: grid;
    grid-template-columns: repeat(4, 1fr);
    grid-gap: 20px;
}

.image-item img {
    width: 100%;
    height: auto;
    border-radius: 5px;
}

.image-item {
    position: relative;
}
.image-item p {
    position: absolute;
    bottom: 0;
    left: 0;
    right: 0;
    background-color: rgba(0, 0, 0, 0.7);
    color: #fff;
    padding: 10px;
    margin: 0;
}

/* Contact */
.contact {
    padding: 80px 20px;
    text-align: center;
    background-color: #141414;
    color: #fff;
}

.contact-container {
```

```
    max-width: 600px;
    margin: 0 auto;
}

.contact h2 {
    font-size: 56px;
    margin-bottom: 40px;
}

.contact-info {
    display: flex;
    justify-content: center;
    margin-bottom: 40px;
}

.info-item {
    margin: 0 20px;
    text-align: center;
}

.info-item i {
    font-size: 24px;
    margin-bottom: 20px;
}

.contact-form input, .contact-form textarea {
    display: block;
    width: 100%;
    padding: 10px;
    margin-bottom: 20px;
    border-radius: 5px;
    border: none;
}

.contact-form textarea {
    resize: vertical;
}

.contact-form button {
    display: inline-block;
    background-color: #ff0000;
    color: #fff;
    padding: 10px 20px;
    border-radius: 5px;
    text-decoration: none;
    font-weight: bold;
    transition: background-color 0.3s ease;
}

.contact-form button:hover {
    background-color: #e60000;
}

/* Footer */
.footer {
    background-color: #141414;
    padding: 40px 20px;
    color: #fff;
    text-align: center;
}

.footer-content {
    display: flex;
    flex-direction: column;
    align-items: center;
    margin-bottom: 10px;
```

```
}

.footer-logo img {
    max-width: 300px;
    height: 80px;

}

.footer-links a {
    color: #fff;
    margin: 0 10px;
    text-decoration: none;
}

.footer-social a {
    color: #fff;
    margin: 0 5px;
    text-decoration: none;
}

.footer-text {
    font-size: 14px;
}

/* Upload Button */
#uploadBtn {
    display: none;
}
.summary-content-button{
    text-align: center;
}

label[for="uploadBtn"] {
    display: inline-block;
    text-transform: uppercase;
    color: #fff;
    background: #c0392b;
    text-align: center;
    padding: 15px 40px;
    font-size: 18px;
    letter-spacing: 1.5px;
    user-select: none;
    cursor: pointer;
    box-shadow: 5px 15px 25px rgba(0, 0, 0, 0.35);
    border-radius: 3px;
}

label[for="uploadBtn"] i {
    font-size: 20px;
    margin-right: 10px;
}

label[for="uploadBtn"]:active {
    transform: scale(0.9);
}

#submit-button {
    display: inline-block;
    text-transform: uppercase;
    color: #fff;
    background: #c0392b;
    text-align: center;
    padding: 15px 40px;
    font-size: 18px;
    letter-spacing: 1.5px;
    user-select: none;
```

```css
    cursor: pointer;
    box-shadow: 5px 15px 25px rgba(0, 0, 0, 0.35);
    border-radius: 3px;
}

/* Summary Button */
#summary-btn {
    display: inline-block;
    text-align: center;
    margin-left: auto;
    text-transform: uppercase;
    color: #fff;
    background: #c0392b;
    text-align: center;
    padding: 15px 40px;
    font-size: 18px;
    letter-spacing: 1.5px;
    user-select: none;
    cursor: pointer;
    box-shadow: 5px 15px 25px rgba(0, 0, 0, 0.35);
    border-radius: 10px;
    transition: transform 0.2s ease-in-out;
}

#summary-btn:hover {
    background-color: #e74c3c;
}

#summary-btn:active {
    transform: scale(0.9);
}
```

app.py

```python
from flask import Flask, render_template, request, redirect, url_for, jsonify
import os
import subprocess
import sys
app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = 'static/videos'
app.config['ALLOWED_EXTENSIONS'] = {'mp4'}

def allowed_file(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in app.config['ALLOWED_EXTENSIONS']

def run_main_script(video_path):
    virtual_env_activate_cmd = r"sai/Scripts/activate"
    script_path = r"allmodules.py"

    # Activate virtual environment
    subprocess.run([virtual_env_activate_cmd], shell=True)

    # Run the script
    subprocess.run([sys.executable, script_path, video_path], shell=True)

def read_result_file():
    result_file_path = 'gameplay_sentences.txt'
    try:
        with open(result_file_path, 'r') as file:
            content = file.read().replace('\n', ' ')
        return content
    except FileNotFoundError:
        return 'Result file not found.'

@app.route('/')
def index():
    return render_template('index.html')
```

```
@app.route('/upload', methods=['POST'])
def upload_file():
	if 'file' not in request.files:
		return redirect(request.url)

	file = request.files['file']

	if file.filename == '':
		return redirect(request.url)

	if file and allowed_file(file.filename):
		filename = os.path.join(app.config['UPLOAD_FOLDER'], 'cricket_video.mp4')
		file.save(filename)
		run_main_script(filename)
		return redirect(url_for('index')) # Redirect to the 'index' route
	else:
		return 'Invalid file format! Please upload an MP4 file.'

@app.route('/get_summary')
def get_summary():
	result_content = read_result_file()
	return jsonify(result_content)

if __name__ == '__main__':
	app.run(debug=True)
```

allmodules.py

```
import torch
import os
from ultralytics import YOLO
import cv2
import numpy as np
from PIL import Image
import os.path, sys
import re
import csv
import pandas as pd
from paddleocr import PaddleOCR, draw_ocr # main OCR dependencies
from matplotlib import pyplot as plt # plot images
import cv2
import numpy as np
import os
from pickle import load
from numpy import argmax
from keras.preprocessing.sequence import pad_sequences
from keras.applications.vgg16 import VGG16
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
from keras.applications.vgg16 import preprocess_input
from keras.models import Model
from keras.models import load_model
from matplotlib import pyplot as plt
#pip install transformers[sentencepiece]
from transformers import BartTokenizer, BartForConditionalGeneration
import nltk
import random

import csv
```

```
import transformers
from transformers import T5Tokenizer, T5ForConditionalGeneration, pipeline

def video_to_frames(input):
    input_video_path = input
    output_directory = "Frames/"
    os.makedirs(output_directory, exist_ok=True)
    # Open the video file
    video_capture = cv2.VideoCapture(input_video_path)
    if not video_capture.isOpened():
        print("Error opening video file")
        exit()
    # Initialize variables
    frame_count = 0
    try:
        # Loop through the video frames
        while True:
            # Read a frame from the video
            ret, frame = video_capture.read()
            name = './FRAMES/frame' + str(frame_count) + '.jpg'
            print('Creating...' + name)
            # Break the loop if no frame is retrieved
            if not ret:
                break
          # Save the frame
            frame_count += 1
            frame_filename = f'{output_directory}frame_{frame_count:04d}.jpg'
            cv2.imwrite(frame_filename, frame)
            # Display the frame (optional)
            if cv2.waitKey(1) & 0xFF == ord('q'):
                break
    except Exception as e:
        print(f"An error occurred in video_to_frames: {e}")
    finally:
        # Release the video capture object and close any open windows
        video_capture.release()
        cv2.destroyAllWindows()
    return output_directory

def representative_frames(input):
    input_frames_dir = input
    output_frames_dir = "R_frames1/"
    os.makedirs(output_frames_dir, exist_ok=True)
    try:
        # Initialize variables
        prev_frame = None
        threshold = 35    # Adjust this threshold as needed
        frame_count = 0
        # Loop through the input frames directory
        for filename in os.listdir(input_frames_dir):
            if filename.endswith('.jpg') or filename.endswith('.png'):
                # Read the frame
                frame_path = os.path.join(input_frames_dir, filename)
                frame = cv2.imread(frame_path)
                if frame is None:
                    continue
                # Convert frame to grayscale
                gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
                # Calculate frame difference
                if prev_frame is not None:
                    frame_diff = cv2.absdiff(gray_frame, prev_frame)
                    difference = cv2.mean(frame_diff)[0]
                    # If difference is below threshold, skip frame
                    if difference < threshold:
                        continue
                # Write the frame to the output directory with renamed file name
```

```
                output_frame_path = os.path.join(output_frames_dir,
f"frame_{frame_count:05d}.jpg")
                cv2.imwrite(output_frame_path, frame)
                # Increment frame count
                frame_count += 1
                # Store current frame as previous frame for next iteration
                prev_frame = gray_frame.copy()
    except Exception as e:
        print(f"An error occurred in representative_frames: {e}")
    finally:
        print("Execution completed.")
    return output_frames_dir

def crop_framrs(input):
    IMAGES_DIR = input
    model_path = "best.pt"
    model = YOLO(model_path)
    threshold = 0.7
    OUTPUT_DIR = "cropped_frames/"
    # Create output directory if it doesn't exist
    os.makedirs(OUTPUT_DIR, exist_ok=True)
    def crop_and_save_image(input_image_path, output_image_path, x1, y1, x2, y2):
        try:
            # Read the input image
            frame = cv2.imread(input_image_path)
            # Crop the image within the specified region
            roi = frame[int(y1):int(y2), int(x1):int(x2)]
            # Save the cropped image to the specified path
            cv2.imwrite(output_image_path, roi)
            print("Cropped image saved successfully at:", output_image_path)
        except Exception as e:
            print(f"Error occurred while processing {input_image_path}: {e}")
    os.makedirs(OUTPUT_DIR, exist_ok=True)
    try:
        # Iterate through the images in the input directory
        for image_file in os.listdir(IMAGES_DIR):
            if image_file.endswith('.jpg'):
                image_path = os.path.join(IMAGES_DIR, image_file)
                # Read the image
                frame = cv2.imread(image_path)
                # Perform object detection
                results = model(frame)[0]
                # Iterate through the detected objects
                for result in results.boxes.data.tolist():
                    x1, y1, x2, y2, score, class_id = result
                    if score > threshold:
                        # Define output image path
                        output_image_path = os.path.join(OUTPUT_DIR, "cropped_" + image_file)
                        # Crop and save the image within the specified ROI
                        crop_and_save_image(image_path, output_image_path, x1, y1, x2, y2)
    except Exception as e:
        print(f"An error occurred: {e}")
    return OUTPUT_DIR

def text_extraxt_ocr(input):
    # Setup model
    ocr_model = PaddleOCR(lang='en', use_gpu=False)
    def ocr_on_folder(folder_path, output_file):
        try:
            filenames = sorted([filename for filename in os.listdir(folder_path) if
filename.endswith(('.jpg', '.png', '.jpeg'))])
            with open(output_file, 'w', encoding='utf-8') as f:
                for filename in filenames:
                    try:
                        img_path = os.path.join(folder_path, filename)
                        frame_number = os.path.splitext(filename)[0].split('_')[-1]
```

```
                    result = ocr_model.ocr(img_path)
                    f.write(f"Frame Number: {frame_number}\n")
                    write_strings(result, f)
                    f.write("\n\n")
                except Exception as e:
                    print(f"Error occurred while processing {filename}: {e}")
        except Exception as e:
            print(f"Error occurred while opening or writing to the output file: {e}")
    def write_strings(result, file):
        for item in result:
            if isinstance(item, str):
                file.write(item + "\n")
            elif isinstance(item, list) or isinstance(item, tuple):
                write_strings(item, file)

    output="output_ocr.txt"
    # Call the function to perform OCR on all images in the "frames" folder
    ocr_on_folder(input, 'output_ocr.txt')
    return output

def crop(path1):
    def crop_and_save_bottom_half(image_path, output_path):
        # Read the image
        image = cv2.imread(image_path)

        # Get image dimensions
        height, width, _ = image.shape

        # Calculate midpoint for horizontal division
        midpoint = height // 2
        # Crop the bottom half of the image
        bottom_half = image[midpoint:, :]
        # Save the bottom half
        cv2.imwrite(output_path, bottom_half)

    # Input and output directories
    input_dir = path1
    output_dir = "output_images/"

    # Create output directory if it doesn't exist
    if not os.path.exists(output_dir):
        os.makedirs(output_dir)

    # Iterate over each image file in the input directory
    for filename in os.listdir(input_dir):
        # Check if the file is an image
        if filename.lower().endswith(('.png', '.jpg', '.jpeg', '.bmp')):
            # Construct input and output paths
            input_image_path = os.path.join(input_dir, filename)
            output_image_path = os.path.join(output_dir, filename)

            # Perform cropping and saving the bottom half three times
            for i in range(3):
                if i == 0:
                    crop_and_save_bottom_half(input_image_path, output_image_path)
                else:
                    crop_and_save_bottom_half(output_image_path, output_image_path)

    print("Image processing completed!")
    return output_dir

def text_formatting(input):

    text_output = "text_format.txt"
    # Combined regex patterns
    patterns = {
```

```
            "frame_no": r'^Frame Number: (\d{1,})$',
            "team": r'\b(?!RUN|REQ|OCR|SRI|REO)([A-Z]{2,3})([O0oa@e]?)?(\d+/\d+)?\b',
            "team_score": r'^\d+/\d+$',
            "striker": r'^(?=.*\*)(?!.*(?:NEED|TRAIL|TARGET|WIN))([A-Z]+(?:\s*[A-
Z]*)*)\s*\*\s*(\d+)\s*(?:\((\d+)\))?',
            "non_striker": r'^(?!.*(?:NEED|TRAIL|TARGET|WIN))([A-Z]+(?:\s*[A-
Z]*)*)\s*(\d+)\s*(?:\((\d+)\))?(?=\s|$)',
            "bowler": r'([A-Z]{4,})\s*(\d+)[/](\d+)',
            "overs": r'(?:OVERS\s*)?(\d{1,2}\.\d)(?!\s*KM/H)(?=\s|$)',
            "runrate": r'(RUN\s*RATE)\s*(\d+(\.\d+)?)',
            "reqrun": r'(REQ\.?\s*RATE)\s*(\d+(\.\d+)?)',
            "speed": r'(?:SPEED\s*)?(\d+(\.\d+)?\s*KM/H)',
            "action": r'(?:NEED|TRAIL|TARGET|WIN|BALLS)'
        }

        # Pre-compile regex patterns
        compiled_patterns = {key: re.compile(pattern) for key, pattern in patterns.items()}


# Function to extract information from a frame and write to file
        def extract_frame_info_and_write(frame_text, output_file, frame_number):
            values = {
                "frame_no": "N/A",
                "team_name": "N/A",
                "team_score": "N/A",
                "striker": {"striker_name": "N/A", "striker_runs": "N/A", "striker_balls": "N/A"},
                "non_striker": {"non_striker_name": "N/A", "non_striker_runs": "N/A",
"non_striker_balls": "N/A"},
                "bowler": {"bowler_name": "N/A", "bowler_runs": "N/A", "wickets": "N/A"},
                "overs": "N/A",
                "runrate": "N/A",
                "reqrun": "N/A",
                "speed": "N/A",
                "action": "N/A" # Initialize unmatched lines string
            }

            lines = frame_text.split('\n')

            for line in lines:
                for key, pattern in compiled_patterns.items():
                    match = pattern.match(line)
                    if match:
                        if key == "frame_no":
                            values["frame_no"] = match.group(1)    # Store frame number
                        elif key == "team":
                            values["team_name"] = match.group(1)
                            values["team_score"] = match.group(3) or "N/A"
                        elif key == "team_score":
                            values["team_score"] = match.group()
                        elif key == "striker":
                            values["striker"]["striker_name"] = match.group(1)
                            values["striker"]["striker_runs"] = match.group(2)
                            values["striker"]["striker_balls"] = match.group(3) or "N/A"
                        elif key == "non_striker":
                            values["non_striker"]["non_striker_name"] = match.group(1).strip()
                            values["non_striker"]["non_striker_runs"] = match.group(2)
                            values["non_striker"]["non_striker_balls"] = match.group(3) if
    match.group(3) else '0'
                        elif key == "bowler":
                            values["bowler"]["bowler_name"] = match.group(1)
                            values["bowler"]["bowler_runs"] = match.group(3)
                            values["bowler"]["wickets"] = match.group(2)
                        elif key == "overs":
                            values["overs"] = match.group(1)
                        elif key == "runrate":
                            values["runrate"] = match.group(2)
```

```
                elif key == "reqrun":
                    values["reqrun"] = match.group(2)
                elif key == "speed":
                    values["speed"] = match.group(1)
            elif re.search(patterns["action"],line):
               values["action"]=line

        # Write values to file
        for key, value in values.items():
            output_file.write(f"{key}: {value}\n")
        output_file.write("\n")
        return values

    try:
        # Read input from file
        with open(input, 'r') as input_file, open(text_output, 'w') as output_file:
            input_text = input_file.read()

            # Split text into frames using empty lines as separators
            frames = input_text.split('\n\n')

            completed_frames = 0
            # Process each frame separately

            for i, frame in enumerate(frames, 1):
                try:
                    frame_info = extract_frame_info_and_write(frame, output_file, i)
                    completed_frames += 1
                    print(f"Frame {i} completed. Total completed frames: {completed_frames}")
                except Exception as e:
                    print(f"An error occurred while processing frame {i}: {e}")

        print("Frame processing completed.")

    except Exception as e:
        print(f"An error occurred: {e}")

    return text_output

def to_csv(input):

    input_file = input
    output_file = "cricket_data.csv"

    # Initialize a list to store the cricket data
    cricket_data = []

    # Function to parse the input file and extract cricket data
    def parse_input_file(input_file):
        try:
            with open(input_file, "r") as file:
                current_frame = {}
                for line in file:
                    line = line.strip()
                    if line:
                        key, value = line.split(": ", 1)
                        if key.startswith("striker") or key.startswith("non_striker") or
key.startswith("bowler"):
                            # Extract nested data from string and convert to dictionary
                            nested_data = eval(value)
                            # Update current frame with nested data
                            current_frame.update(nested_data)
                        else:
                            current_frame[key] = value
                    else:
                        cricket_data.append(current_frame)
```

```
                current_frame = {}
        except Exception as e:
            print(f"Error occurred while parsing the input file: {e}")

    def write_to_csv(output_file):
        try:
            with open(output_file, "w", newline="") as csvfile:
                fieldnames = ["frame_no", "team_name", "team_score", "striker_name",
 "striker_runs", "striker_balls", "non_striker_name",                "non_striker_runs","non_striker_balls",
"bowler_name",                "bowler_runs", "wickets", "overs", "runrate", "reqrun", "speed", "action"]
                writer = csv.DictWriter(csvfile, fieldnames=fieldnames)

                # Write header
                writer.writeheader()

                # Write cricket data
                for data in cricket_data:
                    writer.writerow(data)
        except Exception as e:
            print(f"Error occurred while writing to the CSV file: {e}")

    # Parse the input file
    parse_input_file(input_file)

    # Write cricket data to CSV file
    write_to_csv(output_file)
    print("Cricket data has been successfully stored in", output_file)
    return output_file

# extract features from each photo in the directory
def extract_features(filename, model):
    image = load_img(filename, target_size=(224, 224))
    image = img_to_array(image)
    image = image.reshape((1, image.shape[0], image.shape[1], image.shape[2]))
    image = preprocess_input(image)
    feature = model.predict(image, verbose=0)
    return feature

# Map an integer to a word
def word_for_id(integer, tokenizer):
    for word, index in tokenizer.word_index.items():
        if index == integer:
            return word
    return None

# Generate a description for an image
def generate_desc(model, tokenizer, photo, max_length):
    in_text = 'startseq'
    for i in range(max_length):
        sequence = tokenizer.texts_to_sequences([in_text])[0]
        sequence = pad_sequences([sequence], maxlen=max_length)
        yhat = model.predict([photo, sequence], verbose=0)
        yhat = argmax(yhat)
        word = word_for_id(yhat, tokenizer)
        if word is None:
            break
        in_text += ' ' + word
        if word == 'endseq':
            break
    return in_text

def preprocess_images(filenames):
    images = [load_img(filename, target_size=(224, 224)) for filename in filenames]
    images = [img_to_array(image) for image in images]
    images = np.array(images)
    return preprocess_input(images)
```

```
def batch_extract_features(images, model):
    features = model.predict(images, verbose=0)
    return features

def test(directory):
    with open('tokenizer1.pkl', 'rb') as file:
        tokenizer = load(file)

    max_length = 25
    model = load_model('./Final_model.h5')
    lis = []
    with open('transcript.txt', 'w') as fobj:
        list1 = os.listdir(directory)
        batch_size = 8

        for i in range(0, len(list1), batch_size):
            batch_filenames = [os.path.join(directory, name) for name in list1[i:i+batch_size]]
            batch_images = preprocess_images(batch_filenames)
            batch_features = batch_extract_features(batch_images, vgg_model)

            for j, name in enumerate(list1[i:i+batch_size]):
                img = plt.imread(os.path.join(directory, name))
                plt.imshow(img)
                photo = batch_features[j:j+1]
                description = generate_desc(model, tokenizer, photo, max_length)
                description = ' '.join(description.split()[1:-1])
                fobj.write(f"Transcription: {description}\n")
                #plt.imshow(img)
                print(description)
                lis.append([i, description])
    print("Transcriptions Generated to transcript.txt!!")

    existing_data = pd.read_csv('cricket_data.csv')
    # Read the transcript data
    with open('transcript.txt', 'r') as file:
        transcription_lines = file.readlines()
    # Extract the transcription data
    transcription_data = pd.DataFrame(transcription_lines, columns=['transcription'])
    transcription_data['transcription'] =                transcription_data['transcription'].str.extract(r"Transcription: (.*)")
    # Update the existing 'transcription' column with new data
    existing_data['transcription'] = transcription_data['transcription']
    # Save the updated data back to a CSV
    fileexisting_data.to_csv('cricket_data_with_updated_transcription.csv', index=False)
    existing_data.to_csv('cricket_datanew.csv', index=False)
    print("Transcriptions appended to cricket_datanew.csv!!")

def perfect_csv(input):
    def remove_duplicate_rows(csv_file, ignore_column):
        try:
            df = pd.read_csv(csv_file)
            df_no_duplicates = df.drop_duplicates(subset=[col for col in df.columns if col !=
ignore_column])
            df_no_duplicates.to_csv(csv_file, index=False)
        except Exception as e:
            print(f"Error occurred while removing duplicate rows: {e}")

    def remove_consecutive_duplicates(csv_file, ignore_column):
        try:
            df = pd.read_csv(csv_file)
            df_no_consecutive_duplicates = df.drop_duplicates(subset=[col for col in
  df.columns if col != ignore_column])
            df_no_consecutive_duplicates.to_csv(csv_file, index=False)
        except Exception as e:
            print(f"Error occurred while removing consecutive duplicates: {e}")

def process_cricket_data():
```

```
    csv_file='cricket_datanew.csv'
    output_file='gameplay_sentences.txt'
    # Reading CSV data from a file using pd.read_csv
    cricket_data = pd.read_csv(csv_file)
    #fill_missing_values(cricket_data)
    generate_gameplay_summary(cricket_data)
    print(f"Gameplay sentences have been saved to {output_file}.")

def read_transcripts_from_file(file_path):
    with open(file_path, 'r') as file:
        transcripts = file.readlines()
    return transcripts

def summarygeneration():

    # Load tokenizer and model
    tokenizer = BartTokenizer.from_pretrained("./distilbart-cnn-12-6")
    model = BartForConditionalGeneration.from_pretrained("./distilbart-cnn-12-6")

    # Read input text file
    with open("gameplay_sentences.txt", "r") as file:
        file_content = file.read().strip()

    # Tokenize input text into sentences
    sentences = nltk.tokenize.sent_tokenize(file_content)

    # Maximum tokens in the longest sentence
    max_chunk_length = tokenizer.max_len_single_sentence - 2

    # Split sentences into chunks not exceeding max_chunk_length
    chunks = []
    chunk = ""
    length = 0
    for sentence in sentences:
        tokenized_sentence = tokenizer.tokenize(sentence)
        sentence_length = len(tokenized_sentence)

        if sentence_length > max_chunk_length:
            # Split long sentences into multiple chunks
            while sentence_length > 0:
                if sentence_length <= max_chunk_length:
                    chunks.append(' '.join(tokenized_sentence[:max_chunk_length]))
                    sentence_length = 0
                else:
                    chunks.append(' '.join(tokenized_sentence[:max_chunk_length]))
                    tokenized_sentence = tokenized_sentence[max_chunk_length:]
                    sentence_length = len(tokenized_sentence)
        else:
            combined_length = sentence_length + length
            if combined_length <= max_chunk_length:
                chunk += sentence + " "
                length = combined_length
            else:
                chunks.append(chunk.strip())
                chunk = sentence + " "
                length = sentence_length

    # Append remaining chunk
    if chunk.strip():
        chunks.append(chunk.strip())

    # Combine chunks into paragraphs
    paragraphs = []
    paragraph = ""
    for chunk in chunks:
        if len(paragraph.split()) < 100:    # Adjust the number of words per paragraph as needed
```

```
                paragraph += chunk + " "
            else:
                paragraphs.append(paragraph.strip())
                paragraph = chunk + " "
        if paragraph.strip():
            paragraphs.append(paragraph.strip())

        # Generate summaries for each paragraph
        with open("summary.txt", "w") as summary_file:
            for paragraph in paragraphs:
                inputs = tokenizer(paragraph, return_tensors="pt", max_length=1024, truncation=True)
                try:
                    summary_ids = model.generate(**inputs)
                    summary = tokenizer.decode(summary_ids[0], skip_special_tokens=True)
                    summary_file.write(summary + "<br><br>")
                except IndexError:
                    pass # Skip the current input and continue with the next one

def main():
    # Path to the input video file
    input = 'static/videos/cricket_video.mp4'
    # Directory to save the frames
    input2 = video_to_frames(input)
    print("Frames created")
    input3 = representative_frames(input2)
    print("representative frames created")
    input4 = crop_framrs(input3)
    print("Frames are cropped")
    inputx=crop(input3)
    input5 = text_extraxt_ocr(inputx)
    print("Text has been Extracted")
    print("started Text formatting!!")
    input6 = text_formatting(input5)
    print("Completed text formatting!!")
    input7 = to_csv(input6)
    vgg_model = VGG16()
    vgg_model = Model(inputs=vgg_model.inputs, outputs=vgg_model.layers[-2].output)
    test(input3)
    print("Transcripts appended to CSV Completed")
    input_csv='cricket_datanew.csv'
    input8 = perfect_csv(input7)
    print("perfect csv has been created")
    process_cricket_data()
    print("Processed CSV to Text file")
    output_file_path='summary.txt'
    transcripts = read_transcripts_from_file('gameplay_sentences.txt')
    summarygeneration()
    print()
    print("SUMMARY COMPLETED!!!!!")



if __name__ == "__main__":
    main()
```

CHAPTER 6

RESULTS

The below graph shows the absolute frame index difference varying throughout the frames. An image is considered only if the difference is greater than threshold and if an image has frame index less than threshold then that image is removed.

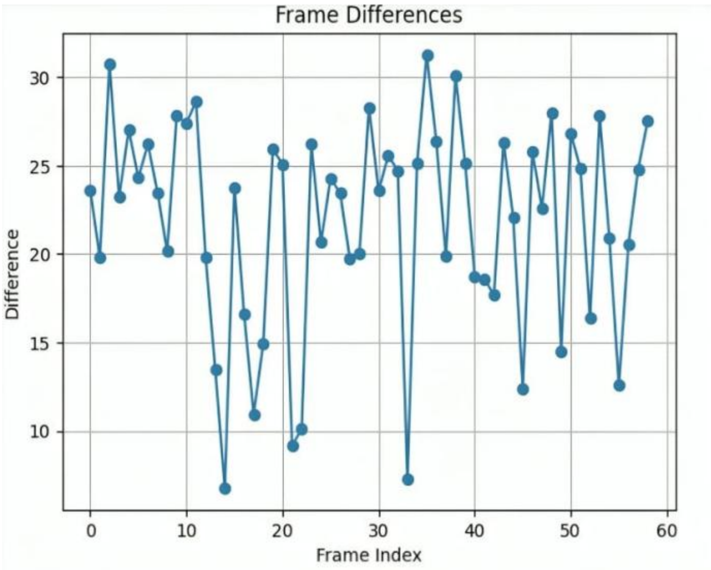


Figure 14: Absolute Difference Graph of Representative Frames

The below graph depicts the performance of Paddle OCR across various images. It shows how close the predicted results are to the actual ground truth.

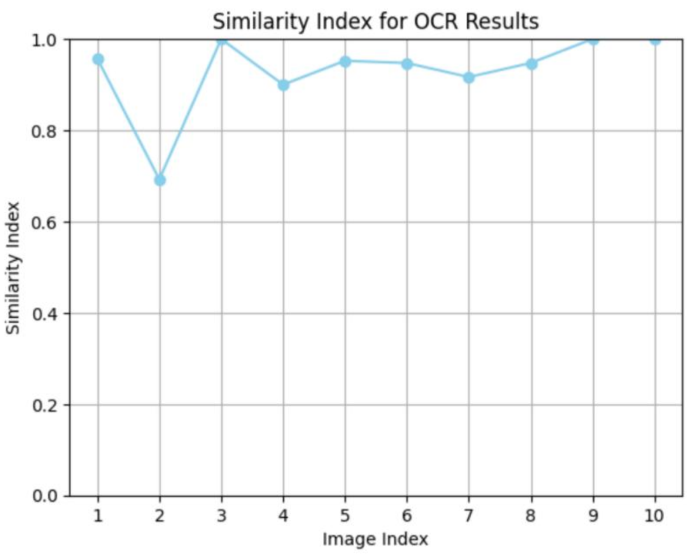


Figure 15: Similarity index graph of various images

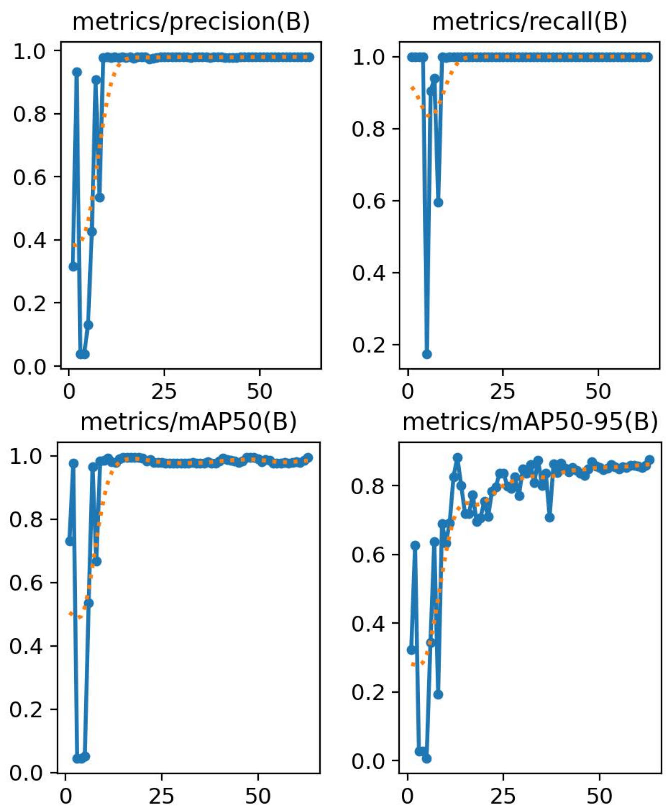


Figure 16: YoloV8 with SGD optimizer recall, and mean average precision (mAP) graph

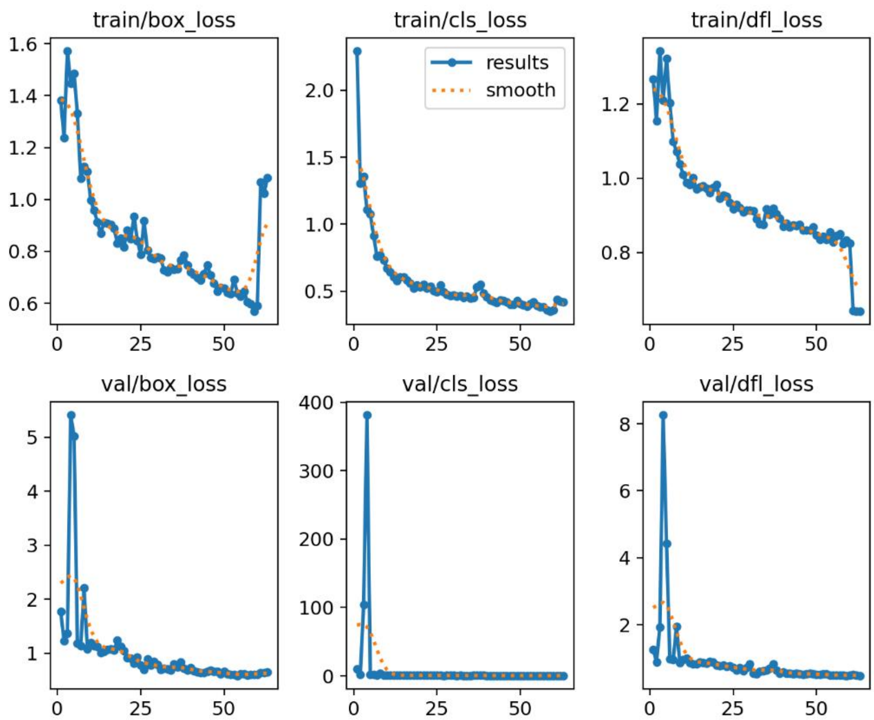


Figure 17: YoloV8 with SGD optimizer train/loss/val accuracy graph

The above graphs show all the plots of the YOLOv8 model i.e. box loss, objectness loss, classification loss, precision, recall, and mean average precision (mAP) over the training epochs for the training and validation set.

The below graph shows all the plots of the VGG16-LSTM encoder decoder model's accuracy and VGG16-LSTM encoder decoder model's loss history respectively over the training epochs for the training and validation set.

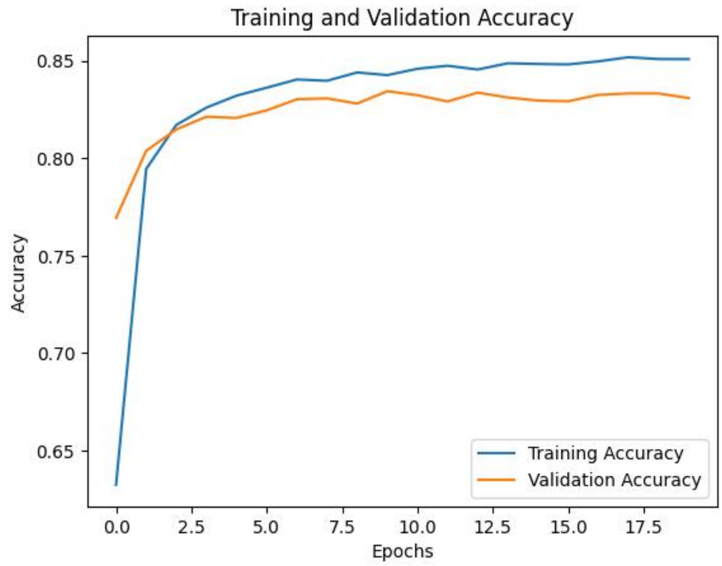


Figure 18: VGG16-LSTM Training and Validation Accuracy

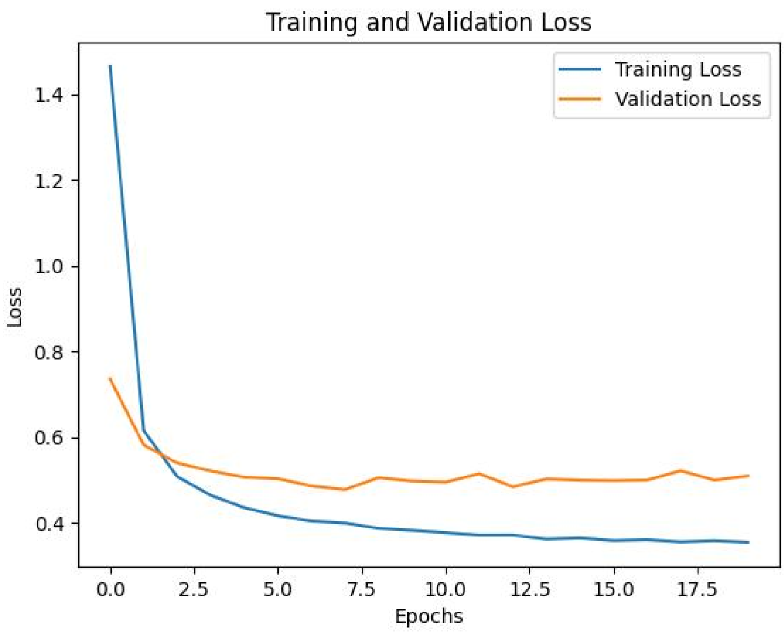


Figure 19: VGG16-LSTM Training and Validation Loss

The following figures shows the frontend website for getting the summaries of uploaded videos.

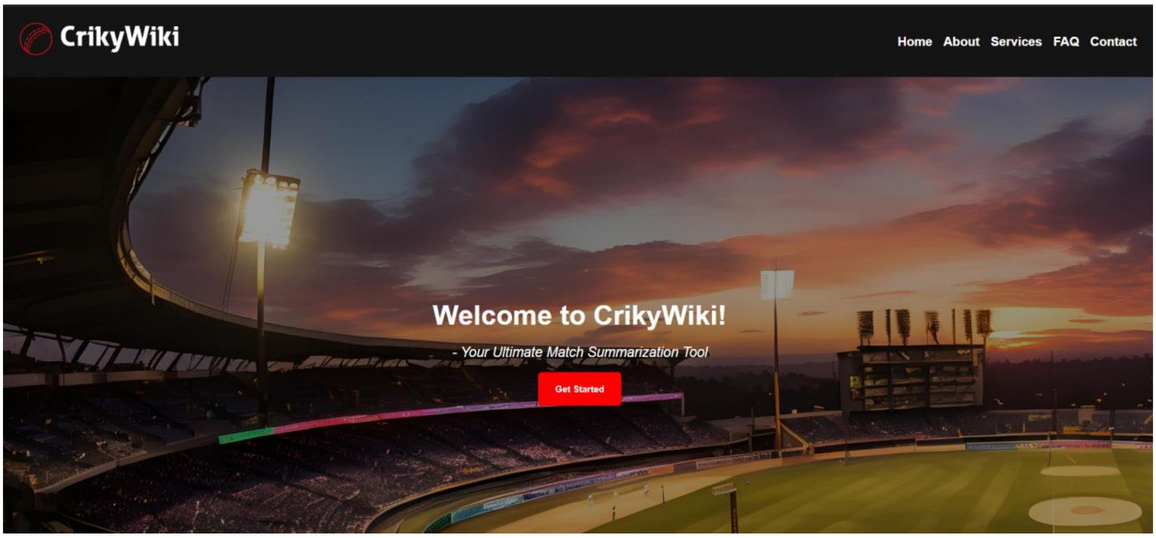


Figure 20: CrikyWiki fronted web interface

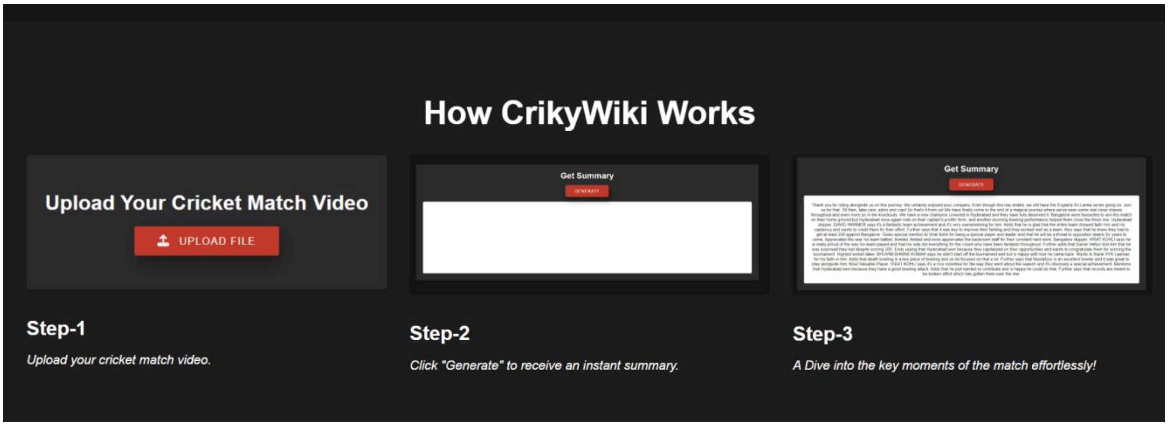


Figure 21: Working of website to get summaries

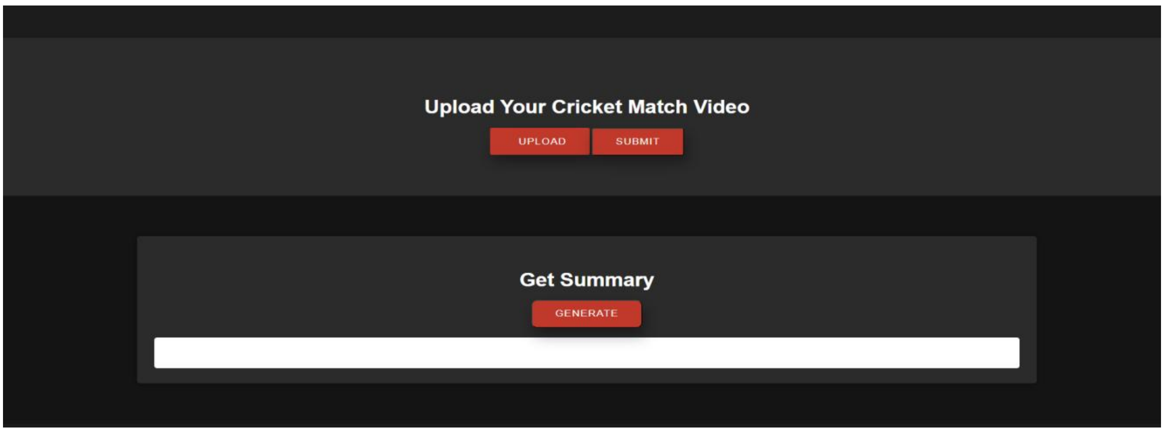


Figure 22: Web interface to upload and submit the video for summary

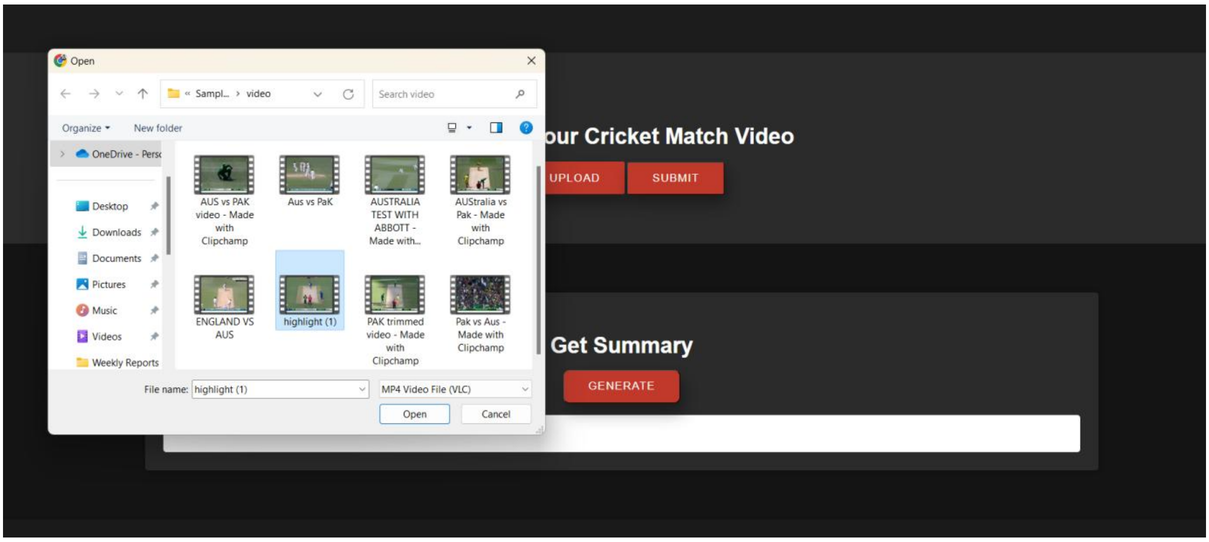


Figure 23: Uploading and submitting a Cricket Match Video

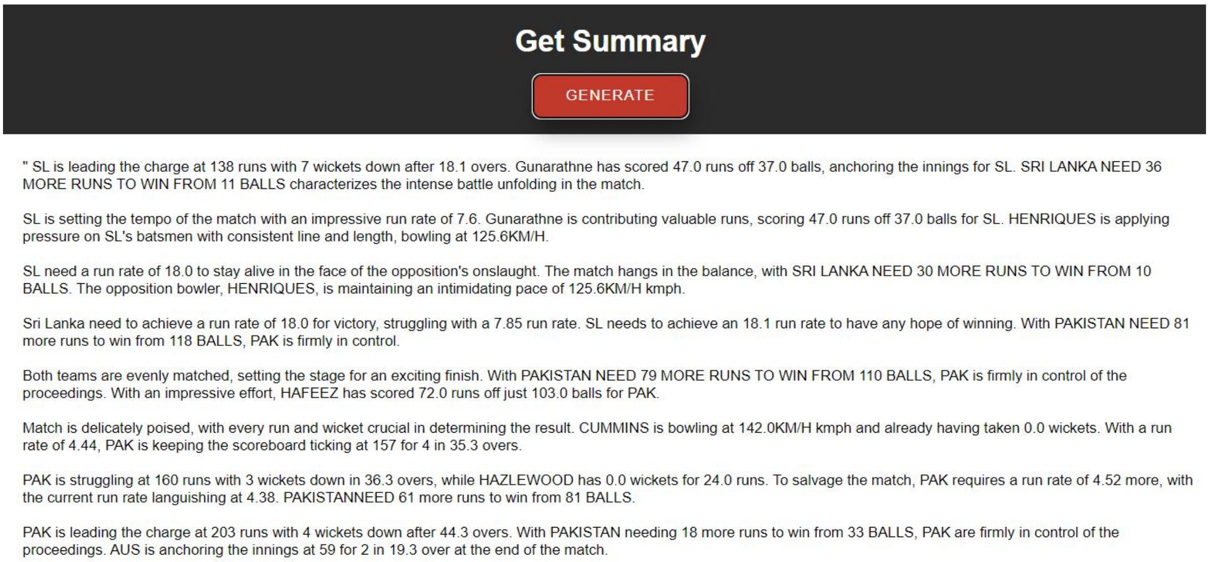


Figure 24: Result summary of uploaded cricket match video

CHAPTER 7

CONCLUSION

This research project endeavors to establish articulate models by employing a combination of techniques, including coarse segmentation through a heuristic mask of the VGG-16 CNN, to facilitate the transformation into an OCR reader. Furthermore, it integrates LSTM recurrent neural networks to generate summaries in the language of cricket video generalized reports. Prior to implementation, the model underwent rigorous training on annotated cricket images, enhancing its proficiency in summarizing cricket-related content. Despite these advancements, the inherent complexity of contextual nuances presents challenges in developing a script summarizer for sports events. Nevertheless, this study contributes significant findings to the field of cricket video summarization research, shedding light on the intricate process of automatic summarization. It underscores the importance of establishing robust neural networks, firmly grounded in cricket-specific knowledge, to achieve success in automatic video summarization on the cricket field.

By incorporating coarse segmentation and LSTM networks, this study introduces novel methodologies that advance the state-of-the-art in cricket video summarization. The utilization of heuristic masks and OCR technology enables the model to effectively interpret visual data, while LSTM networks facilitate the generation of coherent summaries. However, the complexity of sports events necessitates a deeper understanding of the contextual intricacies to ensure accurate summarization. Despite these challenges, the insights gleaned from this research pave the way for future advancements in automatic video summarization, particularly in the domain of sports analysis. It underscores the importance of merging artificial intelligence with domain-specific knowledge to achieve optimal results in summarization tasks.

FUTURE ENHANCEMENTS AND DISCUSSIONS

The implementation of the system significantly enhances the user experience by providing concise and informative summaries, thereby making cricket match analysis more accessible and time-efficient. This capability opens up diverse opportunities for applications, including sports analysis, broadcasting, and fan engagement. While the initial results are promising, ongoing refinement and validation are crucial to address potential limitations and improve effectiveness. Future research may focus on enhancing the system's ability to capture nuanced semantics and improve the quality of generated summaries, possibly through the integration of advanced machine learning techniques or user feedback mechanisms. Moreover, expanding the system's scope beyond cricket to encompass other sports or multimedia content could broaden its applicability, with collaborative efforts with industry stakeholders and domain experts facilitating the incorporation of domain-specific knowledge to enhance adaptability and robustness. Continuous innovation and iteration will be essential in unlocking the system's full potential, ensuring its relevance in enhancing user experiences, and facilitating comprehensive cricket match analysis.