

Institution Details



|  |  |  |  |
| --- | --- | --- | --- |
| **Province** | Sindh | **City** | Karachi |
| **Institution** | National University of Computer and Emerging Sciences (FAST-NU) | **Campus** | Karachi |
| **Department** | Artificial Intelligence | **Degree Level** | BS |
| **Degree Program** | Artificial Intelligence | **Telephone** |  |
| **Fax** |  | | |

Supervisor Details



|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Basit Jasani | **Gender** | Male |
| **Mobile** | +92 345 3603970 | **Office No** |  |
| **Email** | basit.jasani@nu.edu.pk | **Designation** |  |
| **Qualification** |  | | |

Co-Supervisor Details



|  |  |  |  |
| --- | --- | --- | --- |
| **Name** |  | **Gender** |  |
| **Mobile** |  | **Office No** |  |
| **Email** |  | **Designation** |  |
| **Qualification** |  | | |

Head of Department Details



|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Dr. Mohammad Rafi | **Mobile No.** | - |
| **Email** | muhammad.rafi@nu.edu.pk | **Gender** | Male |

Project Details



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Title** |  | | |  | |  |
| **Group Details** | **Member 1 Name:** Shaikh Abdul Rafay    **Member 1 Roll#:** 21K 3051 | | **Member 2 Name:** Rayyan Ahmed    **Member 2 Roll#:** 21K 3079 | | **Member 3 Name:** Syeda Minal Alam    **Member 3 Roll#:** 21K 3072 |  |
|  |  |  | |  | |  |
| **Project Area of** | Deep Learning, Artificial Intelligence, Web | | | | |  |
| **Specialization** |  |  | |  | |  |
|  |  |  | |  | |  |
| **Project Start** | 2nd September 2024 | **Project End Date** | |  | |  |
| **Date** |  |  | |  | |  |
|  |  |  | |  | |  |
| **Project** | Cricket matches, depending on the format, can last anywhere from several hours to five days, making post-match editing a time-consuming process. My Final Year Project (FYP) is focused on automating the generation of cricket highlights using cutting-edge technologies like computer vision and Optical Character Recognition (OCR). The goal is to develop a system that takes cricket broadcast footage as input and automatically extracts key moments—such as boundaries, sixes, and wickets—without the need for manual intervention. This innovation will drastically reduce the workload of video editors and expedite highlight generation for cricket broadcasts.  In addition to highlights generation, the project integrates multiple other features into a comprehensive web-based platform. One key feature is a chatbot powered by Natural Language Processing (NLP), allowing users to interact with the platform by asking questions about the ongoing match in everyday language. For instance, users could inquire, "Who just got out?" or "How many runs are needed to win?" and receive accurate, real-time responses.  Another significant aspect is the inclusion of a generative commentary system. Instead of relying on original match commentary, the platform will use machine-generated commentary to narrate the events in the extracted clips, offering a unique viewing experience.  Lastly, a machine learning-based score prediction model will provide users with insights into possible match outcomes, enhancing their engagement and understanding of the game. All these features are designed to make cricket highlights more accessible, interactive, and engaging for a broad audience. | | | | |  |
| **Summary (less** |  | | | | |  |
| **than 2500** |  | | | | |  |
| **characters)** |  | | | | |  |
|  |  | | | | |  |
|  |  |  | |  | |  |
| **Project** | The primary objective of this project is to address the inefficiencies in the manual process of cricket highlights generation. Traditional methods of extracting key moments from cricket matches require editors to spend hours reviewing footage to identify significant events such as boundaries, sixes, wickets, and other match-defining moments. Given that cricket matches can span from a few hours to several days, this method is both time-consuming and resource-intensive.  Additionally, there is a growing demand for faster content delivery in the digital age, where audiences expect immediate access to key moments, often in real-time or shortly after they occur. This project seeks to meet that demand by using computer vision and OCR technology to automate the process of highlights generation, reducing the time it takes to deliver high-quality sports content.  Another key issue this project addresses is the lack of interactive features in current highlight systems. Viewers often have questions about the ongoing match, but existing platforms do not provide a way to interact with the content in real-time. By integrating a chatbot powered by NLP, the project aims to enhance the viewer experience by allowing users to ask questions such as “Who got out?” or “How many runs are needed to win?” and receive real-time, relevant answers.  The project also tackles the problem of creating engaging content for various fan demographics. Traditional commentary is often fixed and may not cater to different viewer preferences. By implementing generative commentary, the system will offer personalized, AI-driven narration, providing a fresh and customizable viewing experience.  Finally, this project aims to provide predictive insights into cricket matches. Existing highlight systems do not offer any analytical tools for predicting match outcomes. The inclusion of a machine learning-based score predictor addresses this gap by offering fans statistical insights into possible outcomes, further enhancing their engagement with the game. | | | | |  |
| **Objectives (less** |  | | | | |  |
| **than 2500** |  | | | | |  |
| **characters)** |  | | | | |  |
|  |  |  | |  | |  |
| **Literature Review / Background Study** | **Automated Highlights Generation**  The process of automatic cricket highlight generation has gained significant attention due to the growing need to manage vast volumes of sports content. Several researchers have proposed various models incorporating computer vision and machine learning techniques to identify and extract key events from cricket broadcasts.  Javed et al. (2018) introduced a multimodal framework that combined both audio and visual features for summarizing cricket videos [?]. Their approach utilized acoustic local binary pattern (LBP) features to assess the excitement level in the audio stream, such as crowd cheers or commentary. These features were processed by a Support Vector Machine (SVM) classifier to distinguish between excited and non-excited audio frames. Key-video frames were selected based on the excited audio segments, and a decision-tree-based classifier was employed to detect key events such as boundaries, sixes, and wickets. The method achieved an impressive 95.5% accuracy in summarization, proving its robustness in handling diverse video formats and cricket tournaments.  Shukla et al. (2018) proposed an approach combining both event-driven and excitement-based features to generate cricket highlights. Their method relied on detecting events such as wickets, boundaries, sixes, and milestones using event-driven features like Optical Character Recognition (OCR), and analyzing excitement through audio intensity and replays. The system achieved a robust performance, yielding highlights comparable to those generated manually by professionals. Their framework proved efficient in summarizing long-duration cricket matches, showing significant overlap between the generated highlights and those from official broadcasters. [3] Similarly, Agarwal et al. (2019) developed an automatic annotation and highlight generation framework using Convolutional Neural Networks (CNN) and OCR. They segmented cricket match videos into individual ball clips, where the CNN detected the start frame of a ball, and OCR recognized score changes to detect the end frame. Their system effectively generated highlights by filtering out less relevant clips and focusing on important events like boundaries, wickets, and milestones. The results, measured using Intersection over Union (IoU) metrics, indicated high accuracy in comparison to broadcaster-provided highlights. [1]  Gaikwad et al. (2022) proposed a deep learning-based video summarization method for generating cricket highlights [?]. Their approach used Convolutional Neural Networks to analyze pre-processed video frames, extracting features to summarize key events in cricket matches. This method showed improved results over other summarization techniques by effectively reducing video length without sacrificing key content. Their model demonstrated a high level of accuracy, making it suitable for creating short highlight clips of cricket matches. [2]  Finally, Shingrakhia (2023) proposed a deep learning-based method that combines both event-driven and excitement-based features. Event-driven features detect significant moments like boundaries and wickets using Optical Character Recognition (OCR) to track score changes, while excitement-based features leverage audio cues to detect heightened moments of excitement, such as crowd reactions or replays. The system demonstrated strong performance, efficiently summarizing long-duration cricket matches with minimal manual intervention.  Together, these studies highlight the effectiveness of combining event-driven analysis with machine learning techniques, such as CNNs and OCR, to enhance the process of cricket highlight generation, reducing manual effort while maintaining content quality.  **Automated Commentary**  AI-powered automated commentary has advanced significantly, offering substantial benefits over traditional manual commentary. These systems provide real-time, accurate, and engaging narration, enhancing the viewer experience while reducing reliance on human commentators. By leveraging large language models and machine learning, automated commentary ensures consistent and efficient coverage, making it a valuable tool in today’s sports broadcasting domain.  Alec Cook and Oktay Karaku¸s (2024) explores fine-tuning large language models (LLMs) for football commentary, highlighting the Mixed Immediately (MIM) strategy’s promise in managing complex tasks on consumer-grade hardware. They introduce the LLM-Commentator system, which fine-tunes open-source LLMs to generate football commentary on standard hardware. The study evaluates three fine-tuning methods: Layered Mixed Sequentially (MSM), Mixed Immediately (MIM), and a base model (LM). The MIM model achieves a notable 0.91 F1 score, demonstrating efficiency in task management and mitigating catastrophic forgetting. Despite these advancements, challenges in real-time data acquisition and implementation persist, underscoring the MIM model’s potential for future research in real-time generation and fine-tuning of smaller LLMs.[2]  Peter Andrews et al. (2024) introduced the Ai Commentator system, featuring dual AI commentators for both non-interactive and interactive football commentary to enhance viewer engagement. Their review covers the evolution from basic play-by-play to sophisticated color commentary, emphasizing the roles of both commentator types. The Ai Commentator system utilizes computer vision (CV) and natural language processing (NLP) to analyze video sequences and provide real-time commentary, employing models like FootyVision for multiobject tracking, the Hungarian algorithm for tracking, GPT-3.5 for commentary generation, and Azure TTS for nuanced speech. Previous systems such as SCoReS and Chitrakala et al. (2023) have aimed to assist human commentators with recommendation systems. The review notes that embedded visualizations, or emphasizing, enhance viewer understanding, and conflicting commentary is found to be more engaging. Despite limitations like preprocessing delays, lack of strategic understanding, miswording issues, and mixed feedback on visualizations, the Ai Commentator system demonstrated real-time capability of 194.34 FPS, suggesting potential for effective, resource-efficient operation. [1]  Jakub Ko´sciotek (2024) focuses on reducing commentary delays in soccer video games using SVM and NN models, detailing the evolution and challenges of live commentary generation. The literature review addresses foundational works on automated commentary, including issues like repetitive commentary and limited comment scope. It explores event prediction and classification In soccer games, referencing studies that use various techniques to predict game events and outcomes. A significant challenge noted is the delay in commentary due to the overhead of large language models (LLMs) and Text-to-Speech algorithms, with delays averaging 6 seconds. The Google Football Environment is highlighted as a valuable tool for data extraction, with research leveraging this environment to train the GPT-3.5 model. Machine learning models such as SVM and ANN are evaluated, with SVM achieving an F1-score of 79% on balanced data but dropping to 69% on imbalanced data. Limitations like data preparation and model enhancement are noted, with future research focusing on improving prediction performance and commentary realism. [3]  The studies collectively demonstrate significant progress in AI-driven sports commentary. The fine-tuning of large language models (LLMs) shows promising results in managing complex tasks and improving performance on standard hardware. The integration of computer vision (CV) and natural language processing (NLP) enhances real-time commentary capabilities. Additionally, advancements in machine learning models address challenges such as commentary delays and prediction accuracy, highlighting both the potential and the need for continued research in optimizing AI for sports commentary.  **Score Prediction And Stats Analysis**  Cricket score prediction has emerged as a significant area of research, driven by advancements in machine learning and data analytics. Various models have been developed to predict match outcomes, player performance, and team scores, using both traditional machine learning algorithms and big data frameworks.  In a study by Mundhe et al. (2021), a web application was developed for live T20 match prediction. The system used Multivariate Polynomial Regression for predicting the final score and a Random Forest Classifier to predict the match winner. The model relied on real-time data scraping, considering factors like the current score, overs bowled, and wickets lost. While moderately accurate, the study highlighted the challenges of predicting outcomes in T20 cricket due to its volatile nature. Their approach emphasized the need for real-time data integration and improved algorithms to handle the unpredictable aspects of cricket [3].  Hatharasinghe and Poravi (2021) explored the application of data mining and machine learning in cricket, focusing on player performance, match simulation, and team selection. They reviewed several models, including those based on historical cricket data and social media analysis, to predict match outcomes. Despite the potential of these models, they noted limitations in predicting international matches due to insufficient training data, particularly for new players. The study suggested that a combination of approaches, integrating both historical and real-time data, could significantly improve accuracy [4].  Awan et al. (2021) applied a big data approach using the Spark ML framework to predict team scores and match outcomes. Their study demonstrated that Spark ML, with its ability to handle large datasets, outperformed traditional frameworks like Scikit-learn. Using linear regression, the model achieved test accuracy rates of 95%, with improved performance in terms of mean absolute error (MAE) and root mean square error (RMSE). The authors highlighted the scalability of big data approaches in cricket analytics and their potential for application across other sports [1].  Bharadwaj et al. (2024) emphasizes the importance of analyzing cricket player performance using machine learning techniques. The study focuses on various parameters such as player consistency, form, performance against specific opponents, and venue impact. It applies algorithms like Na¨ıve Bayes, Decision Tree, Random Forest, and Support Vector Machine (SVM) to predict batting and bowling outcomes. Among these, Random Forest and Decision Tree were found to be the most effective in capturing player performance dynamics, offering reliable predictions for both batting scores and wickets taken. [2]  Each of these studies contributes to the growing body of work on cricket score prediction, illustrating the evolution from traditional machine learning models to big data frameworks. The integration of real-time data and ensemble methods, such as Random Forest and XGBoost, has led to improved predictive accuracy, but challenges remain, particularly in fast-paced formats like T20 cricket. | | | | |  |
| **Project Implementation Method (less than 2500 characters)** |  **Data Collection**: Gather live cricket match data, including scores, player stats, and match events, from reliable sources via APIs.   **Frontend Development**: Design a user-friendly interface to display real-time scores and match summaries.   **AI Features**:   * Implement **Automatic Highlight Generation** using computer vision and event detection algorithms to capture key moments. * Develop **Automatic Commentary** using natural language generation techniques. * Build a **Score and Win Prediction** model using machine learning to forecast match outcomes.    **Stats Analytics**: Integrate a module for advanced statistical analysis of player and match data.   **Chatbot Integration**: Create a chatbot using NLP to answer user queries related to live match details and history.   **Backend Development**: Set up a server to manage data fetching, storage, and AI model deployment, ensuring real-time updates and user interaction.   **Testing and Deployment**: Conduct rigorous testing to ensure system stability, accuracy, and performance, followed by deploying the platform on a cloud server. | | | | |  |
|  |  | | | | |  |
|  |  | | | | |  |
|  |  | | | | |  |
|  |  | | | | |  |
| **Benefits of the** | The automated cricket highlights generation system will revolutionize the way post-match content is processed and delivered. Currently, the task of editing and compiling key moments from cricket matches is a manual and labor-intensive process, often requiring hours of work to condense matches that can last anywhere from 5 hours to 5 days. By leveraging computer vision and OCR technologies, this project will automate the extraction of significant events like boundaries, sixes, and wickets, thereby dramatically reducing the time and resources needed for highlight generation. This will streamline workflows for broadcasters, content creators, and sports media companies, making the process more efficient.  One major benefit is the democratization of content creation. Smaller platforms or independent content creators who do not have access to large editing teams will be able to produce high-quality cricket highlights automatically, making cricket content more accessible to a broader audience. The generative commentary feature also allows for a personalized viewing experience, moving away from traditional commentary to a tailored, AI-driven narrative.  The integration of a Chabot powered by NLP enhances user interaction by allowing casual viewers to ask match-related questions in real-time. This is particularly valuable in an age where fans expect instant updates and personalized engagement. This feature not only improves user experience but also opens up new possibilities for fan engagement during live broadcasts.  In terms of research contributions, the project demonstrates how AI, particularly computer vision and OCR, can be applied in the field of sports broadcasting to automate tedious tasks, a use case with significant commercial value. The machine learning-based score prediction model introduces an innovative way to provide fans with analytical insights into match outcomes, offering a layer of engagement beyond traditional viewing. This project showcases the potential of AI to revolutionize sports broadcasting, making it more efficient, engaging, and scalable. | | | | |  |
| **Project (less** |  | | |  | |  |
| **than 2500 characters)** |  | | | | |  |
|  |  | | | | |  |
|  |  |
|  |  |  | |  | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Technical** | * Highlights Generation Using OCR and CV * Automated Commentary using LLMs * Live Score Prediction * Comparison and Result analysis | | |  |
| **Details of Final Deliverable (less than 2500 characters)** |  | | |  |
|  |  |  |  |  |
| **Final Deliverable of the Project** | * Website * Robust models. * Result analysis of proposed model. | | |  |
|  |  |  |  |  |
| **Core Industry (Optional)** | Cricket and Entertainment |  |  |  |
|  |  |  |  |  |
| **Other** |  |  |  |  |
| **Industries**  **(Optional)** |  |  |  |  |
|  |  |  |  |  |
| **Core** | MERN, OCR, Computer Vision, LLM, Machine Learning, Deep Learning. |  |  |  |
| **Technology** |  |  |  |  |
|  |  |  |  |  |
| **Other** |  |  |  |  |
| **Technologies (Optional)** |  |  |  |  |
|  |  |  |  |  |
| **Sustainable** |  |  |  |  |
| **Development** |  |  |  |  |
| **Goals**  **(Optional)** |  |  |  |  |
|  |  |  |  |  |
| References     |  |  |  | | --- | --- | --- | | 1. Automated Commentary    * [1] Peter Andrews, Oda Elise Nordberg, Nj˚al Borch, Frode Guribye, and Morten Fjeld. Designing for automated sports commentary systems. In Proceedings of the 2024 ACM International Conference on Interactive Media Experiences, IMX ’24, page 75–93, New York, NY, USA, 2024. Association for Computing Machinery.    * [2] Alec Cook and Oktay Karaku¸s. Llm-commentator: Novel fine-tuning strategies of large language models for automatic commentary generation using football event data. Knowledge-Based Systems, 300:112219, 2024    * [3] Jakub Ko´scio. Enhancing live commentary generation in soccer video games through event prediction with machine learning methods, June 2024. 2. **Highlights generation**    * + [1] Sanchit Agarwal, Nikhil Singh, and Prashant Shambharkar. Automatic annotation of events and highlights generation of cricket match videos. International Journal of Innovative Technology and Exploring Engineering, 8:3878–3881, 09 2019.      + [2] D. Gaikwad, S. Sarap, and Dinesh Dhande. Video summarization using deep learning for cricket highlights generation. Journal of Scientific Research, 14:533–544, 05 2022.      + [3] Pushkar Shukla, Hemant Sadana, Apaar Bansal, Deepak Verma, Carlos Elmadjian, Balasubramanian Raman, and Matthew Turk. Automatic cricket highlight generation using event-driven and excitement-based features. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 1881–18818, 2018 3. **Stat Analysis and Score Prediction**    * [1] Mazhar Javed Awan, Syed Arbaz Haider Gilani, Hamza Ramzan, Haitham Nobanee, Awais Yasin, Azlan Mohd Zain, and Rabia Javed. Cricket match analytics using the big data approach. Electronics, 10(19), 2021.    * [2] F. Bharadwaj, A. Saxena, R. Kumar, R. Kumar, S. Kumar, an   Z. Stevi´c. Player performance predictive analysis in cricket using machine learning. Revue d’Intelligence Artificielle, 38(2):449–457, 2024.   * + [3] Manuka Maduranga Hatharasinghe and Guhanathan Poravi. Data mining and machine learning in cricket match outcome prediction: Missing links. In 2019 IEEE 5th International Conference for Convergence in Technology (I2CT), pages 1–4, 2019.   + [4] Eeshan Mundhe, Ishan Jain, and Sanskar Shah. Live cricket score prediction web application using machine learning. In 2021 International Conference on Smart Generation Computing, Communication and Networking (SMARTGENCON), pages 1–6, 2021. |  |  | | |  |  |  |
| Project Key Milestones | |  |  |  |
|  | |  |  |  |
|  | |  |  |  |
| **Elapsed time in (days or weeks or month or quarter) since start of the project** | | **Milestone** | **Deliverable** |  |
|  |  |  |  |  |
| Month 1 |  | Title Writing and proposal | Proposal Submission |  |
|  |  |  |  |  |
| Month 2 |  | Researching and selecting dataset | Dataset downloading |  |
|  |  |  |  |  |
| Month 3 |  | * Researching on preprocessing techniques * Frontend | Implementing preprocessing  Front end |  |
|  |  |  |  |  |
| Month 4 |  | * Model Training * Backend Development | - |  |
|  |  |  |  |  |
| Month 5 |  | * Model Training and Testing * Backend Development | - |  |
|  |  |  |  |  |
| Month 6 |  | Testing : Model & Backend | - |  |
|  |  |  |  |  |
| Month 7 |  | Model Result Analysis | - |  |
|  |  |  |  |  |
| Month 8 |  | * Documentation * Deployment | * Documentation * Website |  |
|  |  |  |  |  |

Project Equipment Details



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item(s) Name** | **Type** | **No. of Units** | **Per Unit Cost (in Rs)** | **Total (in Rs)** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  | **Total in (Rs)** |  |
|  |  |  |  |  |

