|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Software Design Specifications**  **CrickHigh -**  **Automated Sports Highlights Generation**  **Version: [02.00]**   |  |  | | --- | --- | | Project Code | F24-102 | | Supervisor | Mr. Basit Jasani | | Co Supervisor | - | | Project Team | *21k-3079 Rayyan Ahmed*  *21k-3051 Shaikh Abdul Rafay*  *21k-3072 Syeda Minal Alam* | | Submission Date | 15-05-2025 | |   **[Instructions]**   * *No section of template should be deleted. You can write ‘Not applicable’ if a section is not applicable to your project. But all sections must exist in the final document.* * *All comments/examples mentioned in square brackets ([]) are in the template for explanation purposes and must be replaced / removed in final document.* * *This’ Instruction’ section should also be removed in final document.* * *MS-Word Reviewing feature must be used to get the document reviewed by PMs or supervisors.*       **Document History**  *[Revision history will be maintained to keep a track of changes done by anyone in the document.]*   |  |  |  |  | | --- | --- | --- | --- | | Version | Name of Person | Date | Description of change | | 1.0 | Shaikh Abdul Rafay | 11-12-24 | Document Created | | 2.0 | Syeda Minal Alam | 14-05-25 | Document Updated |     **Distribution List**  *[Following table will contain list of people whom the document will be distributed after every sign-off]*   |  |  | | --- | --- | | **Name** | **Role** | | Mr. Basit Jasani | Supervisor | | - | Co Supervisor | |  |  |     **Document Sign-Off**  *[Following table will contain sign-off details of the document. Once the document is prepared and revised, this should be signed-off by the sign-off authority.*  *Any subsequent changes in the document after the first sign-off should again get a formal sign-off by the authorities.]*   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Version** | **Sign-off Authority** | **Project Role** | **Signature** | **Sign-off Date** | | 1.0 | Supervisor | Supervisor |  |  | | 2.0 | Supervisor | Supervisor |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |

**Document Information**

| **Category** | **Information** |
| --- | --- |
| Customer | - |
| Project | Automated Sports Highlights Generation |
| Document | Software Design Specification |
| Document Version | 2.0 |
| Status | Draft |
| Author(s) | Shaikh Abdul Rafay  Rayyan Ahmed  Minal Alam |
| Approver(s) | Mr. Basit Jasani |
| Issue Date |  |
| Document Location |  |
| Distribution | Advisor  Project Coordinator’s Office (through Advisor) |

**Definition of Terms, Acronyms and Abbreviations**

*[This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly. ]*

| **Term** | **Description** |
| --- | --- |
| CNN | Convolutions Neural Network |
| OCR | Optical Character Recognition |
| AI | Artificial Intelligence |
| Tf | Tensor Flow |
| LLM | Large Language Model |
|  |  |
|  |  |
|  |  |

**Table of Contents**

[**1**](#_heading=h.1v1yuxt"﷟HYPERLINK "bookmark://_heading=h.1v1yuxt) ﷟HYPERLINK "bookmark://\_heading=h.1v1yuxt"**Introduction** **8**

[*1.1*](#_heading=h.4d34og8"﷟HYPERLINK "bookmark://_heading=h.4d34og8) ﷟HYPERLINK "bookmark://\_heading=h.4d34og8"*Purpose of Document* *8*

[*1.2*](#_heading=h.4f1mdlm"﷟HYPERLINK "bookmark://_heading=h.4f1mdlm) ﷟HYPERLINK "bookmark://\_heading=h.4f1mdlm"*Intended Audience* *8*

[*1.3*](#_heading=h.2u6wntf"﷟HYPERLINK "bookmark://_heading=h.2u6wntf) ﷟HYPERLINK "bookmark://\_heading=h.2u6wntf"*Document Convention* *8*

[*1.4*](#_heading=h.19c6y18"﷟HYPERLINK "bookmark://_heading=h.19c6y18) ﷟HYPERLINK "bookmark://\_heading=h.19c6y18"*Project Overview* *8*

[*1.5*](#_heading=h.3tbugp1"﷟HYPERLINK "bookmark://_heading=h.3tbugp1) ﷟HYPERLINK "bookmark://\_heading=h.3tbugp1"*Scope* *8*

[**2**](#_heading=h.28h4qwu"﷟HYPERLINK "bookmark://_heading=h.28h4qwu) ﷟HYPERLINK "bookmark://\_heading=h.28h4qwu"**Design Considerations** **9**

[*2.1*](#_heading=h.nmf14n"﷟HYPERLINK "bookmark://_heading=h.nmf14n) ﷟HYPERLINK "bookmark://\_heading=h.nmf14n"*Assumptions and Dependencies* *9*

[*2.2*](#_heading=h.37m2jsg"﷟HYPERLINK "bookmark://_heading=h.37m2jsg) ﷟HYPERLINK "bookmark://\_heading=h.37m2jsg"*Risks and Volatile Areas* *9*

[**3**](#_heading=h.2jxsxqh"﷟HYPERLINK "bookmark://_heading=h.2jxsxqh) ﷟HYPERLINK "bookmark://\_heading=h.2jxsxqh"**System Architecture** **10**

[*3.1*](#_heading=h.1mrcu09"﷟HYPERLINK "bookmark://_heading=h.1mrcu09) ﷟HYPERLINK "bookmark://\_heading=h.1mrcu09"*System Level Architecture* *10*

[*3.2*](#_heading=h.3j2qqm3"﷟HYPERLINK "bookmark://_heading=h.3j2qqm3) ﷟HYPERLINK "bookmark://\_heading=h.3j2qqm3"*Software Architecture* *10*

[**4**](#_heading=h.46r0co2"﷟HYPERLINK "bookmark://_heading=h.46r0co2) ﷟HYPERLINK "bookmark://\_heading=h.46r0co2"**Design Strategy** **11**

[**5**](#_heading=h.2lwamvv"﷟HYPERLINK "bookmark://_heading=h.2lwamvv) ﷟HYPERLINK "bookmark://\_heading=h.2lwamvv"**Detailed System Design** **12**

[*5.1*](#_heading=h.2xcytpi"﷟HYPERLINK "bookmark://_heading=h.2xcytpi) ﷟HYPERLINK "bookmark://\_heading=h.2xcytpi"*Database Design* *12*

[5.1.1](#_heading=h.1ci93xb"﷟HYPERLINK "bookmark://_heading=h.1ci93xb) ﷟HYPERLINK "bookmark://\_heading=h.1ci93xb"ER Diagram 12

[5.1.2](#_heading=h.3whwml4"﷟HYPERLINK "bookmark://_heading=h.3whwml4) ﷟HYPERLINK "bookmark://\_heading=h.3whwml4"Data Dictionary 12

[5.1.2.1](#_heading=h.2bn6wsx"﷟HYPERLINK "bookmark://_heading=h.2bn6wsx) ﷟HYPERLINK "bookmark://\_heading=h.2bn6wsx"Data 1 12

[5.1.2.2](#_heading=h.qsh70q"﷟HYPERLINK "bookmark://_heading=h.qsh70q) ﷟HYPERLINK "bookmark://\_heading=h.qsh70q"Data 2 12

[5.1.2.3](#_heading=h.3as4poj"﷟HYPERLINK "bookmark://_heading=h.3as4poj) ﷟HYPERLINK "bookmark://\_heading=h.3as4poj"Data 3 12

[5.1.2.3](#_heading=h.3as4poj"﷟HYPERLINK "bookmark://_heading=h.3as4poj) ﷟HYPERLINK "bookmark://\_heading=h.3as4poj"Data 4 12

[5.1.2.3](#_heading=h.3as4poj"﷟HYPERLINK "bookmark://_heading=h.3as4poj) ﷟HYPERLINK "bookmark://\_heading=h.3as4poj"Data 5 12

[5.1.2.3](#_heading=h.3as4poj"﷟HYPERLINK "bookmark://_heading=h.3as4poj) ﷟HYPERLINK "bookmark://\_heading=h.3as4poj"Data 6 12

[5.1.2.3](#_heading=h.3as4poj"﷟HYPERLINK "bookmark://_heading=h.3as4poj) ﷟HYPERLINK "bookmark://\_heading=h.3as4poj"Data 7 12

[5.1.2.3](#_heading=h.3as4poj"﷟HYPERLINK "bookmark://_heading=h.3as4poj) ﷟HYPERLINK "bookmark://\_heading=h.3as4poj"Data 8 12

[5.1.2.3](#_heading=h.3as4poj"﷟HYPERLINK "bookmark://_heading=h.3as4poj) ﷟HYPERLINK "bookmark://\_heading=h.3as4poj"Data 9 12

[*5.2*](#_heading=h.1pxezwc"﷟HYPERLINK "bookmark://_heading=h.1pxezwc) ﷟HYPERLINK "bookmark://\_heading=h.1pxezwc"*Application Design* *14*

[5.2.1](#_heading=h.49x2ik5"﷟HYPERLINK "bookmark://_heading=h.49x2ik5) ﷟HYPERLINK "bookmark://\_heading=h.49x2ik5"Sequence Diagram 14

[5.2.1.1](#_heading=h.2p2csry"﷟HYPERLINK "bookmark://_heading=h.2p2csry) ﷟HYPERLINK "bookmark://\_heading=h.2p2csry"<Sequence Diagram 1> 14

[5.2.1.2](#_heading=h.147n2zr"﷟HYPERLINK "bookmark://_heading=h.147n2zr) ﷟HYPERLINK "bookmark://\_heading=h.147n2zr"<Sequence Diagram 2> 14

[5.2.1.3](#_heading=h.3o7alnk"﷟HYPERLINK "bookmark://_heading=h.3o7alnk) ﷟HYPERLINK "bookmark://\_heading=h.3o7alnk"<Sequence Diagram n> 14

[5.2.2](#_heading=h.23ckvvd"﷟HYPERLINK "bookmark://_heading=h.23ckvvd) ﷟HYPERLINK "bookmark://\_heading=h.23ckvvd"State Diagram 14

[5.2.2.1](#_heading=h.ihv636"﷟HYPERLINK "bookmark://_heading=h.ihv636) ﷟HYPERLINK "bookmark://\_heading=h.ihv636"<State Diagram 1> 14

[5.2.2.2](#_heading=h.32hioqz"﷟HYPERLINK "bookmark://_heading=h.32hioqz) ﷟HYPERLINK "bookmark://\_heading=h.32hioqz"<State Diagram 2> 14

[5.2.2.3](#_heading=h.1hmsyys"﷟HYPERLINK "bookmark://_heading=h.1hmsyys) ﷟HYPERLINK "bookmark://\_heading=h.1hmsyys"<State Diagram n> 14

[**6**](#_heading=h.2grqrue"﷟HYPERLINK "bookmark://_heading=h.2grqrue) ﷟HYPERLINK "bookmark://\_heading=h.2grqrue"**References** **15**

[**7**](#_heading=h.3fwokq0"﷟HYPERLINK "bookmark://_heading=h.3fwokq0) ﷟HYPERLINK "bookmark://\_heading=h.3fwokq0"**Appendices** **16**

# Introduction

## Purpose of Document

The purpose of this Software Design Specification (SDS) document is to provide a comprehensive and detailed description of the design architecture for the **CrickHigh** system. It outlines the system components, data flow, architecture, interfaces, and design choices that support the implementation of key functionalities such as automated highlight generation, win prediction, and AI-based cricket insights.

* Guide the development team in implementing the system according to a well-defined design.
* Ensure technical clarity and maintain consistency across the frontend, backend, and machine learning components.
* Document architectural decisions, design patterns, and system interactions for future maintenance and scaling.
* Bridge the gap between requirements and implementation, ensuring alignment with the functional and non-functional goals stated in the SRS.
* Provide traceability for design decisions throughout the development lifecycle.

## Intended Audience

* Fast NU
* Jury
* Supervisor (Mr. Basit Jasani)
* Students of Fast NU
* Our Team(Designer, Developer, Tester)
* Potential Users of this product

## Document Convention

* Font Family = Arial
* Font Size = 12 for headings, 10 for the rest of the content

## Project Overview

**CrickHigh** is designed to automate the generation of cricket match highlights and perform match win prediction using advanced machine learning and computer vision techniques. Its primary goal is to enhance the cricket-viewing experience by detecting significant moments and predicting outcomes with high accuracy and efficiency.

**Included Functionalities:**

* Detection of cricket delivery starts using a CNN model.
* OCR-based detection of score changes and event-worthy moments.
* Timestamp-based clipping of cricket highlights.
* Win prediction using historical and real-time match data.
* LLM with Gemini API and Local knowledge base.
* Integration of AI models to enhance accuracy and performance.

**Excluded Functionalities**:

* While full-scale frontend development was not a primary focus, a basic user interface has been implemented to allow users to upload video files for highlight generation and to interact with an AI assistant chatbot for match-related queries. These frontend components are functional but kept minimal, as the core emphasis of the project was on backend intelligence and model integration.

## Scope

**Included Functionalities:**

* Detection of cricket delivery starts using a CNN model.
* OCR-based detection of score changes and event-worthy moments.
* Timestamp-based clipping of cricket highlights.
* Win prediction using historical and real-time match data.
* LLM with Gemini API and Local knowledge base
* Integration of AI models to enhance accuracy and performance.

**Excluded Functionalities**:

* While full-scale frontend development was not a primary focus, a basic user interface has been implemented to allow users to upload video files for highlight generation and to interact with an AI assistant chatbot for match-related queries. These frontend components are functional but kept minimal, as the core emphasis of the project was on backend intelligence and model integration.

# Design Considerations

*[This section describes many of the issues which need to be addressed or resolved before attempting to devise a complete design solution. In other words, this section is used to formally set the groundwork for the system design.]*

## Assumptions and Dependencies

* The system assumes the **availability of consistent and high-quality cricket match video footage** to ensure accurate detection and highlight generation.
* Both the **OCR and CNN models rely on pre-trained weights** to deliver optimal performance in detecting deliveries and score changes.
* The system depends on **machine learning frameworks** such as **PyTorch** and **scikit-learn**, as well as **external APIs** for accessing historical match data and integrating real-time information.

## Risks and Volatile Areas

* Variations in video formats or quality may affect the accuracy of delivery detection and OCR-based score extraction.
* Model performance may fluctuate depending on the quality, quantity, and diversity of the training dataset.
* Real-time data integration for win prediction carries potential risks of latency, data inconsistency, or temporary unavailability.

# 3 System Architecture

## System Level Architecture

The architecture decomposes the system into three main components:

* **Video Processing Module**: Responsible for frame extraction and processing through CNN and OCR models.
* **Highlight Clipping Module**: Generates video clips based on the extracted timestamps.
* **Win Prediction Module**: Utilizes historical and real-time data for predictive analytics.

These components interact via defined interfaces to ensure modularity and scalability.

External dependencies include video storage systems and APIs for historical data retrieval.

Error handling strategies are embedded in each module to ensure system robustness.

## Software Architecture

The software architecture includes:

* **Presentation Layer**: A simple frontend has been developed to allow users to upload cricket match videos for highlight generation and to interact with an AI assistant chatbot for match-related queries.
* **Middle Tier**: Handles core logic, focusing on CNN and OCR integration for detecting delivery starts and score changes, as well as timestamp analysis for highlights and win prediction algorithms.
* **Data Access Layer**: Manages storage and retrieval of results and video data.
* **Interaction between layers**:

The Presentation Layer interacts with the Middle Tier to process user input. The Middle Tier processes data and interacts with the Data Access Layer for storage and retrieval operations.

# Design Strategy

The design strategy for the highlights generation and win predictor system is structured around several key principles, including modularity, scalability, performance optimization, and robustness. Below is a breakdown of the design strategies employed to ensure a successful system:

* **Future System Extension or Enhancement**  
  The system is designed to be easily extendable. This is achieved through the following mechanisms:
  + **Modular architecture**: Each module (video processing, highlight clipping, win prediction, AI assistant) is designed as an independent entity with clearly defined interfaces. This ensures that any future enhancements, such as the addition of more advanced models or new prediction features, can be incorporated without affecting the core system.
  + **Model updates**: Since the system relies heavily on machine learning models (CNN for delivery detection, OCR for score changes, LLM model and Knowledge Base for AI assistant etc.), future updates to models can be implemented without major system overhauls. This modular design allows new models or updated versions of existing models to be swapped in and out seamlessly.
* **System Reuse**:To maximize reuse, several strategies are implemented:
  + **Reusable libraries and frameworks**: The system leverages widely adopted machine learning frameworks like PyTorch and scikit-learn, which are both flexible and reusable for similar tasks in future projects.
  + **Open-source tools**: Where possible, open-source libraries and tools are used to avoid reinventing the wheel. These include tools for video processing, deep learning model training, and data integration.
  + **Abstracted interfaces**: Each component of the system interacts with other components through well-defined interfaces, making it easy to reuse individual modules for other purposes, such as using the video processing module for other types of sports analysis or leveraging the win prediction module for other domains.
* **Data Management (Storage, Distribution, Persistence)**  
  Data is a critical part of the system, and its management is designed for efficiency and scalability:
  + **Storage**: Video data, model weights, and prediction results are stored in a centralized repository (e.g., cloud storage), ensuring that all modules can access and manipulate the data as needed.
  + **Persistence**: For the purposes of ongoing research and system development, experiment results, including prediction data, timestamps for highlights, and processed video segments, are stored in structured formats (e.g., JSON or CSV), making them easy to query and analyze.
  + **Data distribution**: Data (especially video footage and historical match data) is distributed to relevant system components through well-defined APIs, ensuring that each module has the necessary data to perform its function.
* **Concurrency and Synchronization**  
  The system requires handling multiple tasks concurrently, particularly when processing video and performing win prediction:
  + **Asynchronous processing**: Video processing and prediction tasks run asynchronously to ensure that the system can handle multiple inputs (video streams and real-time match data) simultaneously. This allows video frame processing to happen in parallel with win prediction tasks, improving overall system efficiency.
  + **Synchronization**: Timestamps for delivery start and score change events need to be synchronized to ensure that highlights are accurately clipped. This is achieved through the careful management of event time sequences, with timestamps being recorded and processed in a linear order to maintain consistency across the system.

# Detailed System Design

## Database Design

### ER Diagram

Not Applicable

### Data Dictionary

Not Applicable

# Reference

1. **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
2. **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.

# Appendices

*[Include supporting detail that would be too distracting to include in the main body of the document.]*