ROBERT GORDON UNIVERSITY ABERDEEN TEF Gold

INFORMATICS INSTITUTE OF TECHNOLOGY

BSc (Hons) Artificial Intelligence And Data Science
Level 04

CM 4605
Individual Research Project

CricXpert: A Hybrid Approach Combining Facial and Spatio-Temporal Gait Analysis for Enhanced Player Recognition with LLM-Based Statistic Generation

Evaluation Plan

NADUN SHAMIKA SENARATHNE
IIT ID: 20210488
RGU ID: 2117538

Supervised by: Mr. PRASAN YAPA

Table of Contents

1. Introduction

The evaluation plan section is essential for providing a detailed overview of the techniques, strategies, and procedures followed to assess the research outcomes. It ensures that the study's findings can be validated for reliability and robustness. This plan aims to achieve two key objectives: evaluating the effectiveness of cricket player recognition using a hybrid computer vision approach and validating the generation of player-specific statistics using a large language model (LLM).

This section outlines the processes required to assess the quality of the data acquisition, preprocessing, and analysis, followed by a comprehensive evaluation of the models and methodologies utilized. The goal is to offer a clear explanation for validating how the hybrid face recognition and gait analysis models were combined, as well as how powerful machine learning classifiers were employed to achieve high accuracy. This plan also discusses the data sources, assessment criteria, and validation procedures to verify the reliability and generalisability of the results.

2. Data Collection Methods

The data utilized in this study were carefully selected to create an efficient hybrid model for identifying cricket players. The data is divided into three types: face data, gait data, and player statistics. Each dataset improves distinct aspects of the player recognition process, including facial recognition, gait analysis, and statistical retrieval. Data was collected using both customized and publicly available sources.

- **Primary Data**: Facial images and gait patterns of cricket players were extracted from film footage using direct observations and customized data gathering.

- **Secondary Data**: This data was obtained from web sources such as ESPN Cricinfo for player statistics and Google Images for face data. Academic literature and online cricket broadcasts provided raw video clips for gait analysis.

2.1 Data Sources

- **Player Facial Data**: Images were extracted from cricket match video footage sourced from sources like Google Images. These photographs were chosen to provide a variety of lighting and angle settings, replicating real-world match scenarios. They were used to develop and test the face recognition model.

- **Player Gait Data**: This collection includes video snippets of players walking or running during matches. These segments utilize spatio-temporal properties to record unique movement patterns. The video samples were obtained from online cricket broadcasts,YouTube and ICC assuring a variety of match circumstances, such as camera angles and lighting.

- **Player Statistics Data**: This dataset includes statistical information for creating player-specific statistics in an LLM. ESPN Cricinfo provided the statistics, including performance metrics and match records. This dataset was used to create player statistics using natural language queries.

3. Data Preprocessing

To ensure that the data was appropriate for model training, a number of preprocessing steps were performed on each dataset. These stages sought to clean, normalize, and extract significant characteristics from the data in order to improve the hybrid player recognition model's accuracy and efficiency.

3.1 Data Cleaning

- **Player Facial Data**: Images from video clips and Google Images may contain background noise and not related things. Cropping techniques were utilized to isolate the player's face, and low-quality photos (such as significant blurring or occlusion) were removed. YOLO was first employed for face detection, however owing to performance concerns, MTCNN was eventually selected for its greater accuracy in facial area detection.

- **Player Gait Data**: Raw video clips showed somewhat obscured or out of focus players. These frames were removed to ensure that only the best segments were utilized for gait analysis. YOLOv3 was used to find the biggest vertical bounding box, successfully recognising players while avoiding background noise during walking or sprinting sequences.

- **Player Statistics Data**: To ensure data quality, inconsistent or incomplete player statistics were removed. Data entries that included missing values were removed also.

3.2 Data Transformation

- **Normalization**: Continuous characteristics were normalized using min-max scaling to guarantee consistent scale. This includes information like joint angles, velocities, and face embeddings.

- **Feature Extraction**:

    - **Player Facial Data**: Facial embeddings of players have been obtained using Google's FaceNet model, which converts images into 128-dimensional vectors. The identified faces were retrieved using MTCNN and sent to FaceNet for embedding creation.

- **Player Gait Data**: The hybrid model used player gait data with both spatial and temporal components. CLAHE (Contrast Limited Adaptive Histogram Equalisation) was used to improve picture quality before ResNet50 was used to extract features. After recognising joints with Google's Mediapipe library, spatio-temporal characteristics such as step length, velocity, joint angles, joint acceleration, angular velocity, and hip displacement were created for the temporal model. YOLOv3 was utilized for initial player detection, followed by Mediapipe for joint detection.

- **OCR Text Detection**: The EAST text detection model was utilized to detect player names and numbers on jerseys. These text sections were then cropped and sent to Tesseract OCR for text recognition.

3.3 Feature Selection

- **Facial Data**: Reduced dimensionality and improved model performance by excluding low variance features from facial data. To address the overfitting reported in the early stages of SVC classifiers, feature selection was performed via cross-validation and hyperparameter adjustment.

- **Gait Data**: Correlation analysis was used to remove duplicate characteristics from the gait data and train the temporal model with only the most important information. The implementation of Mediapipe enabled accurate joint detection and feature engineering, such as step length, joint angles, hip displacement, and other created characteristics.

4. Model Selection

The necessity for a robust and accurate system capable of recognising cricket players using many modalities, such as facial characteristics, gait analysis, and textual information on jerseys, drove the selection of models and algorithms for this work. The models were chosen after weighing various choices and doing significant experimentation to discover the best successful strategy.

4.1 Facial Recognition Model

- **Initial Approach**: The model was developed using YOLO for face detection and FaceNet and Vision Transformer (ViT) for prediction. However, this combination produced unsatisfactory results due to accuracy, overfitting concerns, and computational expense.

- **Final Model**: The final model used MTCNN (Multi-Task Cascaded Convolutional Neural Networks) for face identification and Google FaceNet for feature extraction. MTCNN was superior at facial detection, but FaceNet created high-quality facial

embeddings. Initially, the retrieved characteristics were categorized by an SVM. However, owing to overfitting, a stacking model based on SVM and KNN was used, which dramatically increased accuracy while reducing overfitting difficulties.

4.2 Hybrid Spatio-Temporal Model

- **Spatial Model**: The hybrid model's spatial component used YOLOv3 to detect players and reduce background noise using the biggest vertical bounding box. CLAHE (Contrast Limited Adaptive Histogram Equalisation) was used to improve image quality. The first objective was to categorize players using deep learning models, which prompted experiments with several architectures such as DenseNet, EfficientNet, Inception, MobileNet, VGG16, Xception, and ResNet50. However, overfitting remained despite early pausing and parameter adjustment. This led to the hypothesis that the overfitting was caused by the classification layers in these architectures. To solve this, a fusion strategy was taken: ResNet50 was utilized only for feature extraction, while SVM and KNN were employed for classification. The extracted features were classified using a stacking ensemble of SVM, KNN, and a final Logistic Regression layer. Hyperparameter tuning was done to all components, which resulted in effective overfitting mitigation and better model performance. ViT models were also investigated, but their computing requirements rendered them unsuitable for our investigation.

- **Temporal Model**: The temporal model analyzed gait patterns. Google's Mediapipe was used to detect player joints, from which temporal features such as step length, joint angles, velocity, joint acceleration, angular velocity, and hip displacement were calculated. These features were inputted into a GRU (Gated Recurrent Unit) model for temporal analysis. The GRU was chosen over the LSTM since it was less complicated and performed better on this task.

4.3 Text Detection and Recognition

An OCR-based technique was used to identify players by their jersey number or name. Initially, Tesseract was utilized, but it did not produce adequate results when detecting text in these dynamic environments. As a result, the EAST (Efficient and Accurate Scene Text Detector) model was used for text detection, followed by Tesseract OCR for text recognition, which greatly improved detection accuracy.

4.4 Ensemble Approach

After extracting features with ResNet50, multiple machine learning classifiers were tested to determine the best technique. The classifiers examined were SVC, KNN, Random Forest, Gradient Boost Machine, and Decision Tree. SVC and KNN demonstrated the best performance, however overfitting continued even after cross-validation and hyperparameter adjustment. To address these concerns further, ensemble approaches such as the Voting Classifier and Decision Level Fusion were investigated. Finally, the Stacking Ensemble method proved the most successful. Overfitting was significantly decreased by merging SVC,

KNN, and a final Logistic Regression layer inside a stacking ensemble and tweaking their parameters, resulting in enhanced model performance and generalization. This technique improved the overall accuracy and dependability of player labeling.

4.5 Player Statistic Generation using LLM

To create player-specific statistics based on user queries, a large language model (LLM) was employed to convert natural language questions into SQL queries that retrieved relevant data from the cricket database. The LLM-based strategy entailed moving from open-source models such Llama 13B, Grok, and Gemini to the OpenAI GPT-4o model for best performance.

SQL queries were generated using prompt engineering and structured parsing approaches. LangChain and Pydantic were used to handle the LLM prompts, assuring proper SQL creation, and the MySQL database was used to retrieve the data. Several difficulties were encountered, including model hallucinations and improper SQL syntax. These concerns were addressed by assigning the model a persona and employing chain-of-thought prompts to facilitate logical inquiry construction.

The workflow included:

1. **Prompt Engineering**: A structured prompt was developed to provide the LLM with extensive explanations of the database schema and interactions between tables, including batting, bowling, and fielding. The prompt provided detailed advice for creating MySQL-compatible queries and ensuring proper syntax.

2. **SQL Query Generation**: The LLM used natural language input to produce SQL queries. Pydantic was used to parse the query output, and OutputFixingParser was used to fix any discrepancies.

3. **Database Interaction**: The generated SQL queries were executed against a MySQL database to get player statistics, including runs, wickets, and match results.

4. **Model Performance Evaluation**: The OpenAI GPT-4o model outperformed other baseline models in terms of SQL query accuracy and response time. The baseline performance was compared to open-source models such as Llama 13B, demonstrating the benefits of employing advanced proprietary models for complicated, domain-specific problems.

Overall, this LLM-based method efficiently converted user questions into SQL queries, resulting in useful player statistics. Model hallucinations were minimized by prompt engineering and chain-of-thought reasoning, resulting in accurate query formulation and efficient database interaction.

5. Experimental Setup

The experimental setting for this work was designed to assess the performance of the suggested hybrid model for cricket player detection, which combines face recognition, spatio-temporal analysis, and player-specific statistics creation. This section describes the experimental settings, datasets, and processes used to assure reliable model training and assessment.

5.1 System Configuration

The tests were carried out using an Apple MacBook Pro with an M1 processor and 8GB of RAM, which was adequate to handle the computational demands of training and testing both deep learning and machine learning models.

The software environment included:

- Python 3.9 as the primary programming language.
- Deep Learning Frameworks: TensorFlow for model development.
- Libraries: OpenCV and CLAHE for video and image preprocessing, Matplotlib for visualization, Scikit-Learn and Joblib for machine learning model development and persistence.
- NumPy for numerical computations, and Pydantic for data validation.
- Pytesseract for OCR text recognition, MTCNN for face detection and Google's Medipipe for player joint detection.
- LangChain for integrating large language models (LLMs) to generate SQL queries.

5.2 Model Integration and Testing

Following individual training, the face recognition, spatiotemporal, and LLM components were combined into a single pipeline for player recognition and statistics production.

The testing phase involved:

- **Sequential Recognition**: New video clips are fed into the system to test the sequential identification process, which includes OCR text detection, facial recognition, spatial and temporal analysis.

- **Player-Specific Statistics**: Generating player-specific statistics using LLM-processed natural language queries, checked against manually collected information to ensure correctness.

5.3 Experimental Procedure

The experimental procedure involved the following stages:

1. **Data Collection and Preprocessing**: Collected, cleaned, and transformed data to suit face recognition, gait analysis, and LLM standards.

2. **Independent Component Training**: Each model component was trained independently, with hyperparameter adjustment to optimize performance.

3. **Model Integration**: The models were incorporated into a system that analyzed a video sequentially to recognise players and provide statistics.

4. **Performance Evaluation**: Real match video footage and user enqueries were used to assess the system's performance. The system's efficacy was evaluated using specific assessment measures for each component (eg. accuracy, F1-score, response time).

The experimental setting ensured that the system was evaluated under dynamic match conditions, taking into consideration varied illumination, player locations, and actions, verifying the hybrid approach's robustness and generalisability.

6. Evaluation Metrics

Several evaluation metrics were used to thoroughly analyze the hybrid model's performance for cricket player recognition and player-specific statistics creation. These metrics were chosen to match the specific needs of each system component, such as facial recognition, spatio-temporal analysis, and SQL query production using a large language model (LLM). The evaluation aimed to establish the system's dependability, accuracy, and generalisability under dynamic match situations.

6.1 Facial Recognition Metrics

The performance of the face recognition model was assessed using the following metrics:

1. **Accuracy**: The accuracy of a facial recognition model is determined by the proportion of correctly detected examples relative to total instances

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

2. **Precision**: The algorithm's success rate in detecting player faces compared to the total number of faces recognised, minimizing false positives.

$$\text{Precision} = \frac{TP}{TP + FP}$$

3. **Sensitivity (Recall):** The ratio of properly detected player faces in a dataset, indicating the model's capacity to detect all relevant occurrences.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Where:

True Positives (TP): Correctly identified instances
True Negatives (TN): Correctly rejected instances
False Positives (FP): Incorrectly identified instances
False Negatives (FN): Missed instances

6.2 Spatio-Temporal Model Metrics

The spatiotemporal component included spatial analysis with deep learning and temporal analysis with gait characteristics. The assessment measures for these components were:

- **Spatial Model**:
    1. **Accuracy**: Evaluated player classification accuracy using ResNet50 spatial features.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

    2. **Precision and Recall**: Machine learning classifiers (SVM, KNN, Logistic Regression) were evaluated for their ability to discriminate players based on spatial information using precision and recall measures.

$$\text{Precision} = \frac{TP}{TP + FP} \qquad \text{Recall} = \frac{TP}{TP + FN}$$

    3. **Confusion Matrix**: Used to measures the distribution of true positives, false positives, true negatives, and false negatives for player recognition.

- **Temporal Model**:
    1. **Sequence Accuracy**: Evaluated the GRU model's accuracy in classifying temporal sequences of posture landmarks and recognising player movement patterns across time.

$$\text{Sequence Accuracy} = \frac{\text{Number of Correctly Predicted Sequences}}{\text{Total Number of Sequences}}$$

    2. **Model Accuracy and Model Loss Plot**: The Model Accuracy and Loss Plot visualizes training and validation accuracy and loss, identifying overfitting or underfitting during training.

        Model Loss: Loss is calculated during training using a specific loss function, Cross-Entropy Loss:

$$\text{Loss (Cross-Entropy)} = -\sum_{i=1}^{N} y_i \log(p_i)$$

Where:

- y_i : Actual label (1 for correct class, 0 for others)
- p_i : Predicted probability of class i

6.3 LLM-Based Player Statistics Generation Metrics

The LLM used for generating player-specific statistics was evaluated using the following metrics:

1. **SQL Query Accuracy**: The percentage of created SQL queries that yielded correct results when performed against the database. This indicates that the LLM accurately translated natural language enqueries into SQL instructions.

$$\text{SQL Query Accuracy} = \frac{\text{Number of Correct SQL Queries}}{\text{Total Number of Generated SQL Queries}}$$

2. **Response Time**: The time it takes for the LLM to produce a SQL query and provide results. This is used to assess efficiency in real-time applications.

$$\text{Average Response Time} = \frac{\sum_{i=1}^{N} t_i}{N}$$

Where:

- t_i : Response time for each query
- N : Total number of queries

3. **Correctness of Retrieved Information**: Verified the accuracy of obtained player statistics by comparing LLM-generated responses with manually validated data.

$$\text{Correctness} = \frac{\text{Number of Correct Results}}{\text{Total Results Retrieved}}$$

6.4 Integrated System Evaluation

The overall performance of the hybrid player recognition system was tested using the following metrics:

1. **Overall System Accuracy**: the percentage of correct player identifications and statistics generated across trials.

$$\text{Overall System Accuracy} = \frac{\text{Total Correct Identifications}}{\text{Total Identifications}}$$

2. **Execution Time**: Measured the time taken for the system to process a new video clip, perform recognition, and generate statistics, ensuring that the system's performance was suitable for near-real-time applications in dynamic environments like cricket matches.

$$\text{Average Execution Time} = \frac{\sum_{i=1}^{N} t_i}{N}$$

Where:

- t_i : Time taken for each trial
- N : Total number of trials

These evaluation criteria gave us a thorough insight of each hybrid system component's strengths and drawbacks. The suggested solution's overall efficacy, dependability, and scalability were carefully evaluated by analyzing each measure separately and as part of an integrated system.

6.5 Novel Dataset Validation

To validate the reliability and robustness of the created cricket player recognition dataset, an LLM-based validation process was carried out. This validation technique compensates for the absence of publicly available assessment benchmarks designed for dynamic match situations.

Validation Objectives

The dataset was validated to:

1. Assess its robustness against challenging conditions such as:
    - **Diverse Angles**: Simulating real-world field conditions with images captured from multiple camera perspectives.
    - **Lighting Variations**: Evaluating under varying daylight, dusk, and artificial lighting conditions.
    - **Occlusions and Overlaps**: Testing recognition when players are partially obscured by other objects or players.
2. Ensure that the dataset supports accurate and consistent player identification across diverse match scenarios.

Validation Methodology

- **Model Used**: A free-to-use Large Language Model (LLM) with multimodal capabilities, specifically the CLIP model, was utilized to evaluate the dataset.

- **Evaluation Process**:
  - A subset of 30 annotated images per player (180 total) was selected, covering diverse conditions.
  - Each image was paired with corresponding player labels, and the LLM was tasked with classifying the images.
  - Metrics such as accuracy, confusion matrix, and prediction confidence scores were computed to evaluate the dataset's quality.

- **Preprocessing**: Images were preprocessed using Contrast Limited Adaptive Histogram Equalization (CLAHE) to improve clarity and feature extraction.

7. Validation Strategy

The validation approach used in this work was intended to validate the hybrid model's robustness, reliability, and generalisability for cricket player recognition and player-specific statistic creation. Validation took place at both the component and system levels, allowing for a thorough review of each model and the combined pipeline.

Cross-Validation

- **K-Fold Cross-Validation**: The facial recognition and spatio-temporal models were evaluated using a 5-fold cross-validation technique (K-fold). This strategy divided the dataset into five subgroups, iteratively training the model on four folds and verifying on the remaining fold. This ensured a diversified training and validation dataset, decreasing bias and overfitting.

- **Temporal Model Validation**: Cross-validation of gait sequences ensured the GRU model's generalisability. Temporal characteristics were verified against various segments of player motion data to assess the model's performance on previously encountered sequences.

Training and Validation Split

- **Training and Validation Split**: The datasets for facial recognition, spatio-temporal analysis, and player statistics creation were separated into training, validation, and testing sets. Specifically, 80% of the data was utilized for training, 10% for validation, and 10% for testing. The validation set was utilized to modify hyperparameters and reduce overfitting, whilst the test set gave an unbiased assessment of the final model.

Early Stopping and Ensemble Validation

- **Early Stopping**: Deep learning models were trained using early stopping to prevent overfitting. Training was paused when the model's validation loss stopped improving after a certain number of epochs, ensuring that the model did not learn noise from the training dataset.

- **Ensemble Model Validation**: Validated the stacking ensemble for facial recognition and spatial analysis using separate base model evaluations. This method assisted in determining the ensemble's efficacy in terms of bias and variance reduction.

End-to-End System Validation

- **Integrated Pipeline Testing**: New footage was used to evaluate the hybrid recognition pipeline's overall performance. Validation consisted of progressive testing, beginning with OCR text detection and progressing to facial recognition, spatial analysis, and temporal gait analysis. The output from each stage was tested to ensure that it accurately contributed to the final player recognition and statistics creation.

- **Manual Verification**: Player recognition and statistics generation results were manually verified against known records to ensure the accuracy of the integrated system. This step was particularly important for validating the LLM's SQL query responses against actual player data.

The validation process ensured that each component and the entire system were thoroughly tested to minimize errors, enhance accuracy, and verify that the models could generalize to real-world cricket match circumstances. This extensive validation technique raised confidence in the proposed hybrid system's dependability and resilience.