SoccerVision: Player Detection and Team Classification using Computer Vision Techniques

Anonymous CVPR submission

Paper ID *****

Abstract

This paper presents SoccerVision, a computer vision system for detecting soccer players and classifying their team affiliations in match imagery. Instead of relying on complex deep learning models, we implement classical computer vision techniques including Canny edge detection, and custom k-means clustering for color-based team classification. Our approach first identifies player regions using edge detection and field segmentation, then classifies players into teams by analyzing jersey colors in HSV color space. The system demonstrates that effective player detection and team classification can be achieved with fundamental computer vision algorithms, without requiring external libraries or pre-trained models. Experimental results show the system can successfully detect players and classify them into appropriate teams across various soccer match scenarios.

1. Introduction

Soccer (football) analytics has become an essential aspect of modern sports, with teams and broadcasters seeking automated ways to extract player information from match footage. While many commercial systems employ sophisticated deep learning approaches requiring extensive training data and computational resources, there remains value in exploring what can be achieved with fundamental computer vision techniques.

This project, SoccerVision, addresses the challenge of detecting soccer players on the field and classifying them into their respective teams using only classical computer vision algorithms. The system demonstrates how a combination of edge detection, color space manipulation, and clustering can produce effective results without relying on neural networks or external machine learning libraries.

The primary challenges in this domain include:

- Distinguishing players from the field and background elements
- Handling variable lighting conditions across different stadiums
- Differentiating between teams based solely on jersey colors
- Managing occlusions when players overlap
- Processing images with sufficient speed for potential real-time applications

By addressing these challenges with classical computer vision approaches, this work serves as both a practical system and an easy accessible resource for collegite teams to analyze soccer matches.

2. Related Work

The detection and tracking of sports players has been an active research area for decades. Early approaches relied primarily on background subtraction techniques, assuming a relatively static background (the field) against which players could be detected. These methods often struggled with changing lighting conditions and camera movements.

With the rise of deep learning, many recent approaches have shifted to using convolutional neural networks for player detection. These methods typically employ object detection frameworks such as Faster R-CNN, YOLO, or SSD to identify players in video frames with high accuracy. While effective, these approaches require substantial training data and computational resources.

For team classification, color-based methods remain common even in modern systems. Previous work has shown success using color histograms in HSV space to differentiate teams, while more recent approaches have employed clustering techniques on color features extracted from player regions. Other work has used machine learning models to identify teams based on features extracted from player regions [1].

Our approach differs from recent work by intentionally avoiding deep learning methods, instead focusing on demonstrating what can be achieved with classical computer vision algorithms. This makes our system more accessible for educational purposes and deployable in environments with limited computational resources. Unlike some prior work that relies on sophisticated tracking algorithms across video frames, our system operates on individual images, making it applicable to both video and still image analysis.

3. Methodology

Our approach consists of two main components: player detection and team classification. Each component utilizes classical computer vision techniques implemented without relying on external machine learning libraries.

3.1. Player Detection

The player detection pipeline employs a series of image processing steps to identify player regions within a soccer match image:

1. **Grayscale Conversion:** The input RGB image is converted to grayscale to simplify subsequent processing.
2. **Gaussian Blur:** A Gaussian blur with a 5×5 kernel is applied to reduce noise while preserving structural information necessary for edge detection.
3. **Gradient Calculation:** Image gradients are computed using Sobel operators in both horizontal and vertical directions to identify regions with intensity changes.
4. **Canny Edge Detection:** The Canny algorithm is applied to find edges in the image, using dynamic thresholding based on the input image characteristics.
5. **Field Segmentation:** In parallel, the green playing field is identified by applying HSV color thresholding, creating a binary mask where the field is white and non-field areas are black.
6. **Non-Field Masking:** The field mask is inverted to focus on regions potentially containing players.
7. **Edge-Mask Combination:** The Canny edge detection results are combined with the non-field mask using a bitwise AND operation, effectively filtering out edges that occur on the field (such as field lines).
8. **Player Region Identification:** Connected edge regions are identified and filtered based on size constraints to locate potential player regions.
9. **Bounding Box Creation:** Rectangular boxes are generated around the identified player regions.
10. **Non-Maximum Suppression:** Overlapping detections are eliminated to prevent multiple detections of the same player.

This approach leverages the observation that players typically create distinct edges against the more uniform field background, and by filtering based on size and location (non-field areas), we can effectively isolate player regions.

3.2. Team Classification

Once players are detected, the system classifies them into teams using a custom implementation of k-means clustering:

1. **Jersey Color Extraction:** For each detected player region, we extract color information in HSV color space, which provides better separation of hue from intensity compared to RGB.
2. **Green Field Filtering:** We apply a mask to filter out green pixels from the player regions, focusing only on jersey colors.
3. **Color Feature Extraction:** For each player, we compute average and dominant hue and saturation values using histograms.
4. **Feature Normalization:** Color features are normalized to ensure balanced influence of different components in the clustering process.
5. **K-means Initialization:** Unlike traditional random initialization, we find the two players with the most different jersey colors as initial centroids by computing pairwise distances between all player color features.
6. **Distance Calculation:** For each player, we compute the distance to both team centroids in the feature space.
7. **Team Assignment:** Each player is assigned to the closest team centroid.
8. **Centroid Update:** Team color centroids are recalculated based on the new assignments.
9. **Iterative Refinement:** Steps 6-8 are repeated until team assignments stabilize or a maximum iteration count is reached.

This binary clustering approach is specifically designed for soccer, where we assume exactly two teams on the field. The smart initialization strategy improves convergence compared to random initialization by starting with the most distinctive player colors.

4. Implementation Details

The system is implemented in Python using OpenCV for basic image processing operations. The core algorithms for player detection and team classification are custom implementations rather than relying on pre-built functions.

For Canny edge detection, we use a dynamic thresholding approach where the lower threshold is set as a percentage of the maximum gradient magnitude, and the upper threshold is set at 3× the lower threshold. This allows the system to adapt to different image contrasts.

In the field segmentation step, we use HSV thresholding with ranges [30, 40, 40] to [90, 255, 255] to identify green regions. This range was empirically determined to work well across different stadiums and lighting conditions.

Our custom k-means implementation includes several optimizations for the binary team classification problem:

- Smart initialization by finding the two most different jersey colors
- Early stopping when team assignments stabilize
- Normalization of color features to handle different scales
- Maximum iteration count to ensure convergence

The system processes a typical soccer image (1280×720 pixels) in under 0.5 seconds on a standard laptop, making it viable for near-real-time applications.

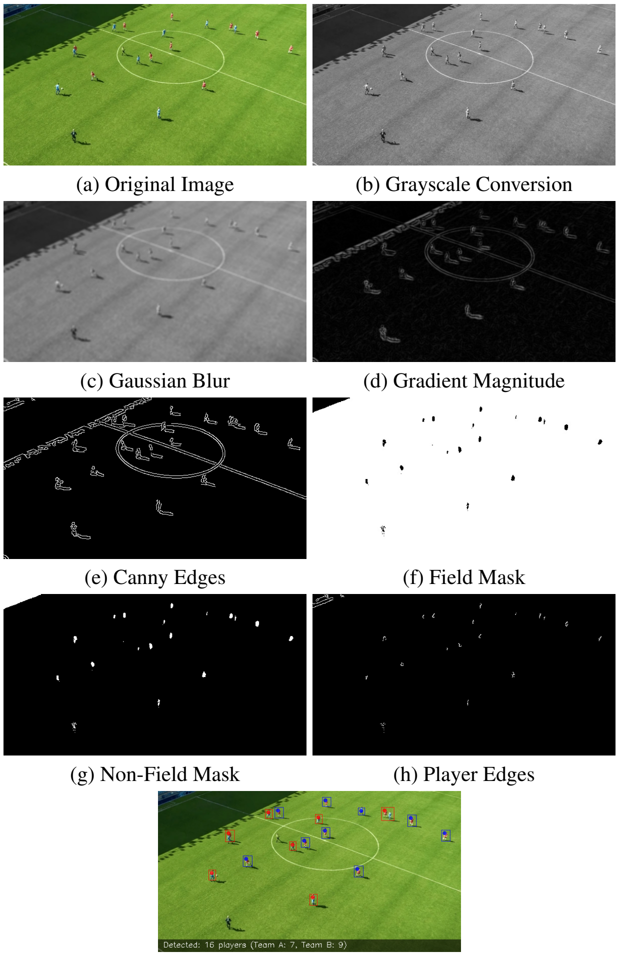
5. Results

We evaluated our system on a collection of soccer match images from different leagues, stadiums, and lighting conditions. The evaluation focused on two criteria: player detection accuracy and team classification accuracy.

5.1. Player Detection Results

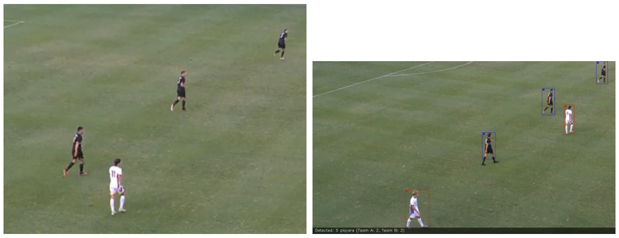
Figure 1 shows the intermediate steps of our player detection pipeline on a sample image. The progression demonstrates how the combination of edge detection and field masking effectively identifies player regions while filtering out field lines and other elements.

In Figure 2, we show successful player detection and team classification results on a test image. The system correctly identifies most players on the field and assigns them to the appropriate teams (red and blue bounding boxes).



(a) Original Image (b) Grayscale Conversion (c) Gaussian Blur (d) Gradient Magnitude (e) Canny Edges (f) Field Mask (g) Non-Field Mask (h) Player Edges (i) Final Detection

Figure 1. Step-by-step visualization of the player detection pipeline



(a) Original Image (b) Detection Results

Figure 2. Player detection and team classification results

5.2. Failure Cases

Our system is not without limitations. Figure 4 shows a case where the system struggled with complex scenarios:

Common failure modes include:

- False positives with field lines and other elements
- Merging multiple players into a single detection when they are close together
- Classifying the referee as a player

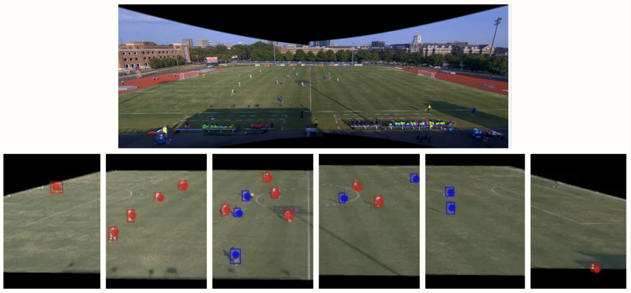


Figure 3. Results of player detection and team classification applied to a panoramic image that was split into segments for processing.

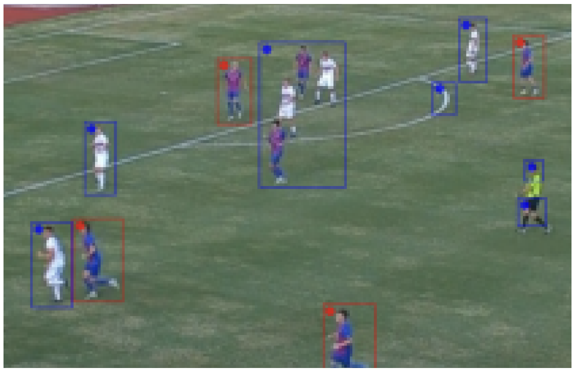


Figure 4. Example of a failure case: The system struggles with tightly grouped players and complex backgrounds.

6. Conclusion

This paper presented SoccerVision, a system for soccer player detection and team classification using classical computer vision techniques. Our approach demonstrates that effective results can be achieved without relying on deep learning or complex machine learning libraries, making it accessible for educational purposes and deployable in environments with limited computational resources.

The key contributions of this work include:

- A comprehensive player detection pipeline combining edge detection with field segmentation
- A custom binary k-means clustering implementation optimized for team classification

6.1. Future Work

Several avenues for future improvement exist:

- Incorporating temporal information across video frames to improve detection stability
- Adding player tracking capabilities to maintain identity across frames. Extending the system to detect additional elements such as the ball, referees, and goalkeepers

7. Individual Contributions

This project was developed by Abel Dagne and Laszlo Bollyky for CS 131. The contributions were split as follows:

Abel Dagne:

- Design and implementation of the player detection pipeline
- Implementation of the edge detection and field segmentation algorithms
- Development of the non-maximum suppression for overlapping detections
- Extensive testing and performance optimization of the detection system
- Documentation of the player detection methodology
- Implementation of the visualization pipeline for intermediate detection steps

Laszlo Bollyky:

- Design and implementation of the team classification system
- Development of the custom k-means clustering algorithm
- Implementation of the HSV color feature extraction
- Assembly and curation of the test dataset
- Creation of the panoramic image preprocessing pipeline
- Documentation of the team classification methodology

AI Tools Used:

- Cursor with Claude 3.7 Sonnet was used as a coding assistant for implementing the pre-processing step to intake the panoramic images and split them into segments for processing.
- We also used Cursor to implement the visualization tools to visualize the intermediate steps of the player detection pipeline.

References

[1] Chun-Wei Lu, Chih-Yang Lin, Chao-Yong Hsu, Ming-Fang Weng, Li-Wei Kang, and Hong-yuan Liao. Identification and tracking of players in sport videos. In *Proceedings of the Fifth International Conference on Internet Multimedia Computing and Service*, pages 113–116, 2013. 1