

README

Ball & Player Tracking - Team Differentiation

CS 5330 Pattern Recognition & Computer Vision Professor Bruce Maxwell, PhD. Spring 2024

Group Member Names:

- Joseph Nelson Farrell
 - Harshil Bhojwani
 - Leonardo DaGraca
 - Priyanka Dipak Gujar
-

Presented Live:

Yes.

Project Description

The project aimed at developing a system to track ball and player movements in sports using the You Only Look Once (YOLO) model architecture. Inspired by existing applications of YOLO in sports analysis the project team seeks to extend its utility in soccer. The project involves utilizing the YOLO model with sport-specific data to enable distinguishing between teams, tracking of players, referees, and the ball. Additionally, with the help of openCV methods, implemented classical CVPR techniques, in this case the Kalman Filter, to predict the position of the ball if it fails to be detected. This project highlights capabilities of combining deep neural networks classical computer vision techniques. The latest iteration, YOLO8, released in January 2023, serves as the foundation for this project. By leveraging the strengths of YOLO and customizing it for soccer analysis, the project aims to contribute to the advancement of sports analytics and enhance understanding of player and ball movements.

Links/Urls:

[Link to Example Videos](#)

Operating System & IDE:

- MacOS
 - Visual Studio Code
-

Executing the Program:

Step 1: (optional)

If the video is long and the user would like to clip it, the user can execute this command:

```
python3 clip_video.py {original file} {output file} {start frame} {end frame}
```

The parameters here are required and there is no default settings.

This will clip the video based on the `start_frame` and `end_frame` arguments provided.

The clipped version will be deposited in the `data_raw` folder upon completion.

Step 2: Process Video

The current setup requires BGR color ranges for the two team jerseys and the referees. Ranges currently defined are:

```
color_ranges = {  
    'white': ([187, 169, 112], [255, 255, 255]),  
    'red': ([0, 0, 50], [100, 100, 255]),  
    'yellow': ([0, 100, 100], [10, 255, 255]),  
    'black': ([0, 0, 0], [50, 50, 50]),  
    'blue': ([43, 31, 4], [250, 88, 50])  
}
```

If the user wishes to process a video where the players and referees are wearing different colors additional color ranges can be added to `utils.utils_label_teams_boolean_mask.py`.

The program may then be executed in the same way that follows.

To reproduce the results obtained and described in the report execute the following command:

```
python3 process_video.py original_1.mp4 original_1_processed.mp4 "  
["white", "red", "yellow"]""['MCI', 'RMA', 'Ref']" 'False'
```

The arguments provided are:

- original file name
- processed file name
- the colors to be used to differentiate teams and referees
- the labels
- a bool indicating NOT to run default YOLO on the entire clip.

These values are all also default values and the program will execute the same with:

```
python3 process_video.py
```

Additional Functionality:

If the user would like to generate graphics of the boolean masking to inspect the classification, this can be done by activating the boolean variable `generate_mask_example_figs` on line 90. This will save figures to the `figs` folder using `matplotlib`. To inspect the masking without generating figures using `openCV`, the user can activate the boolean variable `show_masked_images` on line 89.

Additional Files:

`utils/utils_label_players_HSV.py`

- This file contains functions that were used to explore using HSV histograms and histogram matching to perform team classification.

`utils_general.py`

- This file contains the following helper functions:
 - `clip_video`
 - This function is used to clip videos.
 - `track_ball`
 - This function is used to draw a ployline that tracks the balls trajectory.

`create_figs.py`

- This file is used to generate Matplotlib figures of the specific frames.

Acknowledgements:

Please see `PRCV Final Project Report` doc.