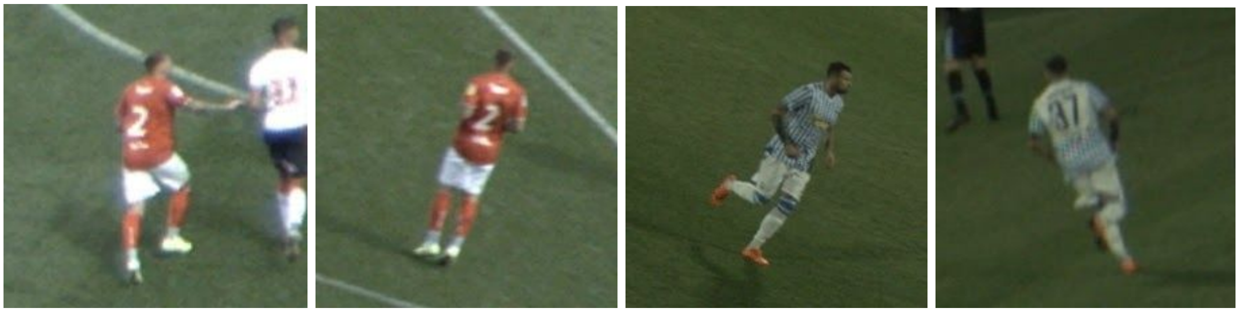
HAWK-EYE
INNOVATIONS

Player & Team Recognition Assignment

Overview:

An important challenge in any player-tracking system is determining the team and identity of a player on the pitch. In a major sport such as football it is impossible to collect images of every team and player in advance and treat this as a classification problem. Especially as each team has a home and away kit, with most teams releasing a new jersey style each season.

Given a set of images and person detections from a stationary camera, your objective is to create a model that can recognize a person across the sequence of images and determine what team they belong on.



Part 1:

For Part 1 you should train a model for player and team recognition. Player recognition is significantly harder than team recognition, which is why we are interested in your approach and methodology. This may be done in Python or C++. We recommend using PyTorch for both Part 1 and Part 2 as it provides good examples on how it can be used in a C++.

We provide a selection of image tiles, where each tile is centered on an individual player. These are grouped into team sub-directories, and then into player directories.

bristol/person_0/0.jpg

bristol/person_1/0.jpg

middlesbrough/person_2/0.jpg

. In the example above bristol and middlesbrough refer to the team labels and person_0, person_1, and person_2 refer to the person labels.

As part of this you should conduct a brief investigation of your final models behaviour. Does it fail in any specific conditions such as when a player faces away from the camera? How well does it generalise to teams and players it hasn't been trained on?

Input Data:

1. The assignment does not have to use all of the image tiles

Required Output:

1. A jupyter notebook saved as a PDF / HTML investigating the output of your model and its behaviour on the image tiles.
2. A saved model that can easily be run on a directory of image tiles.

Part 2:

Expand on your solution to Part 1 by creating a C++ application that runs the model you've trained on a set of images and person detections from a stationary camera. For each image we provide a corresponding CSV where the filename is the frame-number; for example *1234.jpg* and *1234.csv*. Each line in the CSV corresponds to a person detection, where we specify the top-left and bottom-right XY coordinates of the person detection.

Required Output:

1. A copy of the frames where you've coloured the person detections by team
2. A copy of the frames where you've coloured the person detections by person

General Software Requirements:

1. Part 1 may be written in C++ or Python, Part 2 must be written in C++.
2. You should write production quality code
3. The components of your solution should be re-usable. i.e., they would plug-in to a larger piece of software without refactoring or re-organising.

4. Uses classes and ensure that public interfaces are as clear and useful as it is possible to make them.
5. Include code and documentation of your experiments that you did before finalising on your solution. This does not need to be production quality.