THE UNIVERSITY OF ADELAIDE

SCHOOL OF COMPUTER SCIENCE

HONOURS PROJECT THESIS

---

# Visual Tracking for Application to AFL Football

---

*Author:*
Hayden Faulkner

*Supervisor:*
Dr. Anthony Dick

November 7, 2014

# Contents

# List of Figures

# List of Abbreviations

**AFL** ..... Australian Football League

**B&W** ... Black and White

**DET** .... Detection Error Tradeoff

**FN** ...... False Negative

**FNR** .... False Negative Rate

**FP** ...... False Positive

**fppi** ..... False Positives per Image

**FPR** .... False Positive Rate

**FPS** ..... Frames Per Second

**H** ........ Height

**HOG** .... Histograms of Oriented Gradients

**HSV** .... Hue Saturation Value (Colour Format)

**ID** ....... Identification

**LBP** ..... Local Binary Patterns

**MOTA** .. Multi-Object Tracking Accuracy

**MOTP** .. Multi-Object Tracking Precision

**PPV** .... Positive Predictive Value

**PR** ...... Precision Recall

**RGB** .... Red Green Blue (Colour Format)

**ROC** .... Receiver Operating Characteristic

**SVM** .... Support Vector Machine

**TN** ...... True Negative

**TNR** .... True Negative Rate

**TP** ...... True Positive

**TPR** .... True Positive Rate

**W** ....... Width

# Abstract

This work introduces a visual tracking framework for tracking Australian Rules Football (AFL) players in match scenarios. Although pedestrian tracking is a well studied problem in the literature, the particular application of AFL is yet to be explored. The AFL scenario brings about a variety of unique challenges not seen in general tracking problems.

Using a tracking-by-detection approach, the framework comprises three main modules, a detector, a classifier which classes detections into teams, and a tracker. The detector module utilises an efficient cascade of classifiers in combination with the popular machine learning method, AdaBoost, on feature descriptors formed by histogram of orientated gradients (HOG). Team classification is based on weighted colour histograms, and employs a support vector machine (SVM) approach with separate one-vs-all team classification models. Two very different approaches, a local Kalman Filter method and a global energy minimisation technique, are combined to form a more suitable tracking solution.

The findings suggest that current start-of-the-art pedestrian approaches only work when adapted and fine tuned for the AFL problem set. Training a detector on an AFL dataset, rather than a pedestrian dataset, is key to success for not only the detector but also the rest of the downstream pipeline. An appropriately trained AFL detector is able to handle more extreme lighting conditions and pose variations than many state-of-the-art pedestrian detectors. However the AFL detector is unable to perform on pedestrian benchmark sets, highlighting the specificity of the AFL problem. Basing team classification solely on colour works remarkably well when focus is applied to player uniforms using a spatial weight. However, colour and hence classification, is extremely susceptible to the varying lighting conditions which are ever present in AFL matches which are held outdoors. It is therefore necessary to train many different team classification models covering a variety of different environmental conditions and applying them appropriately. The frequently occurring long term occlusions and criss-crossing movement between multiple players as well as the fast speed and direction changes of players makes tracking for the AFL situation incredibly difficult, more-so than for the general pedestrian case. The global energy minimisation tracking technique utilised herein is able to handle some of these problems on occasion, however further work is suggested in constructing a more appropriately refined tracking system for the AFL problem domain.

# Declaration

I declare that this thesis is a record of original work and contains no material which has been accepted for the award of any other degree or diploma in any university. To the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text.

Hayden Faulkner

# Acknowledgements

Despite Honours being the most difficult and stressful year of my life thus far, I am incredibly privileged to have had the opportunity to undertake such a high level degree. Im also very lucky to have had an abundance of support throughout the year which has allowed me to complete the journey to the best of my ability.

Dr. Anthony Dick, my supervisor and mentor, has been a guiding light providing knowledge, experience and encouragement. His dedication to this work coupled with his friendly and upbeat attitude has been a major inspiration for my continual determination to succeed.

Everything that has allowed me to reach this position in my life has been provided by my family, without them I couldnt be here. Their constant support and dedication not only to my education, but to my life, cant be described in words other than that they couldnt possibly do any more.

Id like to thank all of my amazing friends and apologise for not being available at many times throughout year.

Thank you to my partner Ella, who endured her own Honours project this year, it was nice to go through it together. You bring so many wonderful things to my life and teach me more than any university degree ever could.

# 1. Introduction

Visual object tracking is the process of tracking a particular object, or group of objects, path in a video sequence. It is a fundamental task in the field of computer vision and is an essential part of many applications related to surveillance, human-computer interaction, vehicle navigation and video analysis. The growing availability of high speed computers and high definition video cameras, coupled with the increasing demand for automated video analysis systems have led to widespread research and significant advancements over the past few decades. The difficulties involved in visual tracking arise from factors including object appearance changes, abrupt motion changes, partial and full occlusions, illumination differences, camera motion as well as noisy and low resolution images.

Driven by the priority of the surveillance and vehicle navigation applications, a substantial amount of tracking research focuses specifically on pedestrian tracking and detection. To a much lesser extent, the tracking of players in sports video has also received some attention, with most work revolving around popular international sports such as soccer and basketball. In Australia however, the most popular sport is AFL football, with a supporter base of over 750,000[1]. The motivation behind the use of visual tracking systems in sports, including for the AFL application, is to provide a foundation for a system that is able to automate game statistics for match and player analysis.

AFL football is currently only played professionally in Australia, and it is very unique when compared to other sports. This uniqueness presents some distinct properties and challenges that aren't found in other sports and pedestrian tracking problems. Specific challenges related to the AFL situation are (Figure 1.1):

1. the large size of the field makes covering the entire field at a reasonable resolution difficult;
2. the number of persons constantly needing to be tracked is close to 50;
3. the fast movement of players, with sudden direction changes based on play, is generally more erratic than in other sports and for pedestrians whom often follow relatively straight paths;
4. the regular bunching of players into dense packs causing many difficult, often long lasting occlusions;
5. the lack of identifiable appearance differences between players on the same team, and sometimes players on different teams;

[1] http://www.aflmembershipnumbers.com/2014-sort.html

6. players are more deformable and take a more varied set of shapes, for example when making large strides whilst running or when lying on ground after contact with another player; and

7. the light variability of the outdoor environment (for example the bright and dark areas of the field with sunny and shadowy conditions).


*Figure 1.1: Some of the AFL's most challenging and unique situations. Numbers refer to the list numbers above.*

Before the commencement of this work, the tracking of AFL football players had yet to be identified in any research. This project investigates the viability and performance of current state-of-the-art methods for the AFL scenario and refines such methods to improve their results for the AFL application. A tracking-by-detection approach was utilised in this project, and the following pipeline was constructed to obtain player tracks from video sequences (Figure 1.2).


*Figure 1.2: The AFL overall pipeline*

Represented by the dark blue boxes, there are three main modules in the pipeline each with its own purpose and contribution to the final result. The detector searches individual video frames for pixels that represent players, and highlights these pixels by placing a bounding box around each player. The team classifier examines each box and its contents determining which team the player belongs to. The tracker combines both of these findings to build tracks (paths) for each individual player throughout the sequence (Figure 1.3).

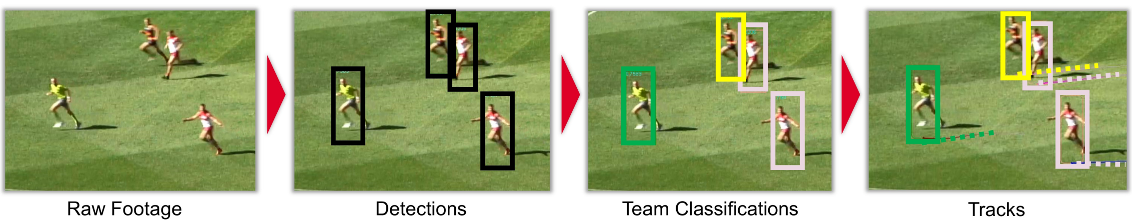



*Figure 1.3: The different outcomes of each of the module.*

The rest of this thesis is structured with reference to the pipeline shown in Figure 1.2, with individual chapters for each of the three main modules, as well as chapters describing related literature and preliminary work. The next chapter will introduce some background related to the general tracking-by-detection process, before presenting more specific related literature on detection and tracking as well as methods utilised in sporting applications. Chapter 3 will give a more detailed description of the framework utilised in this project and will also describe all preliminary work including capturing AFL match footage and dataset construction. Chapter 4 will describe the implementation of the detector module and will also present evaluations and discussion on the performance of the detector. Similarly in Chapters 5 and 6, the team classification module and tracking module respectively will be described and evaluated. Chapter 7, the conclusion, summarises the project and the key findings as well as directions for further work.

# 2. Background and Related Work

Visual tracking is the process of locating a particular object of interest (target), or set of objects of interest (targets), over time in a video sequence. There is a vast expanse of literature describing different approaches to visual tracking and subsets of the problem. As previously mentioned, with the recent advancements in detection and classification methods [40] [65] [10], tracking-by-detection has become an increasingly popular approach [6] [7] [2] [30]. Using a detector has a number of advantages including being able to better handle problematic situations such as cluttering, occlusions and varying backgrounds as well as being relatively resistant to excessive model drift [8] caused by trackers altering their detection model online.

The tracking-by-detection process is made up of two distinctly separate stages (Figure 2.1). Firstly, an object detector is applied to individual video frames separately to obtain target positions, and potentially, target appearance information. Secondly, a tracker uses the position and appearance information to correlate detections referring to the same target over some period of time.


*Figure 2.1: A high level overview of the tracking-by-detection process*

## 2.1 Detectors

### 2.1.1 Overview

Detectors exhaustively search subsets of static images for patches that match a particular predefined pattern representing the target, or object of interest, in this case a player or official (Figure 2.2).

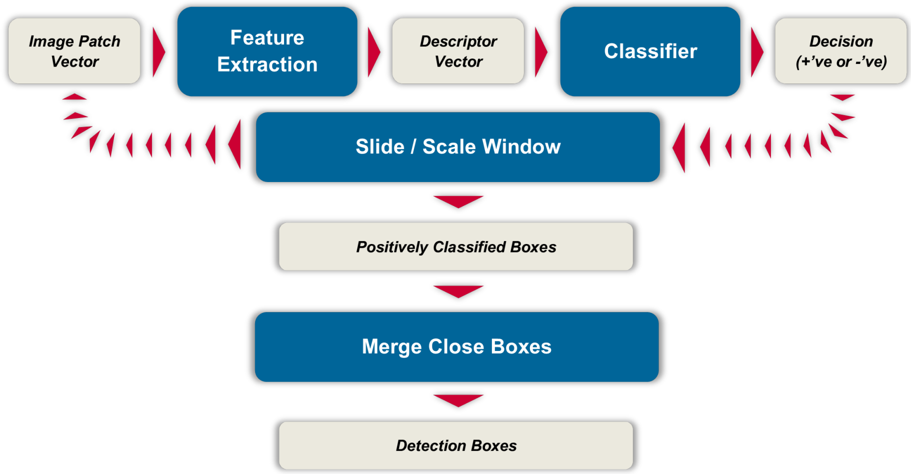



*Figure 2.2: The general detection process for an image*

Firstly, a window (box) of a particular ratio is skimmed across the image extracting sub-image patches to evaluate in search for the target. The window moves left to right, in rows moving down the entire image (frame), with some degree of overlap. To accommodate for a target appearing at different sizes in the image, once the image has been scanned once, the window's scale increases and the process repeats. Each image patch extracted with the window can then be thought of as a $w * h * c$ length vector of pixel intensity values where $w$ and $h$ are the patch dimensions and $c$ is the number of channels (3 for RGB, 1 for B&W).

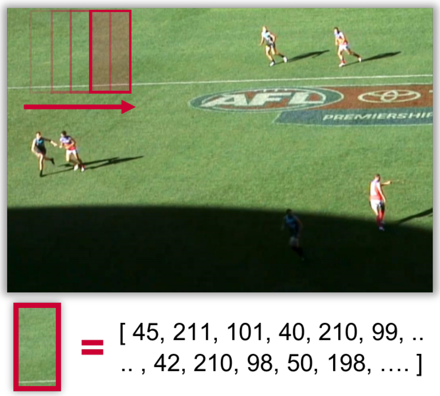



*Figure 2.3: Sliding window process for extracting image patch vectors*

Each image patch, now represented as a vector, needs to be evaluated such that a decision is made on whether the target appears within that patch. This decision is made using a two step process, feature extraction followed by classification. Feature extraction transforms the patch vectors into a more discriminative form, beyond the simple pixel intensities. Many different feature extraction processes have been developed and the choice of which depends on the application. It is key to chose the right feature set to best discriminate and highlight differences between patches containing a target versus patches not containing a target. Different feature options are described in Section 2.1.2.

The transformed patch vectors, generally referred to as descriptor vectors, are then fed into a classifier which makes the decision on whether they contain a target or not. Multiple options for the classification process are also available and the choice is dependent on the particular application. The classifier makes a decision based on some pre-learned decision model. The model requires the classifier to be trained with ground truth data, so the classifier has some understanding of what constitutes a positive 'yes this patch contains a target' or negative 'no it doesn't' result. The training process is carried out before the detector is used, and is much more time consuming and computationally intensive relative to the actual detection process. Training involves providing the classifier with a large number of image patches that are each manually pre-labelled as either containing a target (positive sample) or not (negative sample). The image patches go through the same feature extraction processes to build more discriminative descriptor vectors. The classifier modifies (learns) its decision model to best classify the training data correctly. It is important that the training samples are varied enough to cover the distribution likely to be found with the specific application. Different classifier options are described in Section 2.1.3.

When an image patch is found to contain the object of interest, it is marked in the original image (frame) as a box. Due to the overlapping of patches it is generally the case that multiple patches will be classified as positive for the same actual target, resulting in multiple boxes per target. This behaviour is undesirable, it is much more beneficial to have a single box for each individual target (Figure 2.4). Again, depending on the application, different methods of merging the boxes could be implemented, but most are based on confidence and overlap measures.

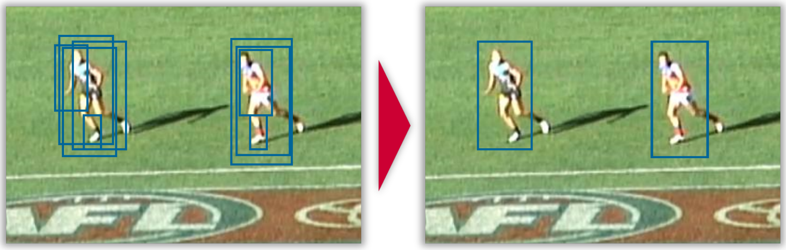


*Figure 2.4: The merging process*

As previously mentioned, a substantial proportion of tracking and detection literature is related to pedestrians, as it is most relevant to high profile applications in surveillance and vehicle navigation. Advantageously, most pedestrian methods will be applicable to this work as player movement and appearance is relatively similar to pedestrians than is the case for animals, cars and other movable objects. Keep in mind however, the AFL scenario has challenges which make it both different and, in many cases, more difficult than general pedestrian problems. The detection techniques described below are only a subset of many years of research in object and pedestrian detection, for a more comprehensive review of pedestrian detection methods please refer to [17], [28], [19].

### 2.1.2 Features

The classification model is reliant on a discriminant descriptor vector that can describe each of the image patches in such a way that they can be easily classed as either containing a player or not. Features are necessary as the use of direct pixel intensities is generally too high dimensional and also not discriminant enough to reliably categorise classifications. There are many different features that could be used to describe an image patch sample, and choosing the feature set that best discriminates the data correctly can be difficult. In this section relevant features are categorised and described, with a focus on the more successful and applicable for this project.

#### 2.1.2.1 Intensity Based Features

Early attempts at object detection in pictures using machine learning techniques date back over twenty years, with Sung et al. first using the ‘example based learning’ technique, as they called it, for frontal face detection [61]. Their approach used 19x19 pixel windows, with descriptor vectors being constructed solely from the image pixel intensities.

Oren et al. later adapted and improved the ‘example based learning’ approach for pedestrian detection [54]. They noted that faces, despite their intra-class variability, are all fairly similar in terms of shape and structural layout of facial features. Such pattern similarity was not prevalent for pedestrians, and variances in pose, colour and backgrounds meant pixel intensities were not adequate for sole use as the descriptor vector. They proposed using Haar wavelet templates, which are calculated based on the average of the difference between neighbouring rectangular image regions (Figure 2.5). The wavelet template ‘features’ captured the ordinal relationships and structure of image regions based on the ratio of the brightness distribution between two neighbouring sub regions, rather than the pixel intensities themselves.

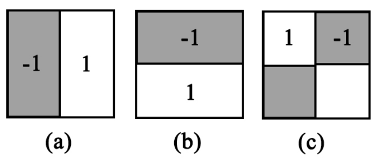



*Figure 2.5: The 3 types of 2-dimensional non-standard Haar wavelets; (a) vertical, (b) horizontal, (c) corner. Reproduced from [54].*

Viola and Jones [65], [64] built upon these past ideas and introduced a few key advancements that are still used in modern detectors. They introduced two new Haar-like features in addition to the wavelet templates as well as the concept of the integral image which allowed for much faster feature calculation of Haar-like features.

#### 2.1.2.2 Gradient Based Features

In 2005, Dalal and Triggs [10] introduced the histogram of orientated gradients (HOG) feature, which is gradient based and has been shown to have greater success than intensity based features, especially for pedestrian detection. HOG is currently the most widely used feature for a range of detection problems, as it is able to capture complex and accurate shape information in a compact form, while still being fast to compute. It works by firstly splitting the image into a grid consisting of small cells (Figure 2.6). For each cell a descriptor is formed by compiling a histogram of oriented gradients from pixel gradient values within that cell. The gradient values are calculated using a simple kernel, and weighted based on the gradient's magnitude. Cells are grouped together into blocks, with all cells in a single block being normalised together, providing improved illumination and shadow invariance. Blocks also generally overlap, resulting in smoother normalisation across the entire image.

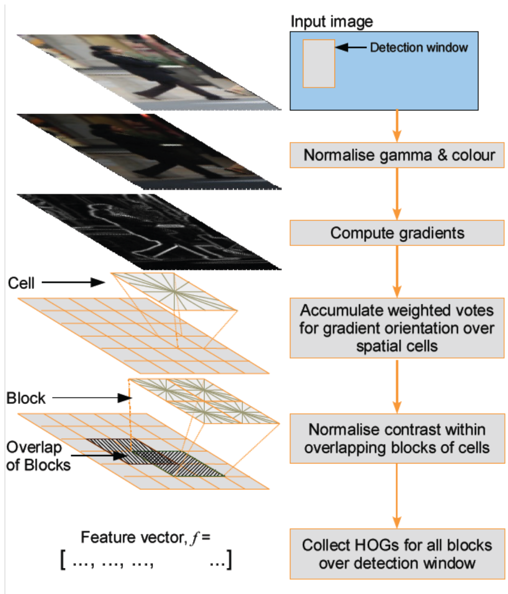



*Figure 2.6: Overview of HOG feature extraction process. Reproduced from [34].*

#### 2.1.2.3 Texture Based Features

The local patterns and textures of images have been increasingly researched over the past decade with one of the most studied being the Gabor wavelet [44]. The Gabor wavelets, also known as Gabor filters, are orientation and scale tunable edge and line detectors, which when analysed, can describe underlying texture patterns. Experiments showed Gabor filters were quite robust to scale and rotation variance.

A more recent texture based feature that has been shown to be very efficient is Local Binary Patterns (LBP) [52]. Developed by Ojala et al., LBP encapsulates spatial texture information in a very simplified, yet effective, form. It is tolerant to illumination differences and can be computed easily and quickly. LBP features are calculated by comparing pixel intensity values with intensities of neighbouring pixels of some radius. When a neighbouring pixel's intensity is greater than the central pixel's intensity plus a chosen threshold value it is assigned a 1, otherwise it's assigned a 0 (Figure 2.7). These binary digits are unfolded in a particular direction to form a vector which is usually transformed into a decimal number for easier usage. A histogram of the frequency of each of the decimal numbers in a cell is then calculated and normalised. Each of the cell's histograms are then concatenated together to form the vector for a window.

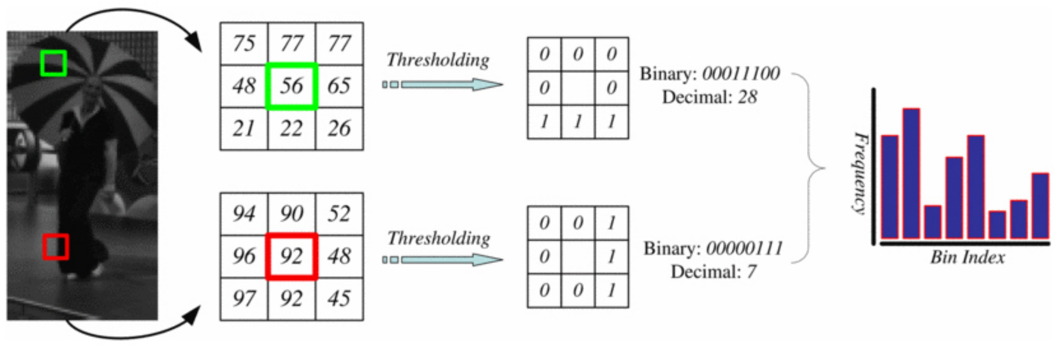



*Figure 2.7: Illustration of LBP. Typically the binary codes obtained by local thresholding are transformed into decimal codes. Note that in this example a threshold of 30 is used, which is slightly different from the original LBP. Reproduced from [50].*

There have been a number of LBP variants proposed such as Local Ternary Patterns (LTP) [62] which improve noise sensitivity in uniform image patches; and Multi-scale Block LBP (MB-LBP) [41] which uses integral images to calculate features based on average values of block sub-regions rather than individual pixels, capturing a more complete image representation.

Mu et al. [50] introduced two variants of LBP, Semantic-LBP and Fourier LBP which can work in colour space and were proven more suitable for human detection in some cases. Semantic-LBP (S-LBP) alters the representation of LBP from decimal numbers to reduce on space complexity. Consecutive 1 bits form arcs around the central pixel, which can be compactly represented by their principle direction and length (Figure 2.8). Fourier-LBP (F-LBP) is a soft version of LBP that uses a similar concept as the Fourier boundary descriptor [29]. A soft LBP is important as it can potentially avoid local errors caused by thresholding, and also allows for controllable compression.

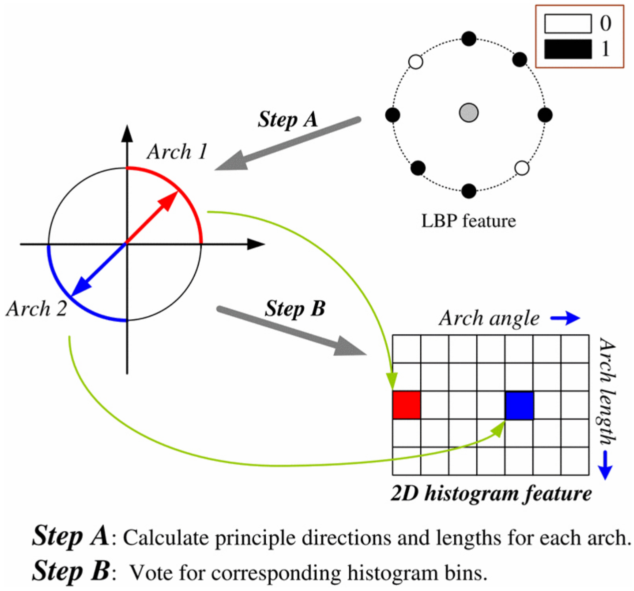



*Figure 2.8: Computing S-LBP. Note that the ring feature has two segments of arches, thus a non-uniform one will be abandoned in practice. See text for more details. Reproduced from [50].*

#### 2.1.2.4 Motion Based Features

Beyond appearance features, motion has been used as another important characteristic for detecting pedestrians in video sequences, without the need to use complete tracking. Viola et al. [66] proposed scanning a detector using thier Haar-like features over two consecutive frames of a video sequence to take advantage of both appearance and motion information. The authors later extended this approach to consider more than two consecutive frames [36]. Their technique greatly improved both runtime and accuracy for their detector, showing promising results for low resolution detections.

#### 2.1.2.5 Feature Combination

HOG is the main basis feature used for pedestrian detection problems. However the extension and use of HOG in combination with other features has provided considerable advancements.

Wojek et al. [69] experimented with a number of the more prominent features described above, finding HOG the most single effective feature. They then experimented with a combination of features, and found the combination of Haar-like features, shapelets, shape context [3], and HOG worked better than sole HOG.

The above combination was later extended by Walk et al. [67] to include motion features derived from optical flow (HOF) [11], and a new feature, local colour self-similarity. This feature captures pairwise relations of spatially localised colour distributions. Self-similarity restricts colour comparisons to single window sub-regions, preventing the entire colour distribution of the window adversely affecting the result.

Wang et al. [68] introduced a new descriptor combining HOG and a form of LBP (HOG-LBP) to better handle partial occlusions. Two kinds of detectors are learnt, a global detector for scanning entire windows, and part detectors for local regions. For each window an occlusion likelihood map is calculated using the HOG response from the global detector for each block, and segmented using Mean Shift [27] (Figure 2.9). The negative segmentations of a window are considered to be occlusions. If a window presents only a partial occlusion with a high likelihood, the partial detectors are used to perform the final classification.

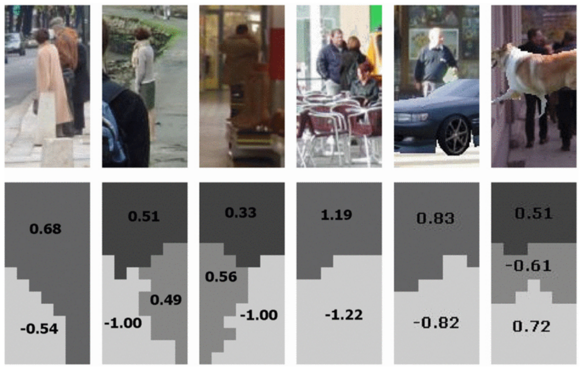



*Figure 2.9: The first row shows ambiguous images in the scanning windows. The second row shows the corresponding segmented occlusion likelihood images. For each segmented region, the negative overall score, i.e. the sum of the HOG block responses to the global detector, indicates possible partial occlusion. Reproduced from [68].*

Hussain et al. [33] experimented with combinations of both HOG, LBP and Local Ternary Patterns (LTP), finding HOG+LBP, HOG+LTP, and LBP+LTP perform reasonably equally, and the combination of all three HOG+LBP+LTP performs marginally better overall.

With the use of combinations of features, dimensionality of the feature vector can quickly become unmanageable, as seen in [59] where the use of features containing edge, texture and colour information created a dimensionality of over 170,000. Using Partial Least Squares analysis [70] the authors were able to reduce the dimensionality to just 20, while still keeping acceptably discriminate information in the feature vector.

### 2.1.3 Classifiers

The classification model is at the core of the detector, abstracting a descriptor vector into a binary decision on whether the image patch the vector represents contains the object or not. So in the AFL case a classifier will be used to decide whether a particular image patch contains a player or not, and further to decide what team a player belongs to. Recent classifiers as part of pedestrian detection systems have generally been one of two machine learning schemes: AdaBoost [26] on a classifier cascade, and Support Vector Machines [9]. It is necessary for both classifier methods to be trained, using a training set of predefined ground truth of positive and negative samples.

#### 2.1.3.1 AdaBoost in a Cascade of Weak Classifiers

Viola and Jones introduced the cascade of weak classifiers approach with AdaBoost alongside their feature contributions in [64]. They later used the same classifier approach with their motion based features [66] [36]. The cascade is constructed from a set of weak classifiers that each eliminate some of the false sub-windows (Figure 2.10).

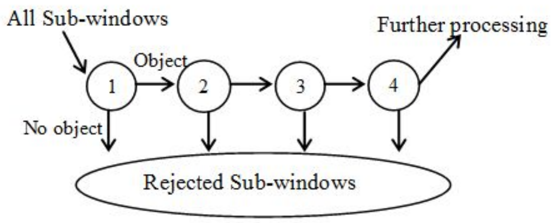



*Figure 2.10: Schematic depiction of the detection cascade containing a series of weak classifiers. The initial classifiers eliminate a large number of negative samples with very little processing. Subsequent layers eliminate additional negatives but require additional computation. Reproduced from [39].*

AdaBoost [26] is an adaptive learning algorithm that is generally used in combination with classification algorithms to improve their performance. In terms of the cascade, AdaBoost calculates a weighted sum of the weak classifiers to form a process which is overall a strong classifier. During the training phase each weak classifier is adjusted to pick up the mistakes of the previous weak classifier (Figure 2.11). In most of the literature describing features, AdaBoost has been used for automatic optimal feature selection and weighting, to form a strong feature set and classifier.

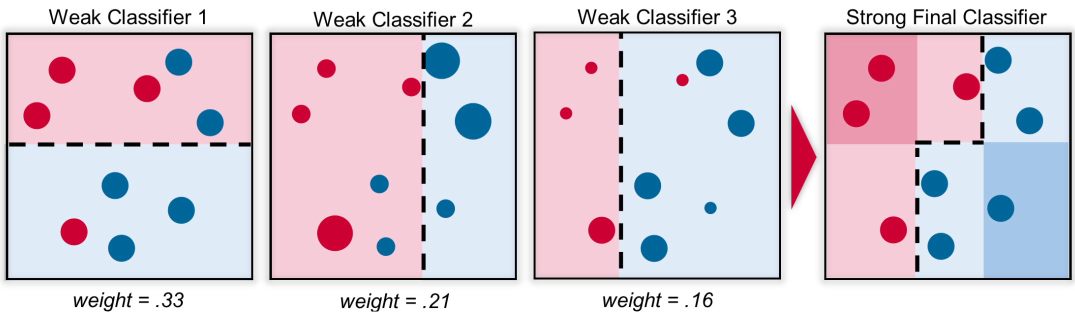



*Figure 2.11: The AdaBoost process of weighting weak classifiers to build a strong classifier using weighted errors of misclassified training samples. The size of the dots represent the error weighting on each sample.*

#### 2.1.3.2 Support Vector Machines

A support vector machine [9] constructs a hyperplane or set of hyperplanes in a high-dimensional space to best separate the training data into two or more classes respectively. The hyperplanes are calculated such that the distance between a hyperplane and any point of any class is maximised so they will naturally fall between classes (Figure 2.12). This allows for new unseen points to be classified based on the side of the hyperplanes they are mapped to. Originally, SVMs were proposed as linear classifiers unable to handle non-linearly-separable classes, but that is now not the case with use of the 'kernel trick' [1]. The kernel trick implicitly maps the non-linearly-separable points to a higher dimensional space where they eventually become linearly separable.

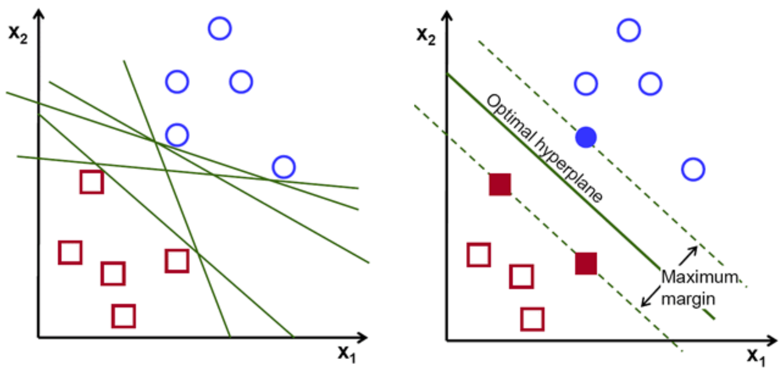



*Figure 2.12: Left: Linearly-separable hyperplanes. Right: The chosen optimal hyperplane. Reproduced from OpenCV website*[1].

Dalal and Triggs [10] used a linear SVM to separate their HOG vectors into classes of pedestrians or background. Zhu et al. [72] were able to significantly improve the efficiency of this approach by replacing the SVM with a boosted cascade-of-rejecters

[1]http://docs.opencv.org/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html

like that used by Viola et al. [65], [64]. They found the fixed sized blocks used by Dalal and Triggs didn't contain enough discriminative information to allow them to be rejected in the early stages of the cascade. Hence they used AdaBoost [26] to select a discriminative subset of blocks from a larger set of blocks of varying sizes and aspect ratios. Using work by Porikli [56] introducing the Integral Histogram, they were able to efficiently calculate gradient orientation histograms over arbitrary sized rectangular regions of the image.

### 2.1.4 Parts-Based Approaches

Pedestrians can often take many different forms based on their pose and the viewpoint of the camera. To deal with such articulation differences, and also partial occlusion problems, parts-based descriptor techniques have been devised to break a detection down into parts that are more easily recognised and classified.

Felzenszwalb et al. [23], [24] propose a system based on mixtures of multiscale deformable part models. Using HOG at different scales with a star-structured part-based model, the authors build higher level coarser root filters and lower level finer part filters (Figure 2.13).

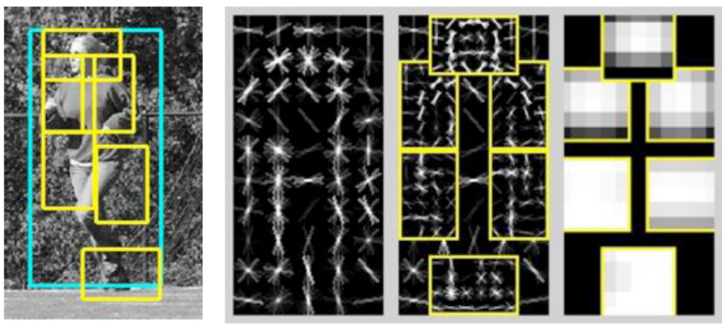


*Figure 2.13: The person parts model, defined by a coarse template, multiple higher resolution part templates and a spatial model. Reproduced from [23].*

Dollár et al. [14] extended the above approach to a new Multiple Component Learning (MCL) method which again automatically learns individual component classifiers, combining them into an overall classifier. The addition of the Haar-like features and Adaboost allowed for further discriminative power.

Tang et al. [63] focus on the problem of missed detections due to partial and full occlusions by building a double-person detector, which they also integrate with a single-person detector to build a joint person detector. Instead of treating occlusions as distractions in the training data, they leverage the idea that person/person occlusions have very distinguishing appearance patterns that can be utilised in training. The double-person detector is built on the Deformable Parts Model (DPM) approach of [23], [24] with initialisation of three different occlusion levels, 5% - 25%, 25% - 55% and 55% - 85% (Figure 2.14). The joint detector is built by using the single-person detector and the double-person detector as different components, with each having the three occlusion levels. During training, samples can be reassigned to different occlusion components of the DPM model, but 2-person samples are prevented from assignment to 1-person components and vice versa (1-person samples to 2-person components).

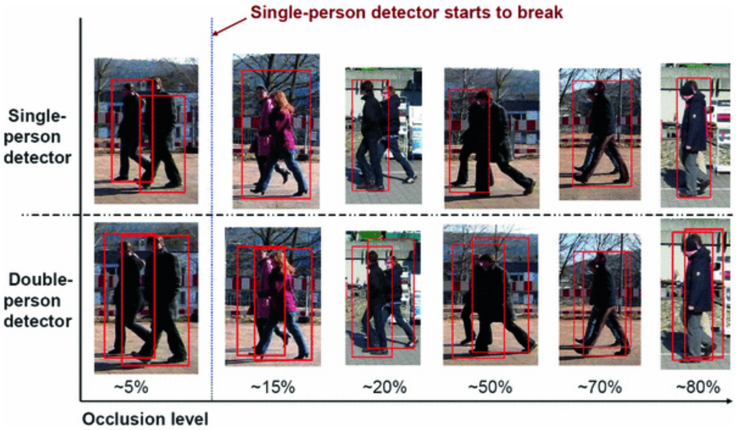



*Figure 2.14: Qualitative comparison of single- and double-person detectors for different occlusion levels. Reproduced from [63].*

### 2.1.5 Runtime Improvements

In 2010, Dollár et al. [15] [13] suggested that the bottleneck of modern detectors was the construction and evaluation of the multi-scale image pyramid. They went on to propose that different intermediate scale features can be approximated from sparsely sampled scales, reducing the overall number of scales that need to be constructed and evaluated. Their approximation technique was shown to reduce runtimes by at least an order of magnitude, while only slightly (1-3%) decreasing accuracy.

## 2.2 Trackers

### 2.2.1 Overview

Trackers are much more varied in their process when compared to modern day object detectors, however they all work towards the same goal of matching detections across frames. Tracking-by-detection trackers attempt to string together detections referring to the same target across sequential frames, to build a path or 'track' for each target over time. The difficulty with this process lies in the decision of what detections correlate to which target, as well as the ability to handle erroneous and noisy 'jittery' detections. Within individual frames, detections may not reflect the true position of a target, and there may be none or too many detections for a single target. These problems need to be handled by the tracking framework, to hopefully construct a single continuous track for each target within the sequence (Figure 2.15).

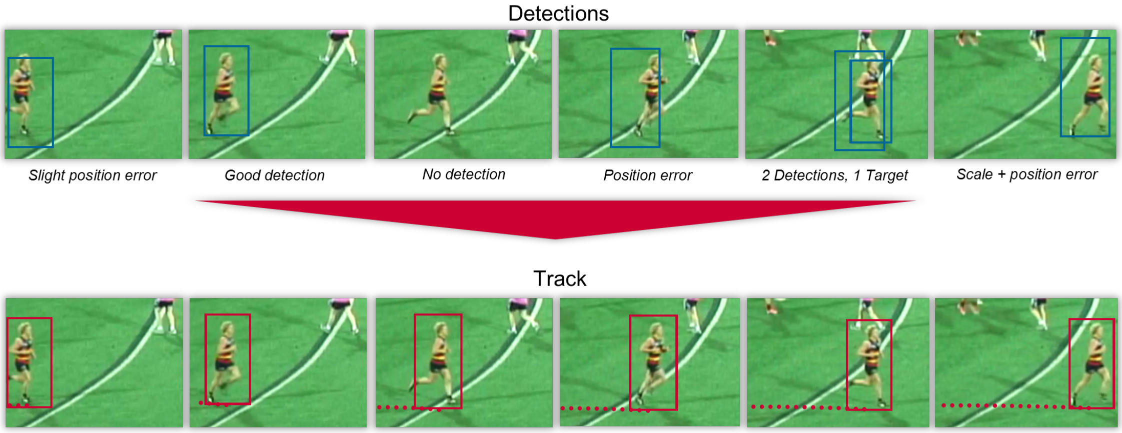



*Figure 2.15: The generalised tracking process. Associating detections across frames to build target paths. Needs to handle detection error and noise, examples of which are shown.*

Like detection techniques, there have been a vast number of different tracking techniques proposed over the past few decades. Early literature focused on single target tracking [5], however with computational capabilities improving and the application necessity, multi-target tracking has become the focus of more recent contributions. Multi-target tracking problems have been proposed as data association problems, where assigning detections to targets is an important factor, as well as path estimation. A more comprehensive review of general tracking techniques can be found in [71] [19].

Multi-target tracking approaches can be broadly separated into two categories: local methods that use information from past frames to estimate the current state recursively; and global methods that estimate the state based on a optimal association for all tracks within a temporal sliding window.

### 2.2.2 Local Methods

The earliest approaches [58] followed the recursive method, as single target tracking didn't require detection to target data association. Still utilised in modern tracking systems, local methods include Kalman Filtering [35] [57] which is highly susceptible to track switching, and Particle Filtering approaches [7] [38] [43] [21] which are able to handle more complex motion estimation. Benfold et al. [4] uphold online runtimes for high-definition sequences by utilising Kanade-Lucas-Tomasi tracking with a Markov-Chain Monte-Carlo Data Association technique.

### 2.2.3 Global Methods

Pirsiavash et al. [55] associates tracks by minimising a joint objective function which binds the detection likelihood with the track smoothness. The minimisation is performed using an iterative greedy shortest-path algorithm, where at each iteration a detection is assigned the best track. Once assigned, the detection and tracks are removed from the search space so they are not doubly assigned.

Milan et al. [48] formulate multi-target tracking as a continuous energy minimisation problem. Provided a set of detections for each frame, the tracker calculates target tracks by minimising an objective function:

$$E(\mathbf{X}) = E_{obs} + \alpha E_{dyn} + \beta E_{exc} + \gamma E_{per} + \delta E_{reg}, \tag{2.1}$$

where $\mathbf{X}$ is a set of tracks, $E_{obs}$ encourages tracks that align with detections; $E_{dyn}$, $E_{exc}$ and $E_{per}$ encode prior assumptions on trajectories that encourage smooth persistent trajectories with few collisions; and $E_{reg}$ is a regulariser that encourages a low number of trajectories. Figure 2.16 presents a graphical representation of the effects of each parameter. The minimisation function is highly non-convex with many local minima, necessitating the need for a heuristic scheme with repeated jump moves.

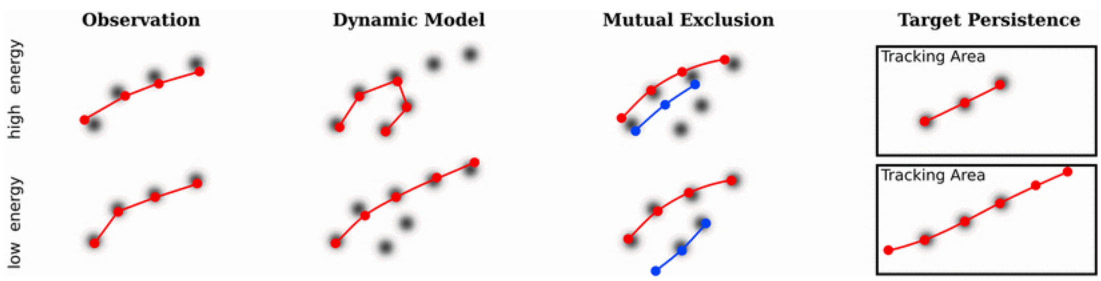



*Figure 2.16: The effects of different components of the energy function. The top row shows a configuration with a higher value for each term, whereas the bottom row shows the effects with a lower value for each individual term. Darker grey-values indicate higher target likelihood. Reproduced from [48].*

Later work by Milan et al. [49] [47] poses multi-target tracking as a discrete-continuous energy minimisation problem, in that association between detections and trajectories is kept discrete, while trajectory fitting is performed in a continuous domain to not restrict the state space. The method iteratively solves the discrete data association by $\alpha$-expansion, while continuously fitting continuous trajectories to the detections (Figure 2.17).

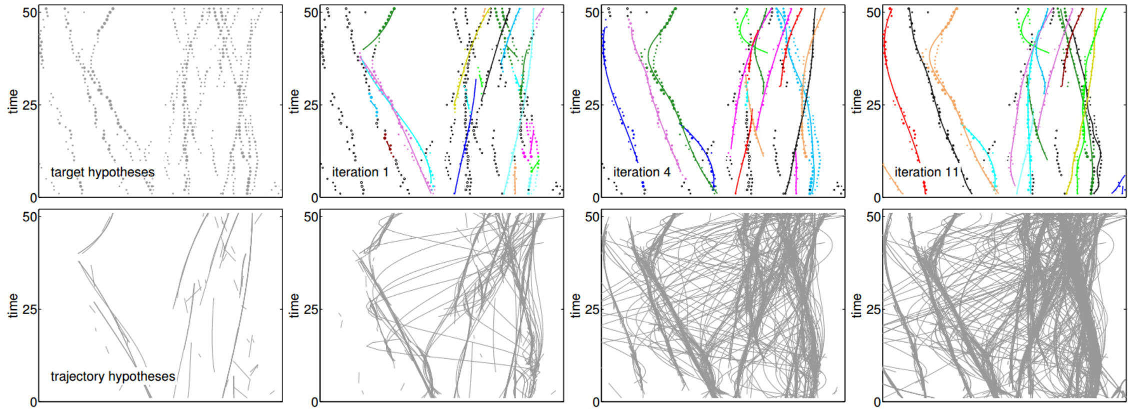



*Figure 2.17: Starting from a set of object detections and trajectory hypotheses (left column), the algorithm performs data association and trajectory estimation by alternating between solving a multi-labelling problem, and minimising a convex, continuous energy. The current set of trajectory hypotheses at each iteration is shown in the second row. Reproduced from [49].*

## 2.3 Detection & Tracking for Sports

Beyond pedestrian detection and tracking there has been some research into methods for sports such as soccer, basketball and hockey. TRACAB[2] is a commercially available application of sports tracking that has been deployed on sports including soccer, tennis, basketball and cricket.

One of the earliest publications involving sports tracking was by Nillius et al. [51], in which they propose the use of a track graph and group association tracking, wherein when two tracks cross and can't be disambiguated a new merged track is formed. The nodes in the graph denote tracks and the edges represent how tracks spilt and merge together. Then basing the problem as one of inference, they aim to find the most likely set of paths for the targets, given the appearance vector of such targets (Figure 2.18). The approach was applied to the case of a soccer match, with a static multi-camera panoramic system. This approach is much more feasible to soccer than AFL, as unlike AFL where highly congested packs often form with over ten players, soccer players are generally sparsely distributed with only short two-three player occlusion.

[2]http://chyronhego.com/sports-data/player-tracking

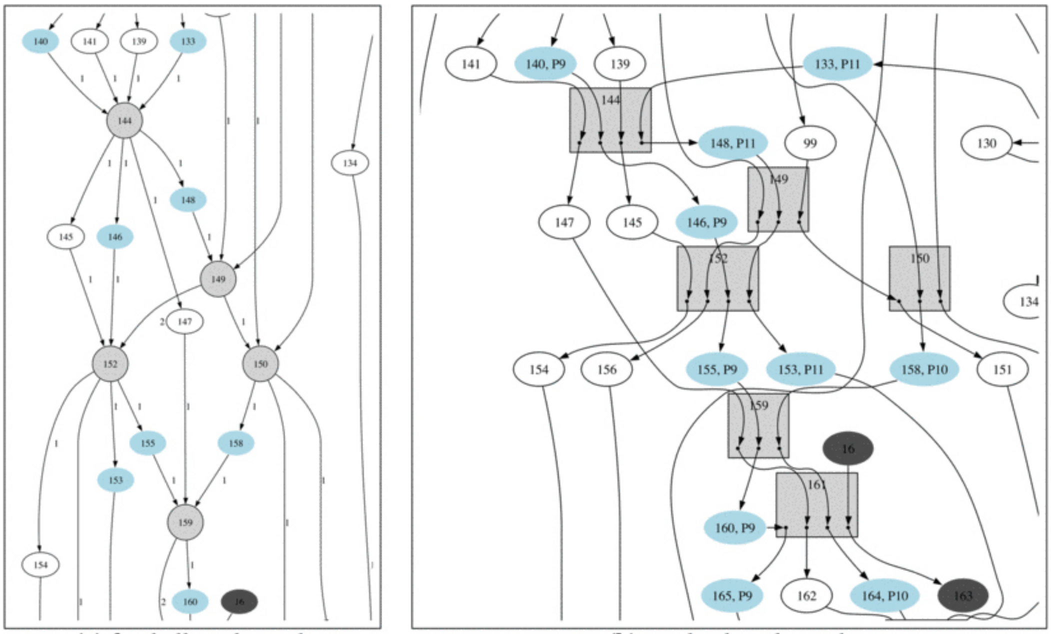



*Figure 2.18: (a) This is a small part of the soccer clip track graph. The node colours correspond to team A (light blue oval), team B (white), referees (dark grey) and multi-target nodes (black). (b) The corresponding resolved track graph. The square nodes display how the split nodes have been resolved. Ground truth player numbers can be seen for the team A players. Reproduced from [51].*

Okuma et al. [53] propose a self-learning framework which improves player localisation, allowing an unconstrained number of target objects to be tracked with non-static cameras. Their approach is novel in its self-learning approach to automatically perform the manual labelling process, using only a sparse set of weakly labelled frames. To classify interpolated detections a latent SVM is used with deformable part models representing a player's shape and colour (Figure 2.19), as well as player motion constraints. They tested their framework on broadcast footage of ice hockey matches, and basketball games, and were able to classify players based on team. The size of the AFL field in comparison to that of an ice hockey rink and basketball court suggests AFL players are much less likely to be captured at high enough resolutions necessary for the parts-based method. Also the lighting conditions in an outdoor AFL match environment are problematic for their method, especially in the team classification process.

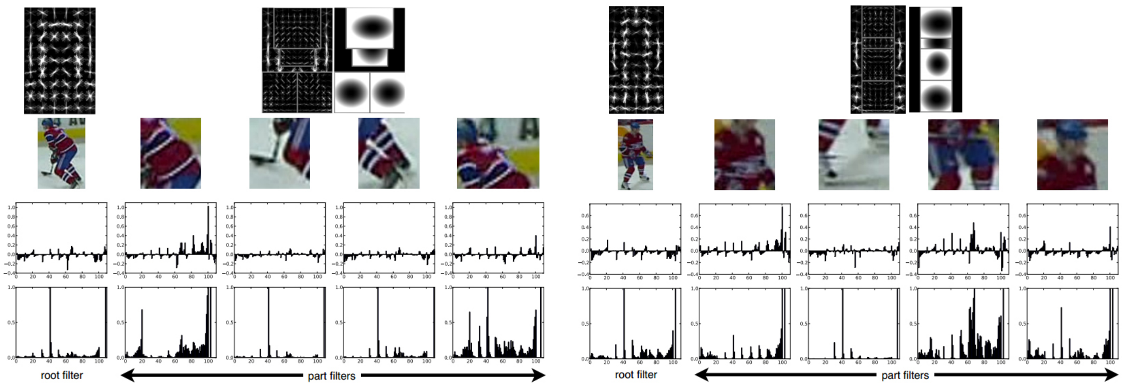



*Figure 2.19: This shows a mixture of two part-based colour models for one of the teams. For each model, the top row shows the root filter, part filters, and deformation model. The second row shows corresponding image regions of the object. The distribution of their learned weights and HSV colour histograms are shown respectively in the third and fourth row. Note noticeably higher weights on those parts that are particularly discriminative for classification. Reproduced from [53].*

Liu et al. [42] introduce a set of Game Context Features (GCF) to describe the current state of play, based on the expected player movements. Using the current track information in combination with the GCF they are able to select a simplified affinity model for each player at any time instant using a random decision forest. The context-conditioned motion models implicitly hold further complex inter-object correlations while still remaining traceable. The GCF are constructed from four match properties: global field occupancy; relative field occupancy; localised focal play areas; and player chasing directions. While applicable to sports, such a model is less effective for general pedestrian tracking as pedestrians are more random in their movement, generally having little to no correlation. Their approach was tested on field hockey and basketball match footage, showing improvements in tracking accuracy by 10% when using the GCF compared to not using them. This type of higher level information abstraction and utilisation may help the AFL case, however it is out of the scope of this project.

Hamid et al. [32], [31] propose an approach for robust localisation of soccer players using a set of cameras viewing the field from different angles. They set up a complete K-partite graph, with each partite corresponding to one of the K cameras. Nodes in a partite represent a player, including their position and appearance, and who is visible to that particular camera. Edges between player nodes are weighted based on player similarity between camera pairs, and their corresponding ground plane distances. Correspondences between players of different cameras are then modelled by K-length cycles (Figure 2.20). This is likely to be beneficial for many broader multi-camera applications, including AFL, however for this particular project it was unfeasible to setup cameras at multiple spots around the ground.

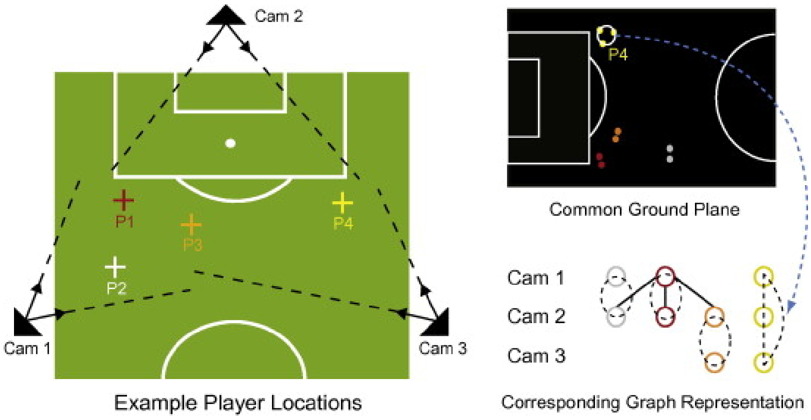



*Figure 2.20: Example player positions on a soccer field. Nodes in the corresponding K (=3) partite graph represent the player blobs detected in the three cameras projected to a common ground plane. In this graph, the dotted lines represent the minimum weight cycles, whereas the solid lines represent node edges. The weights of these edges are a function of the pair-wise appearance similarity of blobs and their corresponding ground plane distances. Reproduced from [31].*

## 2.4 Summary

Detection and tracking are core problems of computer vision, and the literature surrounding the problems is vast and diverse. Most recent approaches have been multi-target tracking-by-detection techniques, where an object detector is use to propose frame-wise target positions. There are a large number of different feature descriptors used for object detection, with the most widely used and applicable being histogram of orientated gradients [10]. Furthermore, there has been promising results with the combination of HOG with other features such as LBP [68] and HOF [67]. Two main machine learning classifiers are suggested, AdaBoost with a cascade and support vector machines, both of which have been shown to be effective. Modern multi-target tracking approaches have focused around global sliding window techniques to construct accurate tracks and associate detections tracks. The energy minimisation approach by Milan et al. [49] [47] is state-of-the-art and is well suited to handle difficult criss-crossing trajectories often found in the AFL scenario.

# 3. Overview and Preliminaries

## 3.1 Framework Overview

The AFL tracking framework implemented for this project is a tracking-by-detection approach, with a detection module followed by a tracking module, however a team classification module has been included in-between to provide more AFL specific information to the tracker (Figure 3.1). The entire pipeline is implemented in Matlab[1] because of its ease of use with vision research tasks, and also all of the external libraries used were written in Matlab.


*Figure 3.1: The AFL overall pipeline*

Each module in the pipeline has a specific task:

- The **detector** finds individual players in video frames and provides their frame-wise spatial position by defining bounding box coordinates and size for each player;
- The **team classifier** examines the contents of each of the bounding boxes and determines the team that the player in the box belongs to;
- The **tracker** uses the bounding box positions and team classifications to build paths for each player across the sequence (multiple frames).

Before work could commence on developing each module of the pipeline it was necessary to carry out some important preliminary work. AFL video sequences suitable for this work weren't available, so appropriate footage had to be captured over a number of weeks. The obtained footage could then be used to annotate ground truth training and testing data for the detector and team classifier stages.

## 3.2 Footage Capturing

It was important to obtain footage from AFL games which could be used for testing, as well as for building a diverse set of training data. Broadcast footage was not suitable for this project due to the large number of cuts, unknown camera positions, fast camera movement and other lack of control over the footage. It was therefore necessary to attend matches at Adelaide Oval and purposely film play with a number of static cameras on tripods. The playing field in AFL is oval shaped and reasonably

[1] http://www.mathworks.com.au/products/matlab/

large in comparison to the playing areas other sports, with the field at Adelaide Oval being 167m long and 124m wide[2]. Using five full high definition 25fps Axis Q1755 cameras[3] and a few tripods, all the cameras were set up on a rig in a corporate box at the top of the grandstand (Figure 3.2). With the resources available, all five cameras had to be setup at the one location, which was acceptable for this work, providing a single point-of-view overlooking the entire field. The elevation that the grandstand provides is beneficial for occlusion handling as it allows a greater ability to see over and behind players which, when viewed from ground level, would be occluded.

*Figure 3.2: The five cameras setup on rig with two tripods overlooking field.*

Cameras were kept static during capturing eliminating the necessity of the detector and tracker to compensate for camera movement and camera blur. Capturing was done in blocks coinciding with the game's quarters, with camera set ups being modified during quarter- and half-time breaks. A range of different camera set ups were experimented with using different zoom levels, orientations, and focusing on different parts of the ground. However, in many cases cameras were set up in a zoomed out horizontal panoramic like formation in an attempt to capture the entire field across all cameras (Figure 3.3).

[2]http://www.afc.com.au/news/2014-02-04/oval-retains-unique-size
[3]http://www.axis.com/products/cam_q1755/

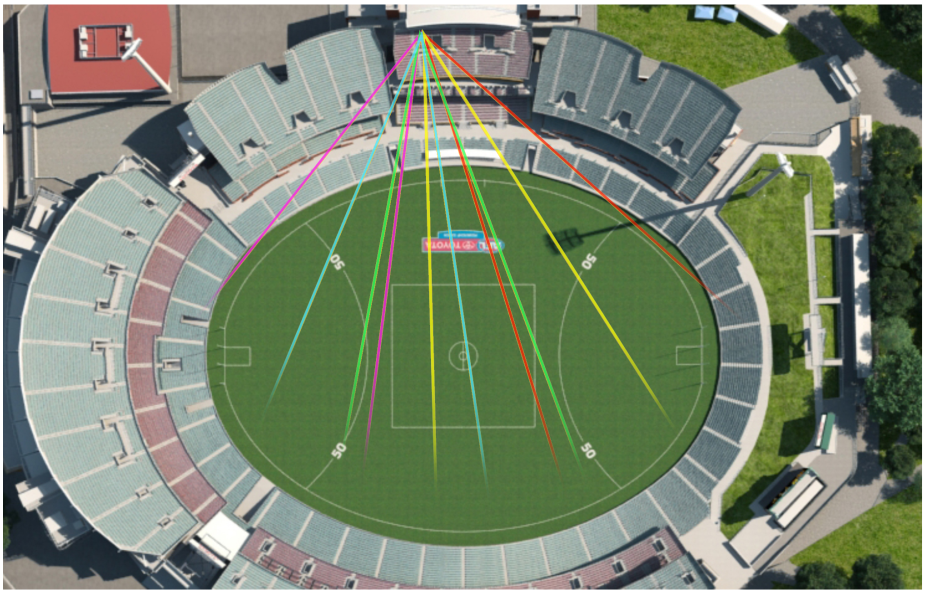


*Figure 3.3: Camera field of view representation over Adelaide Oval, the five cameras represented by the five coloured triangular shapes. They are generally aligned to overlap slightly to allow for the creation of a panoramic sequence of the entire field to be built.*

The effect of zooming out so far on such a large field meant players on the far side of the field could only be represented by a small number of pixels, with heights of around 40 pixels in the worst case (Figure 3.4). Lower resolution players have a lesser number of descriptive pixels and are hence more difficult to detect and classify, especially in crowded scenarios where players tend to visually merge and meld together. To achieve higher resolution results with the five cameras, some parts of matches were captured with zoomed in cameras, resulting in parts of the field not being covered, which is acceptable for particular testing and training sequences.

*Figure 3.4: The subset section shows the pixelation on the far side of the field.*

Another means of capturing, at a higher resolution, the far side of the field and the players located there, was to rotate the cameras onto their side. This meant 1920 pixels could capture the breadth of the field rather than 1080, almost doubling the resolution. As the detector on such footage performed poorly when compared to normal oriented footage, this method proved to be problematic. Upon closer inspection of the side-on orientated frames, it was found that the interlacing of the camera hardware caused undesirable vertical edges cutting off parts of horizontally moving players (Figure 3.5).

*Figure 3.5: Vertical frame problems: Caused by interlacing creating sharp vertical cuts.*

The capturing process was carried out on five separate matches across multiple weeks allowing for a variety of weather conditions to be captured, including sunny with shadows, as well as overcast and night footage under lights (See Appendix A). Unfortunately, over the five weeks there was there stormy and rainy weather, which based on observing broadcast footage, was expected to be one of the more interesting cases. In heavy rain play on the far side of the ground can be 'fogged out' by the rain, especially for zoomed out cameras. Also, in wet weather football there are many more and longer lasting pack scenarios, and players spend more time on the ground in extreme pose variations.

Over the five matches, approximately 50 hours of footage was accumulated, providing a broad and extensive range of AFL scenarios. The footage was split up into more manageable sequences and categorised for each match, quarter and camera. For example, a directory labelled `R04Q2C5` contains data from camera 5, in the 2nd quarter of the round 4 match. Smaller 30-90 second sequences that were particularly interesting were hand selected to be used as training and testing sets to highlight specific AFL scenarios. The focus of these scenarios were varying lighting conditions, varying camera zoom levels and orientations, as well as high levels of activity including different levels and lengths of occlusion with pack formations and, different speed and direction changes in play. The smaller scenarios were labelled with a three digit number with the first digit representing the match (`R03 = 0, R04 = 1, R07`

`= 2,..`) and the other two digits just the number of the dataset starting at `01`.

## 3.3 Ground Truth Annotation

To build the training and testing sets, it was necessary to manually annotate ground truth data by hand. Annotation is one of the slowest processes in the construction of reliable detection, classification and tracking systems. The process involves selecting a number of frames from a sequence and manually marking positive samples by drawing boxes around them. For this project, ground truth annotation was performed on the training sets chosen as mentioned in the previous section. Hence in some cases, for the shorter sequences every 25th frame (1 second) is captured and annotated highlighting short scenarios, and in the quarter long cases every 1000th frame is captured and annotated accounting for change over longer time periods. For the AFL ground truth data, boxes were drawn around all players and officials on the field during play. Each box was categorised and labelled based on a number of factors, such as team, occlusion and pose.

### 3.3.1 Team Identification Numbers

Each box was marked with an identification (ID) number (integer greater than 0) representative of the guernsey that the player in each box wore, representing the team the player belonged to. A team generally has two guernseys, one for home matches and another for away, and during capturing the Port Adelaide team wore two different styles of guernsey. Since the team classification stage of the framework is based on the guernsey appearance it is important to mark persons based on guernsey or uniform rather than team. Team IDs start from 10 and are in no particular order. IDs 1 and 2 are assigned to umpires and runners respectively, with the remaining 7 IDs (3-9) available for future usage if necessary. For each ID a colour, similar to that of the guernsey the ID represents, is applied to the box for an intuitive visualisation of the assigned ID (Figure 3.6).

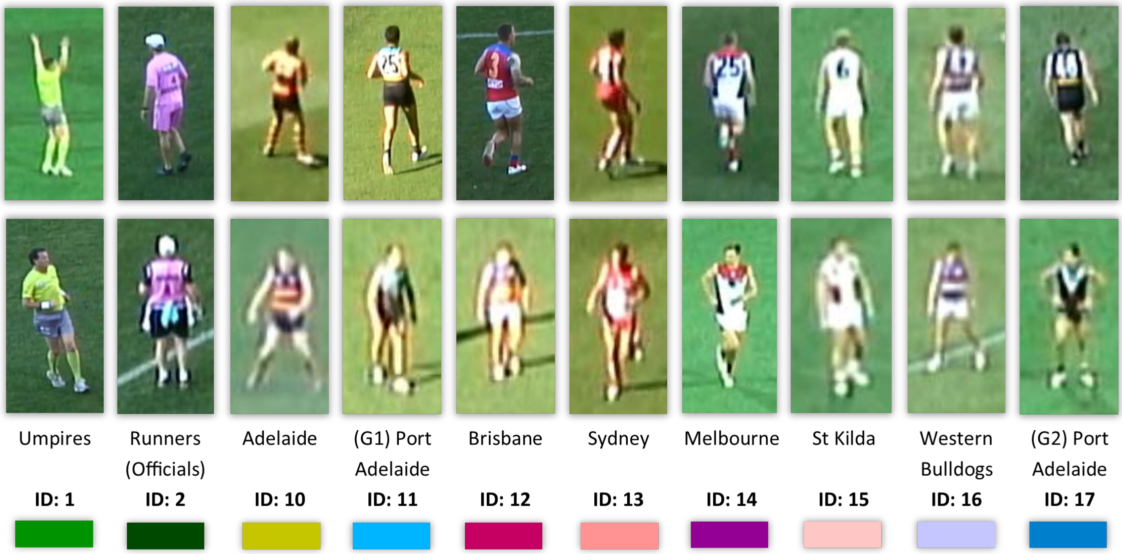



*Figure 3.6: Team examples, with team names, IDs, and classification/tracking colour (more teams exist in the AFL competition, only the above were captured).*

### 3.3.2 Occlusion and Pose Variations

During the annotation process a few cases were marked as specialised circumstances (Figure 3.7). Firstly samples that had the player occluded by another player were marked with an occlusion flag, `1` if they were occluded, `0` otherwise. For this work, occlusion is marked for a player $p_1$ when any part of any other player $p_2$ appears over the top of any part of $p_1$ such that the pixels for $p_1$ are noticeably different by eye. Occlusion isn't flagged when the player is occluding someone else ($p_2$), when only the sample boxes overlap, and for extremely minor cases where the pixels representing the occluded player $p_1$ aren't visually separable from the pixels representing the occluding player $p_2$ around the area of occlusion, such as a hand on an arm.

The other case which is marked as special is extreme pose variation, and is marked with the team ID `0` and corresponding colour black. There are a great variety of pose variation circumstances within the AFL problem domain, however many fit within the confines of a standing person box of ratio 1:2 (W:H). Extreme cases are marked where players are in more horizontal positions such as lying or crawling on the ground.

Players and officials which had parts of their bodies cut-off by frame edges were not marked as positive samples, so no box was drawn around them. In some cases where the majority of the body was visible, pixels that represent players may be considered as negative samples by the detector framework. However, it's expected this number of falsely labelled negative samples is a very small proportion of the correctly assigned negatives so there is minor effect on the detector's classification model.

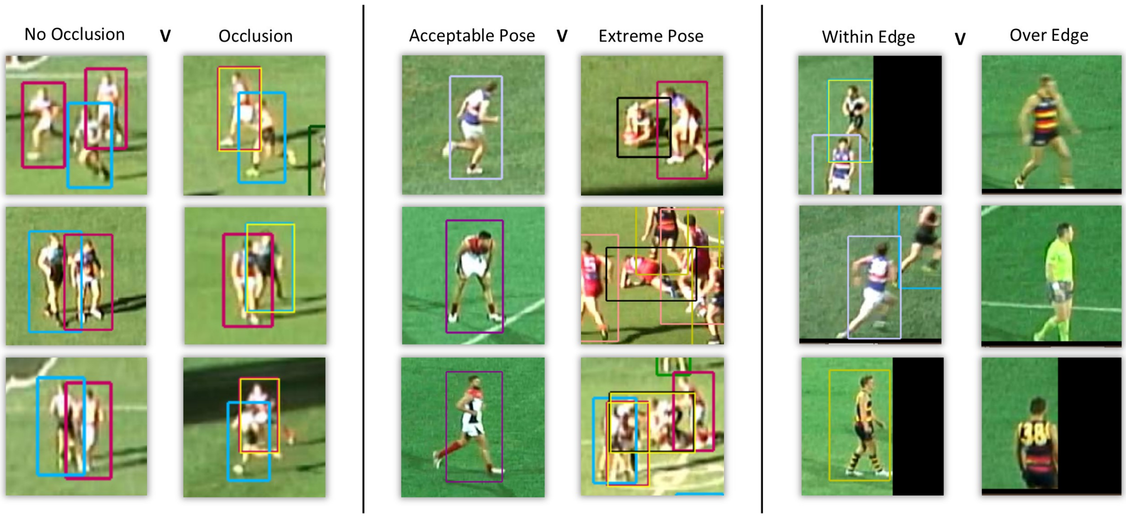



*Figure 3.7: Special case annotations: What's considered special case and what isn't.*

### 3.3.3 The Annotation Tool

The annotation tool used was a modified version of the `bbLabeler.m` code found in Dollár and Appel's Matlab vision toolbox [12]. Renamed `gtAn.m`, the following modifications were made to allow the software to be more usable for the annotation of AFL data:

- Added team ID functionality, including writing out the team IDs for each annotation in the output files.
- Bounding boxes drawn in team colour relating to team IDs. Such a visual cue makes classifying different teams and officials much less error prone.
- Changed the way to define bounding boxes, to increase speed by making first drawn position more accurate. The original method was to click either a top or bottom corner and then click the diagonally opposing corner position. Getting persons centred in the bounding box was difficult. The modified version is to click centre of bounding box at either top or bottom and just click again when the height is desirable, the box will automatically hold aspect ratio of 1:2. (Exception: When team ID is `0`, then bounding box aspect ratio is unrestricted and drawn in original way to account for extreme poses).

Although the annotation process is extremely tedious and time consuming, as will be seen in the next chapter (4), building quality training and testing datasets is a vital part for any good classifier and detector framework.

# 4. Detector Framework

The detector is the first, and most important stage of the tracking-by-detection pipeline with the tracking results directly reliant on reasonably accurate detection bounding boxes (Figure 4.1). Also, in this work, the team classifier is also reliant on accurate detection boxes centred correctly on players. The role of the detector is to examine individual frames separately and mark the positions of every player or official that is visible. This chapter describes the implementation of the detector module, presents the experimental results and discusses the use of different detector models.


*Figure 4.1: The AFL overall pipeline. The detector is the first stage of the process.*

## 4.1 Implementation

The detection framework implemented in the project was that of Dollár and Appel's [12]. The state-of-the-art approach utilises a cascade of classifiers approach with an AdaBoost learning algorithm. A Matlab implementation of the framework was available in their vision toolbox[1] and included HOG [10] as the default feature. For this reason, as well as the fact that it is the most widely used and well researched feature, HOG was chosen as the main feature.

The local binary pattern feature [52] has also achieved relatively good results for pedestrian detection problems [33] and when used in combination with HOG it can improve a detector's accuracy [68]. Using an implementation of LBP modified from the VLFeat library[2], another feature channel was added to the detection framework, so the combination of HOG+LBP could be used and compared with sole HOG. Each of the HOG and LBP features were calculated, normalised and then concatenated together to create one descriptor vector.

In sequences where the camera is zoomed out covering a large area of the field, much of the frame is taken up by crowd areas. Crowd and grandstand areas were considered as an unnecessary complexity for the system, with all focus being the within the field of play. Individual masks were drawn by hand for each of the camera angles and applied to frames before they were examined by the detector, ensuring no

[1] http://vision.ucsd.edu/~pdollar/toolbox/doc/
[2] http://www.vlfeat.org/

detections, nor tracks would be considered outside the field of play (Figure 4.2).

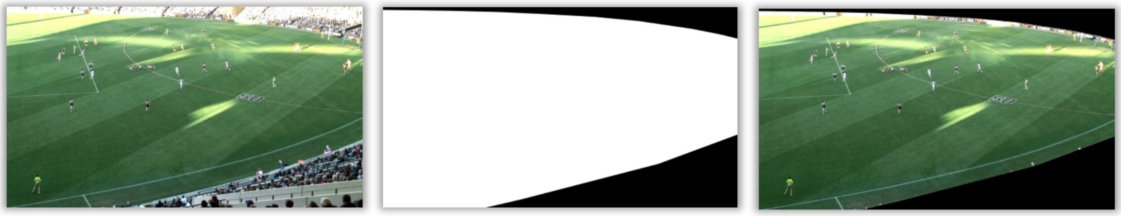



*Figure 4.2: Camera angle and associated mask, which then gets applied before frames are examined by the detector.*

During the first stage of the project detectors were iteratively trained and evaluated on a range of testing data, each modified with the goal of improving results each iteration. Over 30 different detectors each varying in factors such as the number of bootstrapping rounds, number of positive and negative samples, inclusion of positive samples marked with occlusion, feature choice, and other small parameter changes were evaluated and compared.

Bootstrapping is a means of obtaining a good set of hard negatives [61] during the training process. Bootstrapping involves running the classifier on a new image or sequence and adding all of the false positives to the negative training set. Doing this for one or more iterations should eliminate the likely false positives. Walk et al. [67] note that the number of bootstrapping rounds is a key component to a detector's performance with at least two rounds necessary for Dalal and Triggs' [10] HOG with linear SVM to achieve its full performance.

## 4.2 Evaluation Methodology

To compare different detector models in a quantifiable manner, techniques used for previous pedestrian detection and tracking evaluations were performed. The techniques involve direct comparison of a detector's output against a manually labelled ground-truth of a particular test set.

There are two main options for detector evaluation, full-image evaluation and per-window evaluation. Full-image evaluation, as outlined in [17], compares the overlap of the bounding boxes from the detector's output $BB_{op}$ and the ground-truth bounding boxes $BB_{gt}$ against some overlap threshold (Equation 4.1). Each $BB_{op}$ and $BB_{gt}$ may only be matched once, with any association ambiguity being resolved greedily, i.e. higher confidence matches are carried out first. Unmatched $BB_{op}$ are counted as false positives and unmatched $BB_{gt}$ as false negatives.

$$a_o \doteq \frac{area(BB_{op} \cap BB_{gt})}{area(BB_{op} \cup BB_{gt})} > 0.5. \tag{4.1}$$

Rather than looking at a comparison image as a whole, per-window comparisons are based on the cropped positive and negative classifications. That is, each ground-truth bounding box is extracted, passed through the classifier and evaluated to see if it is correctly classified positive. Similarly for negatives, areas of the ground-truth frames that aren't marked positive are passed to the classifier and evaluated as to whether they were correctly classed negatives. Per-window evaluation is more useful for specifically evaluating the individual classifier rather than the entire detector as a system. It has been shown that full-image and per-window evaluations are only slightly correlated [17]. The use of full-image evaluation was employed in this project as it covers the the entire detector, including the merging of close boxes.

Detector evaluations were measured by assessing sensitivity, precision, miss rate (false negative rate) and false positive rate. Sensitivity, also known as recall, refers to the ratio of true detections or tracks that were correctly found by the method, and precision corresponds to the ratio of outputted detections or tracks that were correct. These measures are generally visualized in one of two graphs, the detection error tradeoff (DET) curve [45], which plots the miss rate (FNR) versus the number of false positives per image (FPR) (fppi), and the precision-recall curve which plots precision versus recall (Figure 4.3). Both graphs rely on the spread of results based on varying accepting thresholds on the classifier (confidence thresholds).

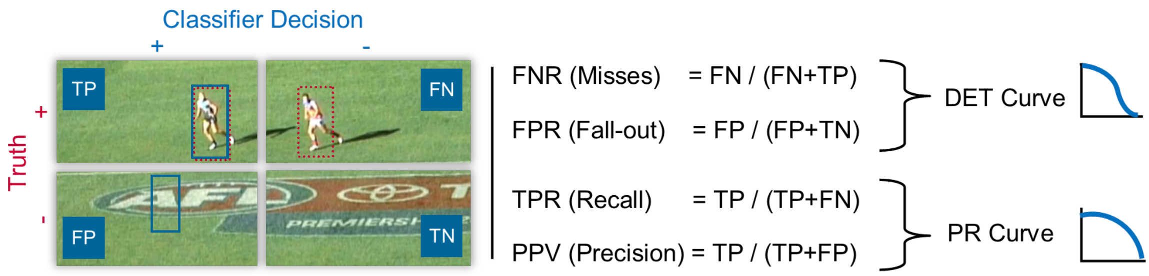



*Figure 4.3: The four possible classification results, True Positive, False Negative, False Positive, and True Negative, and there usage in building evaluation plots.*

With the vast amount of research on pedestrian detection and tracking, a number of classification benchmark training and testing sets have been made available, including INRIA [10], ETH [20], Caltech [16]. Each of the sets are unique, and have grown more complex and thorough in scenario variance and problematic situations. Evaluating detectors on such benchmark sets wasn't necessary, as the focus was specific to the AFL case, however it was important to use a range of different test sets which represented the general scenarios in AFL.

Although runtime wasn't the main priority of this project it is still an important factor, as such a system for a sporting application like football would be more beneficial if it ran online alongside the live match. Runtime analysis therefore was carried out on each module in the pipeline. All runtime analysis was performed on a 64-bit laptop running Windows 8 with a Intel i5-4200U CPU at 1.6GHz and 4GB of RAM.

## 4.3 Results

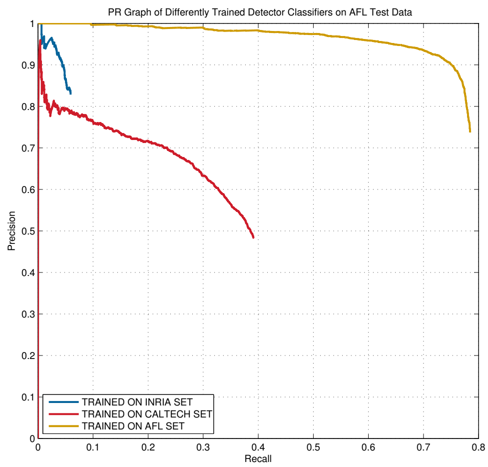
A number of different detectors have been evaluated using different features, training data, and testing data. The results are shown in the following subsections.

### 4.3.1 Different Training Sets

#### 4.3.1.1 INRIA, CALTECH, AFL

The importance and necessity for building an AFL training set and utilising it to build a particular AFL classifier within the detector is further highlighted in Figure 4.4. Two models, each trained on the INRIA training set [10] and CALTECH training set [16], were compared with a model trained on the hand-crafted AFL data. It is clear from the results that neither of the other models are suitable for the AFL case, with the AFL classifier achieving substantially better results on AFL test data. The INRIA and CALTECH models are inadequate for the AFL case for a number of reasons including:

- the pedestrians are in a completely different environment, with large amounts of background and foreground variability;
- the pedestrians can be occluded by much more than other pedestrians;
- the pedestrians are much more upright than in many AFL cases;
- the pedestrians are captured from side-on, where as in the AFL case the camera has some height;
- the pedestrians are in much simpler poses overall.



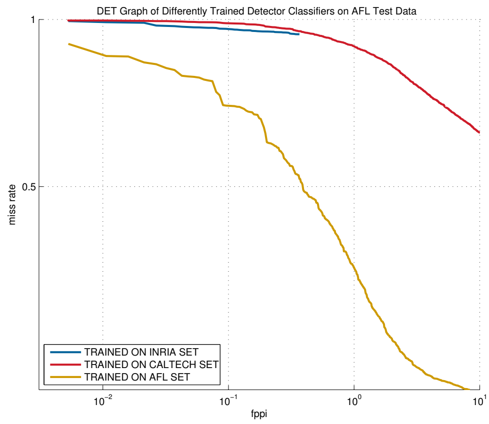




*Figure 4.4: PR and DET graphs comparing models trained on different datasets.*

Performing the opposite experiment and evaluating the AFL detector on the INRIA test set further highlights then need for specifically trained detectors. On the INRIA test set, the INRIA trained detector achieves a log-average miss rate of 12.93% but the AFL detector fails almost completely with a miss rate of 99.39%. In almost all test images in the INRIA dataset the AFL detector misses the person completely, and often in green noisy areas of the frame, such as grass and trees it generates many false positives (Figure 4.5). There is one image however that the AFL detector handles successfully, and that is of people playing soccer. The soccer image is very similar to the AFL data, with a mostly uniform background, similar pose situations, and the camera position at a similar elevated angle.

*Figure 4.5: The AFL detection results on some selected INRIA test images. Performs terribly on all cases except for the soccer image which is reflective of the AFL problem.*

#### 4.3.1.2 Occlusion versus Non-Occlusion

As previously mentioned, during the annotation process certain samples were flagged as being occluded or not. Two classification models were trained each with four rounds of bootstrapping, one with occluded samples included, one with them excluded. Figure 4.6 presents the results of the two classifiers and shows that excluding occluded samples provides a more accurate classifier. Further observation of the outputted detections shows that the classifier trained with occluded samples suggests a greater number of false positives in crowded and heavy occluded scenarios (Figure 4.7). This suggests the model build using occluded samples is much more prone to accepting occluded and busy areas, with a lessened ability to identify individual players over just areas with high pixel intensity variation.

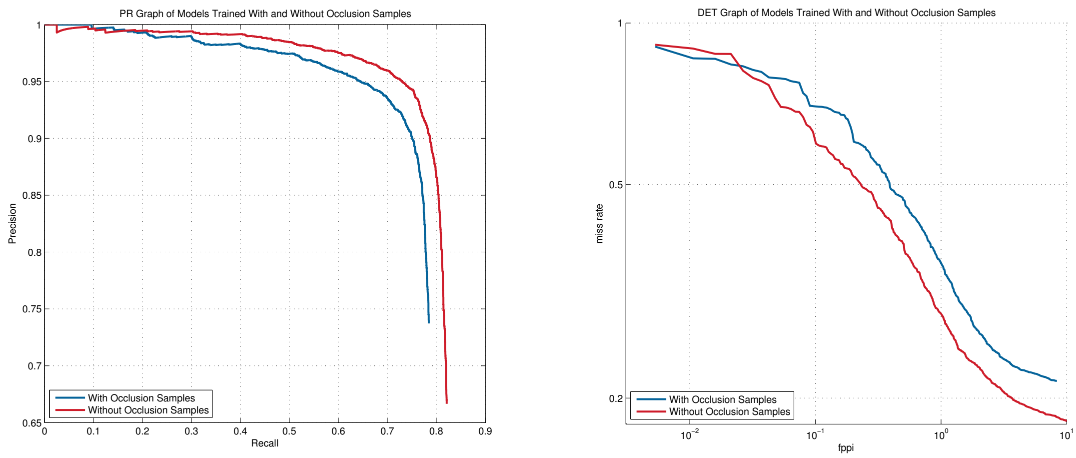
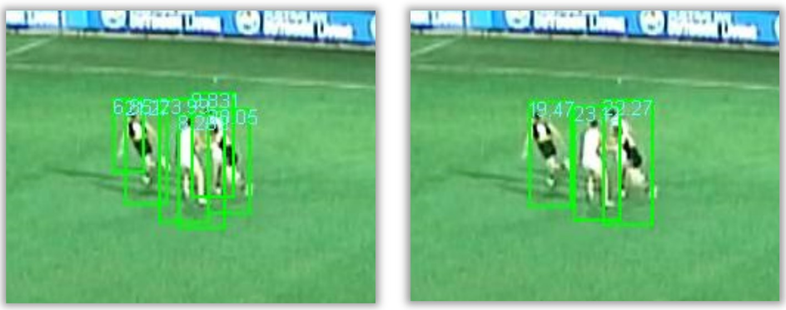



*Figure 4.6: PR and DET graphs comparing models trained with and without occluded samples.*



With Occlusion Samples    Without Occlusion Samples

*Figure 4.7: An example of too many false positives is crowded scenarios when using a detector with occluded samples included in the training data, compared to detector with them excluded.*

#### 4.3.1.3 Number of Negative Samples

Given all of the positive training samples were manually annotated ($\sim 12,000$) and the automatic means of choosing negatives from sample images where positives are not marked, it was interesting to see what the effect of altering the number of negative samples would have on the classifiers accuracy. As seen in Figure 4.8, for negative sample numbers greater than $57,000$ there was not much difference classification accuracy. In fact, the best performing classifier used approximately $180,000$ negative samples (150 from each test frame), with the other classifiers that used both more or less samples performing worse.

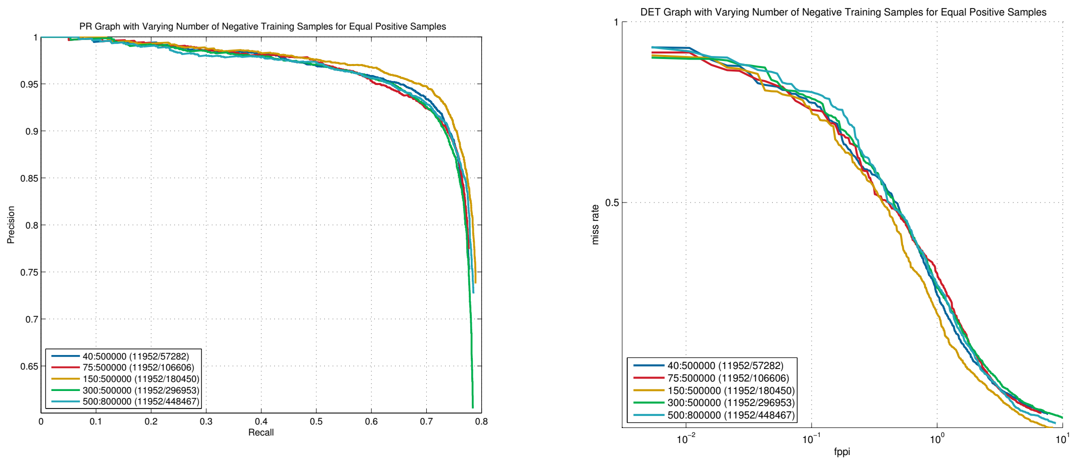



*Figure 4.8: PR and DET graphs comparing the effectiveness of different number of negative samples in the training data. Legend: # per image : max # (# of +'ves / # of -'ves).*

### 4.3.2 Feature Selection

Two main features were experimented with in the detector, sole HOG and a combination of HOG and LBP. It can be seen in Figure 4.9 that overall the HOG+LBP combination performs slightly worse than the sole HOG descriptor vector, however for recall values between 6.5 and 7.5 both feature choices perform equally well, with a slight edge to the HOG+LBP combination. This area of the curve is of most importance as it represents the best and most desirable outcomes for both precision and recall for the AFL application. Since the results are relatively equal, one could use sole HOG and the HOG+LBP feature combination interchangeably, however using the combination would increase the runtime excessively in comparison to any performance gain.

These results are a little counter-intuitive, with the combination expected to be more discriminant and hence more accurate than the sole HOG feature, as suggested in [33]. Further empirical comparison of individual frames from each classifier does not provide any further insight as the output images are from the range where the two classifiers are relatively equivalent. However it can be seen that the HOG+LBP combination does perform better on field signage suggesting greater discriminative power over sole HOG for patterns similar to, but not actually players. This additional discriminative power might suggest the very slim edge of the combination over sole HOG in the main part of the graph, however it is clear that overall both achieve equivalent results.

An explanation for the poorer performance of the HOG and LBP combination feature for lower recall may be that the combination is too specific and discriminative in a negative way. In lower recall cases where the classification thresholds are higher, meaning the positive samples being measured would be ones with higher confidence, the HOG+LBP combination lacks in precision. This suggests that the confidences with sole HOG are a better reflection of questionable detections compared to the addition of LBP. The HOG+LBP combination may set confidences for difficult and questionable pack and occluded samples too high in relation to other easier samples, whereas HOG being more generalised does not.

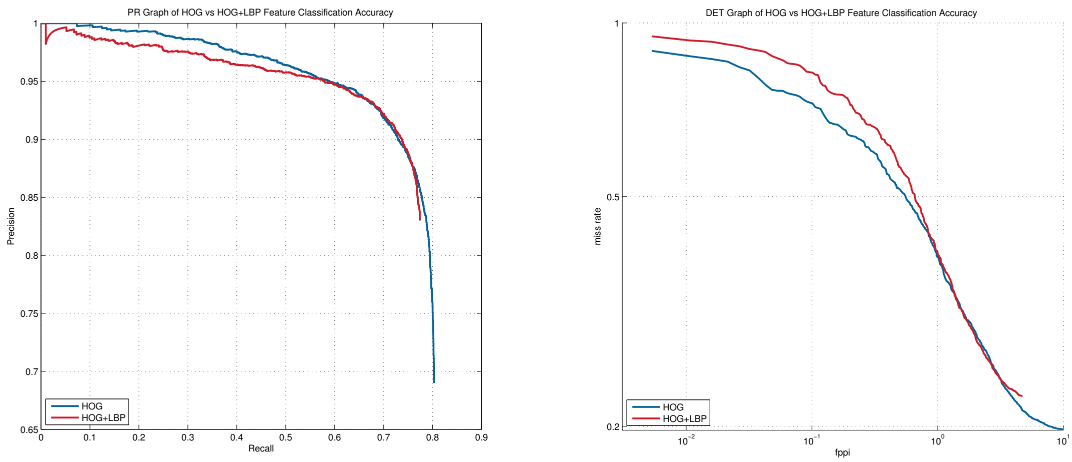



*Figure 4.9: PR and DET graphs comparing sole HOG against a HOG+LBP combination for the descriptor feature.*

### 4.3.3 Number of Bootstrapping Rounds

The number of bootstrapping rounds has been shown to alter the performance of classifiers [67], so it was important to understand the effects of changing the number of bootstrapping rounds for the AFL classifiers. The results again, shown in Figure 4.10, are counter-intuitive at first glance. It can be seen for this case that performing at least one round of bootstrapping is important to improving the accuracy of the classifier, however undertaking more than one round decreases the accuracy of the classifier. From further inspection of the output frames, despite bootstrapping for greater than one round decreasing the number of false positives on field signage, it also causes many positive partially occluded samples to be missed. This result is likely to be attributable to the combination of these classifiers being trained without occluded ground truth and the overlapping degree of 30% which was allowed between positive ground truth and randomly sampled negatives from the same frames. However as seen previously in Section 4.3.1.2, including the occluded ground truth samples on classifiers trained with three rounds of bootstrapping had a negative effect. Further refinement on both the classifier settings, such as positive-negative sample overlap, and the particular positive samples included in the occluded ground truth, may help strike a better balance between clearing up clear negatives like signage and maintaining individual detections in busy high occlusion areas.

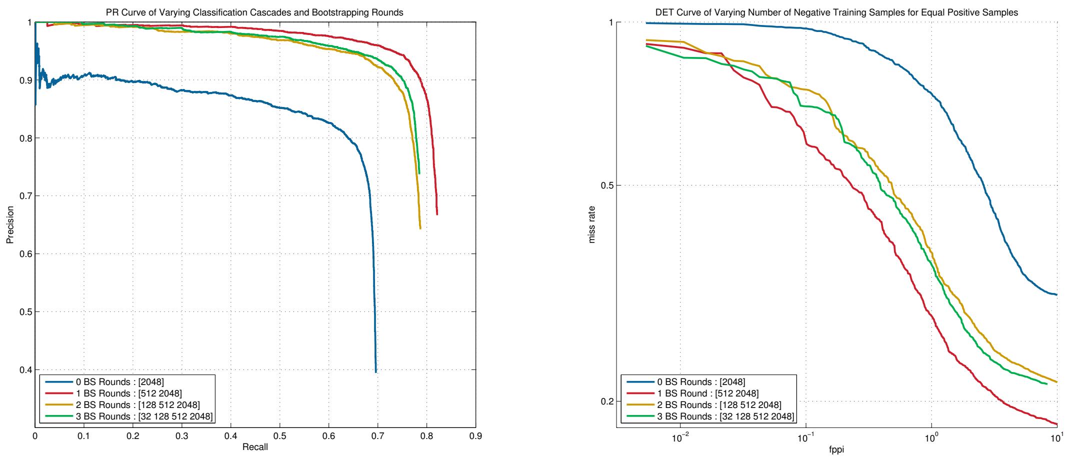



*Figure 4.10: PR and DET graphs comparing the effects of differing the number of bootstrapping rounds on the classifier model. Legend: [number of weak classifiers in the cascade at each stage, first to last].*

### 4.3.4 Runtime Analysis

The detector module is the fastest of all three of the modules, and has a runtime dependent on the number of sample windows evaluated in the cascade classifier. The number of windows is dependent on the frame dimensions, as well as the specified window padding and scaling steps, which for this work were all kept constant. Time can therefore be measured reasonably accurately on a frame per second (FPS) basis, that is, how many frames can the detector analyse within a second. Table 4.1 presents the runtime results for different datasets and varying frame totals. The frames per second value is consistantly around the $3 - 3.5$ mark, which is fast compared to the rest of the pipeline, yet still not sufficient for the entire system to be online.

| Dataset | Frames | Time | FPS | Frames | Time | FPS | Frames | Time | FPS |
|---|---|---|---|---|---|---|---|---|---|
| **R03Q3C5** | **250** | 85.73 | 2.92 | **500** | 151.31 | 3.30 | **1000** | 287.59 | 3.48 |
| **R04Q3C2** | **250** | 70.83 | 3.53 | **500** | 165.5 | 3.02 | **1000** | 288.24 | 3.47 |
| **R07Q4C2** | **250** | 90.07 | 2.78 | **500** | 140.27 | 3.56 | **1000** | 283.16 | 3.53 |
| **R12Q3C1** | **250** | 72 | 3.47 | **500** | 143.42 | 3.49 | **1000** | 282.81 | 3.54 |
| **R14Q4C4** | **250** | 81.7 | 3.06 | **500** | 137.75 | 3.63 | **1000** | 287.37 | 3.48 |

*Table 4.1: Runtimes of the detector module in seconds for varying numbers of frames. Shows constant frames per second (FPS) rate of around 3.*

## 4.4 Summary & Further Development

The detection module performs exceptionally well for a wide range of difficult scenarios including tough pose variation and illumination changes. It is vital that the detector's classifier be trained for the AFL scenario by using manually labelled AFL ground truth data, with off-the-shelf pedestrian trained detectors failing to achieve reliable performance. The experiments carried out for this project suggest the use of HOG and HOG+LBP are equivalent, however this is contradictory to what is found in other literature [33] [68], suggesting further testing is necessary. Using occluded samples in the training data decreases detection accuracy, as the detector is more accustomed to accepting 'messy' multi-person samples, classifying a large number of false positives in crowded areas. The optimal number of bootstrapping rounds necessary was only one, with further rounds restricting the classifier to miss more partially occluded samples. The detector has difficultly handling highly congested and occluded areas, however this problem is not constrained to the AFL case, with more general detection techniques also unable to handle difficult occlusion scenarios.

Further efforts could focus on using parts-based approaches with some recent detection methods using parts-based approaches to help achieve reliably accurate detections in the presence of pose variability and partial occlusions [53] [24]. Since, compared to regular pedestrians walking down a street, football players have a much wider set of potential poses, and often are in more tightly crowded occluding packs, parts-based approaches are likely to be of benefit. However it is likely that for a parts-based approach to be reliable, sequences would need to be captured at higher resolutions, ensuring players on the far side of the ground are represented by enough pixels to be split into smaller, yet still discriminate, parts. A parts-based approach could also allow for the detection of specific body parts in variable poses and occlusions enabling tighter focus on a player's torso and uniform to potentially provide more accurate team classification.

For many of the two player occlusion cases where players are standing or running beside one another, training a specific detector for this case much like the recent work of Tang et al. [63] which uses a joint double-single person detector, looks promising to clear up some of the ambiguity.

Optical flow, like that used in [36], would also be another good feature to investigate as players are generally the only movement on the field, other than the exception of some birds on occasion. This suggests the low-level use of motion in the detector is likely to increase the accuracy of the detector, however not all players move at all times, especially during breaks in play.

# 5. Team Classification Framework

AFL football is a two team sport, and the inclusion of a framework to automatically classify players into teams, as well as identify other persons on the field such as umpires and runners, is a natural and logical next step after the detector (Figure 5.1). The ability to identify persons based on team has benefits for higher level processes including tracking, for example, it would be a first step in individual person identification. The team classification framework follows a similar approach to that of a classifier within a general detector framework. Here, a classifier makes decisions for individual detections, classifying each into a team based on a descriptor vector constructed using feature extraction. This chapter describes the implementation of the team classification module and discusses different approaches based on experimental evaluation.


*Figure 5.1: The AFL overall pipeline. The addition of a team classifier for each detection provides more discriminant information to the tracker.*

## 5.1 Implementation

AFL teams, as well as umpires, runners, and other officials, all have their own specific uniforms, each comprised of certain colours and patterns. Variation in pose and low resolution for players on the far side of the ground leads to great variation in the appearance of team uniform patterns. For this reason, although colour isn't the most sophisticated feature, it was the most reliable discriminative feature available.

Using colour as the discriminative feature has its drawbacks, most notably the fact that most of the pixels in the detection box don't represent the player, with even less representing the uniform. In fact, it was estimated that the area of the detection box containing pixels representing the uniform can range from only 5%-15% (Figure 5.2). This suggested that some form of weighting to focus the descriptor vector on the uniform covered area of the bounding box was likely to be beneficial. Numerous different weights were evaluated, but all have a similar structure with the main focus on the upper middle section of the detection box, around where the uniform is expected to appear (Figure 5.3). Each weight is simply one or two 2D Gaussian functions:

$$f(x, y) = A \exp\left(-\left(\frac{(x-x_o)^2}{2\sigma_x^2} + \frac{(y-y_o)^2}{2\sigma_y^2}\right)\right) \tag{5.1}$$

with different values for the amplitude ($A$), the centre point $(x_o, y_o)$, and spread in each direction $(\sigma_x, \sigma_y)$.

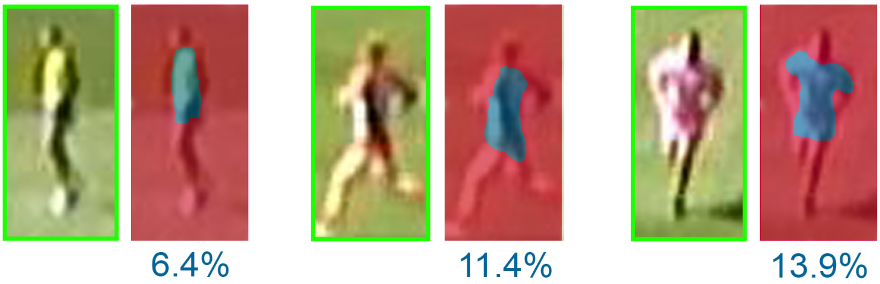



*Figure 5.2: Three examples of the percentage area size of the pixels representing uniforms relative to entire bounding boxes.*

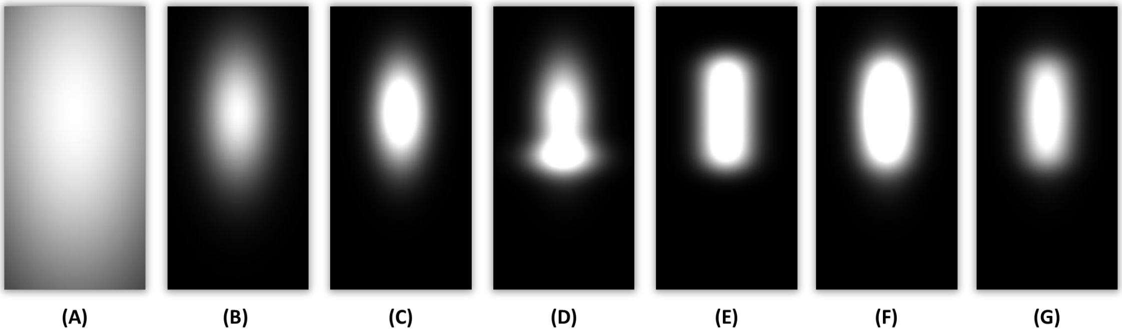



*Figure 5.3: The different spatial weights attempted in presented order left to righ.*

The descriptor vectors were constructed by histogramming the pixel intensities into 64 bins for each colour channel. Histogramming was used to lower the dimensionality of the descriptor vector and also to apply some form of smoothing over the individual colour values by grouping similar values together. Both the RGB and HSV colour formats were tested to see if either format performed better. Often HSV is used in colour classification tasks because it separates luma (image intensity), from chroma (colour information), generally providing greater discriminative power. When histogramming into the bins, the weights are applied based on pixel position, with that pixels likely refer to the guernsey having a larger impact on the histogram.

The decision classifier method employed was a support vector machine, because of their speed, simplicity and discriminative power. Each descriptor vector is plotted into a $b * c$ dimensional space, where $b$ is the number of bins, and $c$ is the number of colour channels $(64 * 3 = 192)$. The latest version of Matlab (2014a/b) comes with an SVM implementation built in, along with a number of different kernel options. SVM models were trained for each individual team, that is, for each model all training samples are labelled as negative except for those from one team. For comparison, two different approaches were used for training the SVM models. In the first approach, each team model was trained regardless of match, so samples collected from the same team across multiple weeks were labelled together. The second approach trained separate models for different matches as well as different teams, resulting in each match having a few possible team models to evaluate. For each test sample, a team is chosen by evaluating the sample with all of the other team models related to the match the sample is from. The model that produces the highest normalised classification confidence for the sample is selected as the team for that sample.

At first a basic linear kernel was attempted with the SVM classifiers, but some of these didn't converge within the maximum number of iterations (15000), hence a quadratic kernel was used which converged for all models.

## 5.2 Evaluation Methodology

Evaluation of the team classification is much like that used for the detector evaluation, but instead of testing overlap of detection boxes, team IDs are compared. The training and testing datasets were constructed by splitting all of the annotated ground truth data in half evenly across the different sets. Evaluations were then carried out for each detection, with each test sample being classified into a team and then compared to ground truth. Once again, precision-recall (PR) curves were used to visualise the performance of each classifier. Instead of detection error tradeoff (DET) curves however, more general Receiver Operating Characteristic (ROC) [22] curves are used instead. This makes more sense for the team classification task where the comparison between the TPR and FPR is more beneficial than the missed 'detections' and false positives per image metrics (Figure 5.4).

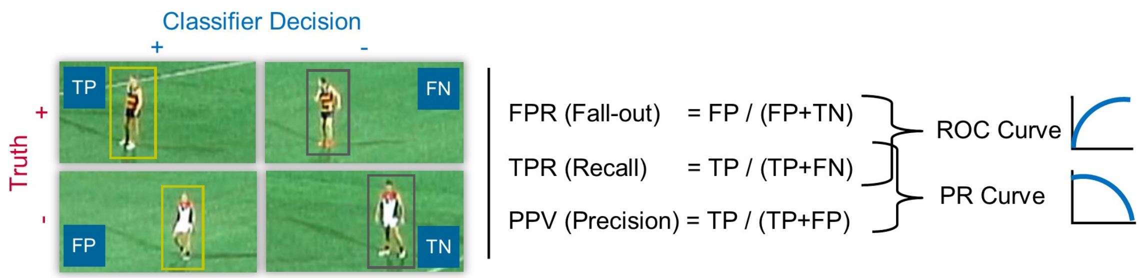



*Figure 5.4: Classification evaluation properties. In this case for the Adelaide team model (ID: 10).*

## 5.3 Experimental Results

The following sub-sections present and discuss the findings related to the team classification framework.

### 5.3.1 RGB versus HSV Colour Formats

Figure 5.5 presents a comparison of team classifier models trained and tested with samples RGB and HSV colour formats. As expected, the HSV colour format has significantly more discriminative power for the AFL scenario over all testing data. The separation of the value channel from the hue and saturation channels provides more robustness to lighting changes and shadows, with value changing while hue and saturation remain rather constant.

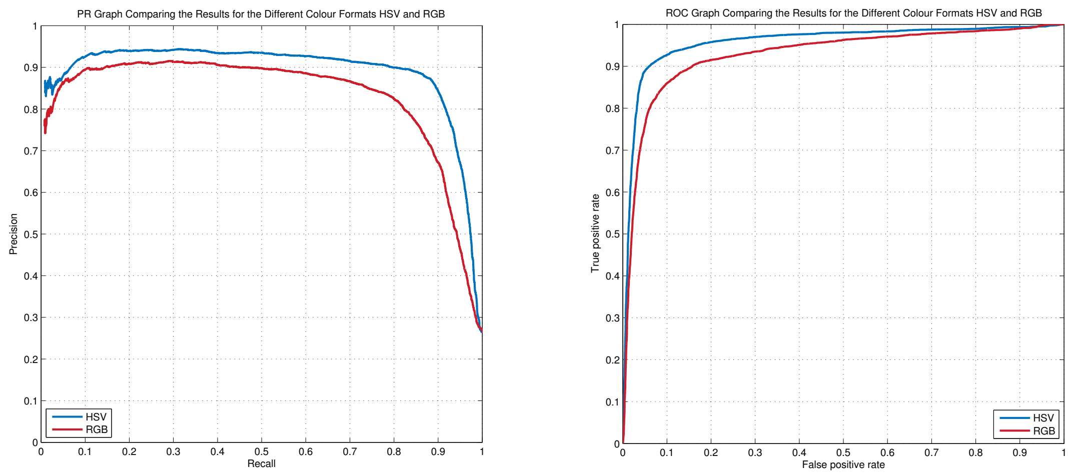



*Figure 5.5: PR and ROC graphs of team classification results for all teams combined for different colour formats RGB and HSV.*

### 5.3.2 Weight Maps

The different weight maps experimented with, and made reference to below, can be seen in Figure 5.3. Figure 5.6 presents a PR curve of the classification results of each of the different weight maps for all teams and testing data combined. The main notable outcome is that using a weight map centred around the guernsey area has a positive effect on the classification accuracy, with all weight maps achieving better results compared to no weightings. Weight map A is relatively poor compared to the other maps since it allows pixels nearer the edge of the bounding box to still have a reasonable and misleading effect on the classifier. The rest of the weight maps (B-G) are relatively similar in design and also results, with the best performing map being map C slightly edging out the similar map G. The other maps still perform reasonably well but appear to still weight parts of the samples which don't contain guernsey information too highly. Map D, where the weighting map also accounts for the shorts of players had a negative effect on the otherwise similar C map.

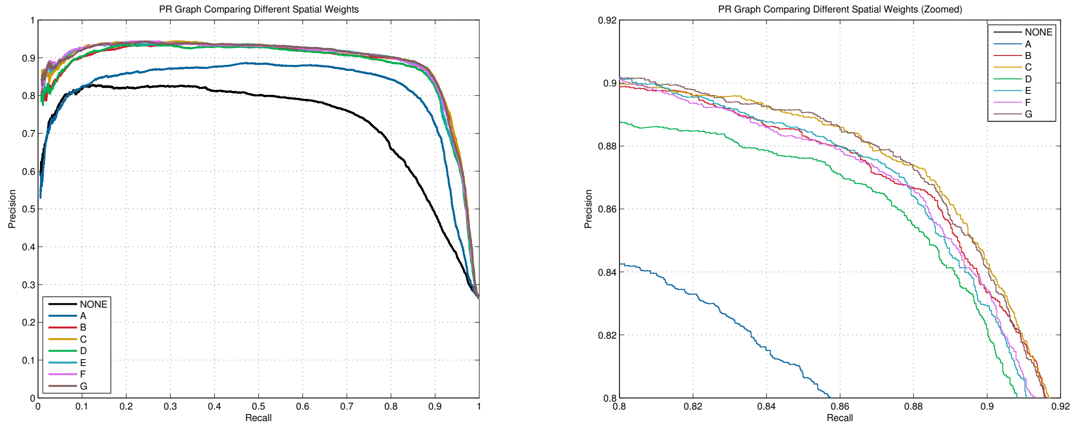



*Figure 5.6: PR and ROC graphs of team classification results with different spatial weights (left), with zoomed subset of figure (right).*

Figure 5.7 presents a comparison of using the weight map C with no weight map for individual teams, with results improving for all teams with the inclusion of the weight map.

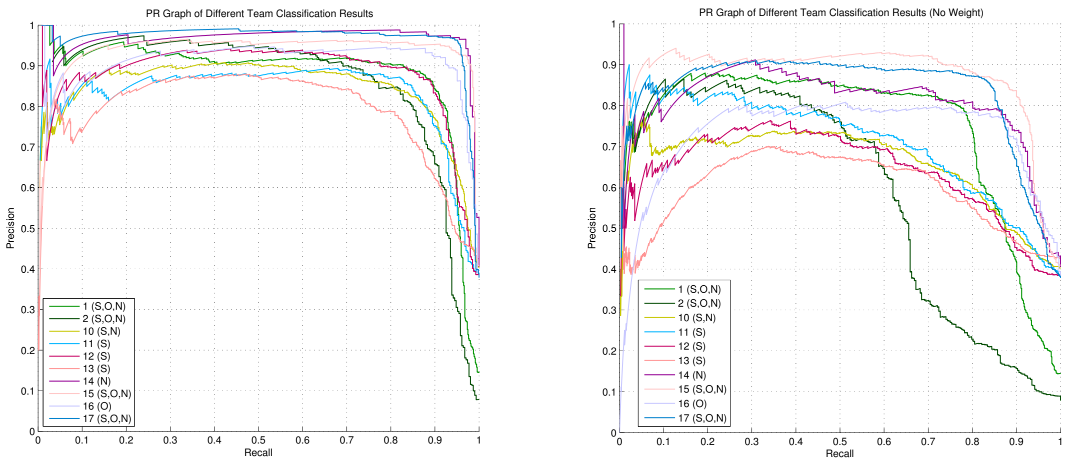



*Figure 5.7: PR and ROC graphs of team classification results with best weight C (left), and with no weight (right). Legend team IDs and capturing conditions: S=Sunny, O=Overcast, N=Night.*

### 5.3.3 Different Teams

Figure 5.8 presents the classification testing results for each of the individual teams. There is a difference in result between some of the teams, however it appears that the environmental conditions have more of an effect than the actual team guernsey. The four best classified teams had testing and training data from mostly night or overcast matches, whereas the bottom 5 had mostly sunny conditions. However, this doesn't mean different guernseys can all be classified to an equal degree, it just suggests that the overall environmental conditions have greater impact. Teams with greater contrasting colours, for example in the round 12 match with Port Adelaide (ID: 17) wearing mostly black played against St Kilda (ID: 15) wearing mostly white, are classified more accurately in comparison to team combinations that have similar guernseys.

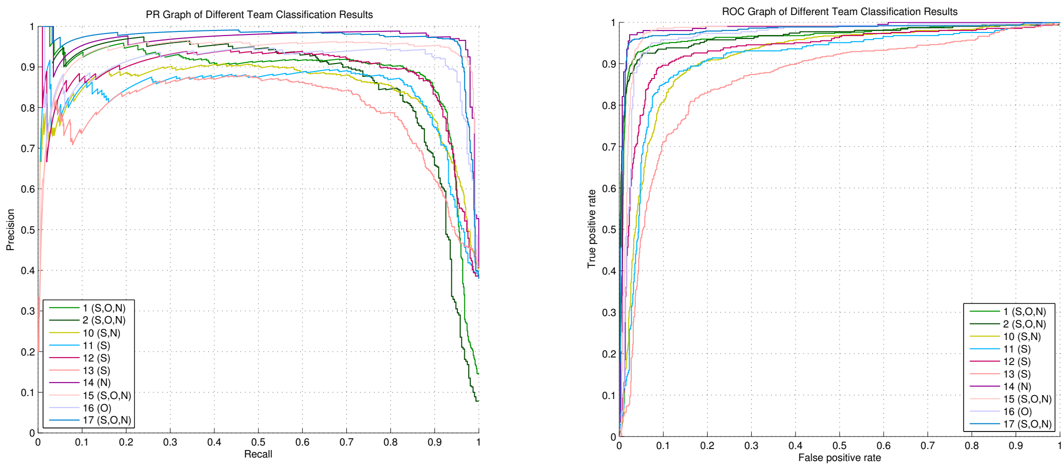



*Figure 5.8: PR and ROC graphs of team classification results for each team.*

### 5.3.4 Different Environmental Conditions

As was just previously identified, environmental conditions have the biggest impact on team classification accuracy. Figure 5.9 presents the classification results for sunny conditions compared with overcast and night conditions for all captured teams. It can be seen for all of the night captured teams, although there is some variation in accuracy, all classifiers perform better than any used during the day. A benefit of capturing footage from a number of different matches over a number of weeks has allowed for the Adelaide team, as well as umpires and runners, to be filmed under differing lighting conditions. Although this can be seen in Figure 5.9 with the double use of IDs 1,2 and 10, it is better visualised in Figure 5.10.

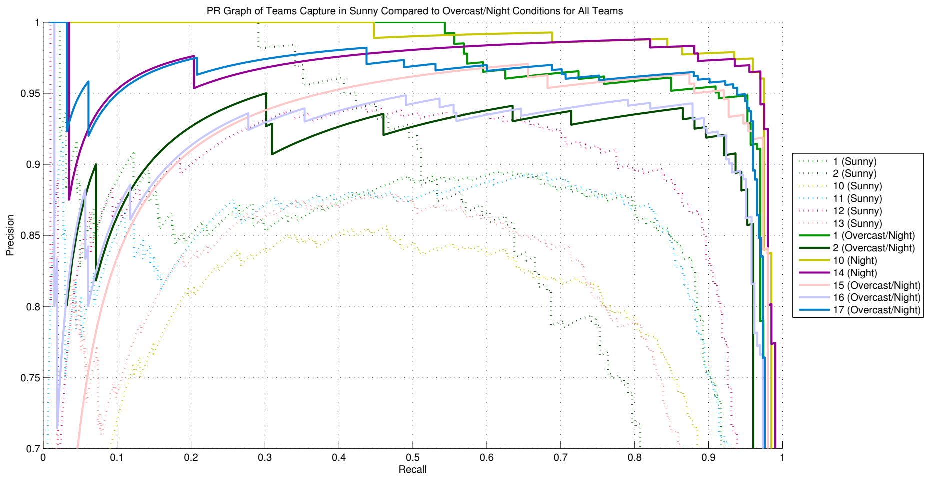



*Figure 5.9: PR graph of team classification results for all teams in different lighting conditions. Dotted lines are captures during sunny conditions, and the solid lines are captures from overcast and night conditions.*

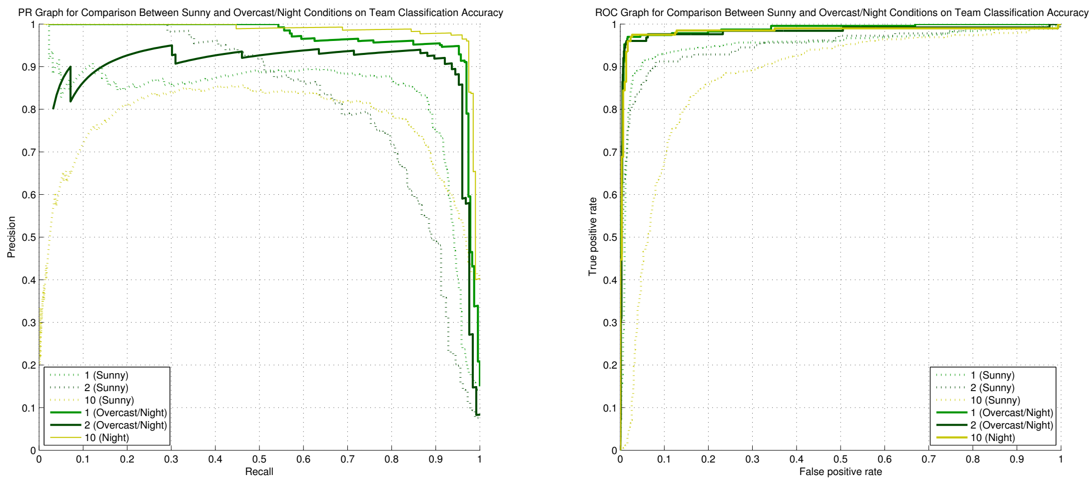



*Figure 5.10: PR and ROC graphs of team classification results for teams captured in both lighting conditions. Dotted lines are captures during sunny conditions, and the solid lines are captures from overcast and night conditions.*

### 5.3.5 Teams or Match Based Classifiers

The impact the environmental and lighting conditions have on the classification accuracy for any team is significant, and suggests that performance may improve with models tailored for specific lighting conditions. Experimenting with this hypothesis, two sets of models were trained. Firstly single team models used for all matches, and secondly individual team models for each scenario (game, quarter and camera). Figure 5.11 shows the results for these tests for all teams measured together. The scenario trained team classifiers, as expected, outperform single team classifiers further highlighting the importance of scene conditional classifiers.

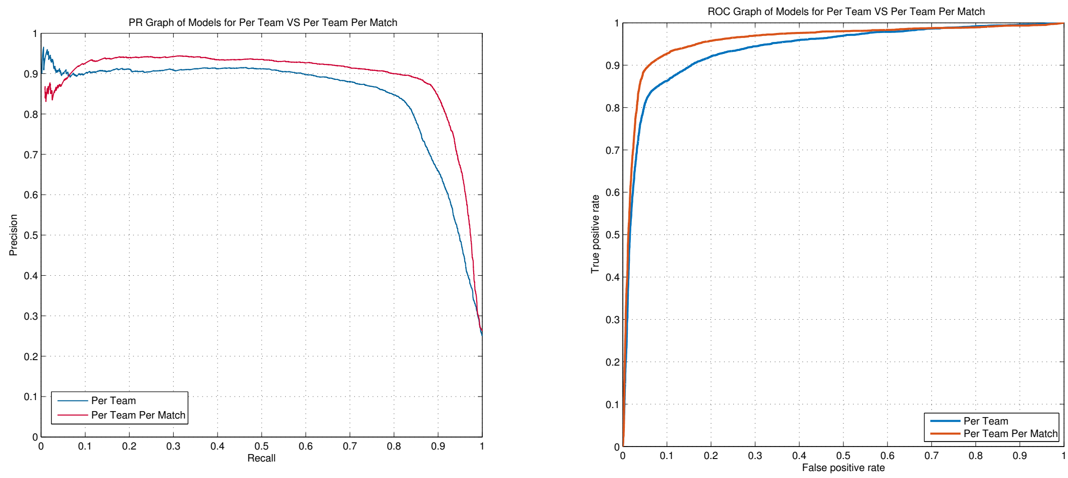



*Figure 5.11: PR and ROC graphs of comparison between classifiers for each team and classifiers for each team in each match.*

### 5.3.6 Runtime Analysis

The runtime of the team classifier can be measured separately for the feature extraction step and the SVM classification step. Unlike the detector, the team classification runtime is dependent on the number and size of the detections. As Table 5.1 presents, the feature extraction stage takes a substantial amount of time in comparison to the actual SVM classification, as well as the detector runtime. In AFL it is likely that there may be fifty or so detections to classify for any single frame, so with the feature extraction process only able to process between approximately 5 and 15 detections per second it wouldn't be able to process detections in real-time. The super-fast speed of SVMs is clear from the table with thousands of detections classified per second.

## 5.4 Summary & Further Development

Team classification based on colour using spatial weights and an SVM classifier works remarkably well. Transforming the RGB pixel intensities into the HSV format provides greater tolerance to varying lighting conditions, and the weight maps ensure players' uniforms have more significance in the quick SVM classifier decision. However it is clear that the framework benefits highly from specific models tuned and trained for specific environmental and lighting conditions.

The use of different models for a single frame is likely to further increase the accuracy of the team classifier, especially in sunny conditions where parts of the field

| Dataset | Detections | Feat. Time | Feat. DPS | SVM Time | SVM DPS |
|---|---|---|---|---|---|
| **R03Q3C5** | **6944** | 927.08 | 7.49 | 1.9 | 3654.74 |
| **R04Q3C2** | **5182** | 350.59 | 14.78 | 1.07 | 4842.99 |
| **R07Q4C2** | **1437** | 242.07 | 5.94 | 0.5 | 2874 |
| **R12Q3C1** | **6408** | 573.19 | 11.18 | 1.43 | 4481.12 |
| **R14Q4C4** | **1788** | 198.03 | 9.03 | 0.73 | 2449.315 |
| **R03Q3C5** | **13468** | 1805.44 | 7.46 | 3.28 | 4106.10 |
| **R04Q3C2** | **9655** | 714.81 | 13.51 | 1.88 | 5135.64 |
| **R07Q4C2** | **1817** | 245.11 | 7.41 | 0.39 | 4658.97 |
| **R12Q3C1** | **11918** | 1094 | 10.89 | 2.39 | 4986.61 |
| **R14Q4C4** | **3194** | 318.33 | 10.03 | 0.68 | 4697.06 |
| **R03Q3C5** | **25232** | 3112.65 | 8.11 | 5.12 | 4928.13 |
| **R04Q3C2** | **21737** | 1783.17 | 12.19 | 6.06 | 3586.96 |
| **R07Q4C2** | **2696** | 397.24 | 6.79 | 0.55 | 4901.82 |
| **R12Q3C1** | **22507** | 2078.79 | 10.83 | 4.5 | 5001.56 |
| **R14Q4C4** | **5840** | 585.48 | 9.97 | 1.14 | 5122.81 |

*Table 5.1: Runtimes of the feature extraction process and the SVM classification (in seconds) and the detections per second (DPS).*

are in bright sunshine while other parts are in dark shadow. The ability to classify a bounding box according to different environmental conditions would likely achieve better results, since it could allow for more specific team classification models, suited to the particular conditions, to be applied. A similar approach may be applicable to the detector module with different lighting and environmental conditions calling for different detection models, however as the detector classifier is gradient based and normalised, it is likely to be much less of a factor on detection performance (other than the extreme cases).

Further extension of the team classifier module could be as a verification tool for detections. As most correct detections have green grassy edges surrounding a player, the SVM models could also be trained to look for detections that are unlikely players. Then a threshold could be set on the detector module's and team classification module's confidences before being sent as detections into the tracker. Improving the colour histogramming and weighting process to run faster would also be of great benefit to this module.

# 6. Tracking Framework

The tracking framework is the last stage of the pipeline, joining detections and associated team classifications into team classified tracks for each player across the sequence (Figure 6.1). This chapter describes the tracking implementation and discusses various experimental results.


*Figure 6.1: The AFL overall pipeline. The final stage is the tracking framework which joins the detections across time.*

## 6.1 Implementation

The tracking framework utilised in this project was that of Milan et al. [47], which uses a discrete continuous energy minimisation technique to perform the data association and trajectory calculation. Their approach was shown to have some promising results on the challenging benchmarks PETS 2009/2010 [18] [25] and has some desirable properties for this project. The global approach explicitly handles partial and full inter-object occlusion, and has natural inclusion of per-frame detection evidence, appearance, dynamics, persistence, and collision avoidance. A Matlab implementation was also publicly available online for download[1].

The implementation available online was tuned for the intricacies of the AFL problem, and iteratively empirically tested. The tuning wasn't a straightforward process, with fifteen separate parameters available and little documentation of the purpose and effect of each. The purpose of a number of parameters were clear from the parameter name, including label and outlier costs which affect the number of tracks generated for fitting with detections. Talks with the author Milan resulted in no extra information about the unknown parameters, so each parameter was modified independently and effects empirically analysed. The greatest difficulty was finding the balance between having one track for each player while still maintaining two tracks for nearby and crossing players without erroneous merging. Lowering the pairwise cost parameter led to the better splitting of nearby targets into their own tracks, which was important for the AFL problem as players often run beside one another.

The tracker was modified to include the team classification results for each of the detections, allowing targets to be assigned a team. Targets were assigned the team that most of their associated detections were classified as. The assignment in this

[1] http://www.milanton.de/files/software/dctracking-v1.0.zip

case works especially well since more often than not detections are classified correctly, resulting in the false classifications being 'drowned' out. However, as will be discussed in the results, tracks often get switched with players regularly crossing paths. This erroneous behaviour can then cause team classification to be incorrect before or after a track has switched between players of opposing teams. Further development of the tracker might improve this problem, and with the incorporation of team classifications when deciding on tracks, such ambiguities will likely be resolved.

Early in the project a Kalman Filter tracking approach was also implemented as it is simple and local, providing a substantially different approach to the global minimisation approach. For a given frame the Kalman Filter algorithm estimates expected position of all current tracks based on track velocities estimated from past frames. Unassigned detections are associated to tracks greedily both spatially using Euclidean distance from the estimated positions and temporally using a temporal sliding window where the past frame has preference over two frames back and so on. If a detection doesn't get associated with a track, likely due to it being a new target entering the frame, or a previously lost target, or false positive detection, it is used to create a new track. At the end of the process tracks lasting for a small number of frames, expected to be false positive detections, are removed from the solution. At first the Kalman Filter approach was tried on its own with reasonably good results, later the energy minimisation approach was included to further refine the Kalman results, as well as being able to be utilised on its own for similar results.

## 6.2 Evaluation Methodology

During the modification and tuning of the tracker, evaluation was carried out empirically by analysing the output footage side-by-side for failures and successes. Only recently have quantitative evaluation methodologies for multi-target tracking been proposed and used in literature to benchmark methods. One technique, CLEAR MOT, was developed by Stiefelhagen et al. [60] [37] and is composed of two metrics, Multi-Object Tracking Accuracy (MOTA) and Multi-Object Tracking Precision (MOTP). An additional technique involves the Mostly Lost (ML) and Mostly Tracked (MT) scores that correspond to the number of tracks held for less than 20% and more than 80% of their life respectively. Milan et al. [46] present a study of such proposed metrics, and highlight the numerous difficulties in accurately evaluating multi-target tracking problems including the notion that multi-target tracking ground truth is not well defined. It is difficult to annotate precise locations for targets in such a continuous space, meaning metrics need to be tolerant to such inprecision.

The AFL problem is too disparate from past pedestrian problems to be directly comparable. Additionally, the results from the different implemented tracking approaches used in this project are able to be compared empirically. It was therefore not essential to perform quantitative evaluations for this project, however in further developments it is likely to be an included procedure.

## 6.3 Experimental Results

### 6.3.1 Kalman Filter VS Energy Minimisation VS Combination

The Kalman Filter on its own performed well in having only one track per target at any one time. However it failed often in circumstances where the detection boxes became more noisy and spatially spread over time, such as when players moved too fast and also when boxes disappear or describe multiple players often occurring with occlusion. These factors meant that the Kalman Filter approach regularly terminated and initiated relatively short tracks for targets. Using the global energy minimisation approach to refine the Kalman Filter provided better length tracks, with many of the short tracks being merged into longer continuous tracks. There was also better recovery from occlusion, holding tracks across frames where detections disappear temporarily. The energy minimisation method was also able to be utilised on its own, however in congested parts of the scene many false positives would 'float' erratically over the busy area locking on various detections representing many different players. This erroneous behaviour is partly attributed to the tracker having difficulty distinguishing close targets, and also partly to the very noisy detections resultant from the detectors inability to handle highly congested areas.

Figure 6.2 is an example case of some of the scenarios just mentioned for each variation of the tracker. At position (a) in the energy minimisation approach the erratic and unrealistic switching and sliding of tracks can be seen, however with the use of the initial Kalman Filter solution these problematic cases don't arise and form part of the combined solution. At position (b) in the Kalman Filter approach the track for the darker player on the right has only just been initialised as that player just passed behind the umpire to his left in previous frames, causing his track to be terminated and re-initialised. In the combined tracker the two shorter tracks get joined into a longer track that exists constantly as the player passes behind the umpire. The players above and beside position (c) present the general effect of the initial Kalman Filter solution on the energy minimisation approach. The Kalman Filter solution restricts one or no track to players at any point in time, which guides the energy minimisation approach, which often uses one or more tracks for each target at any one time, to do the same.

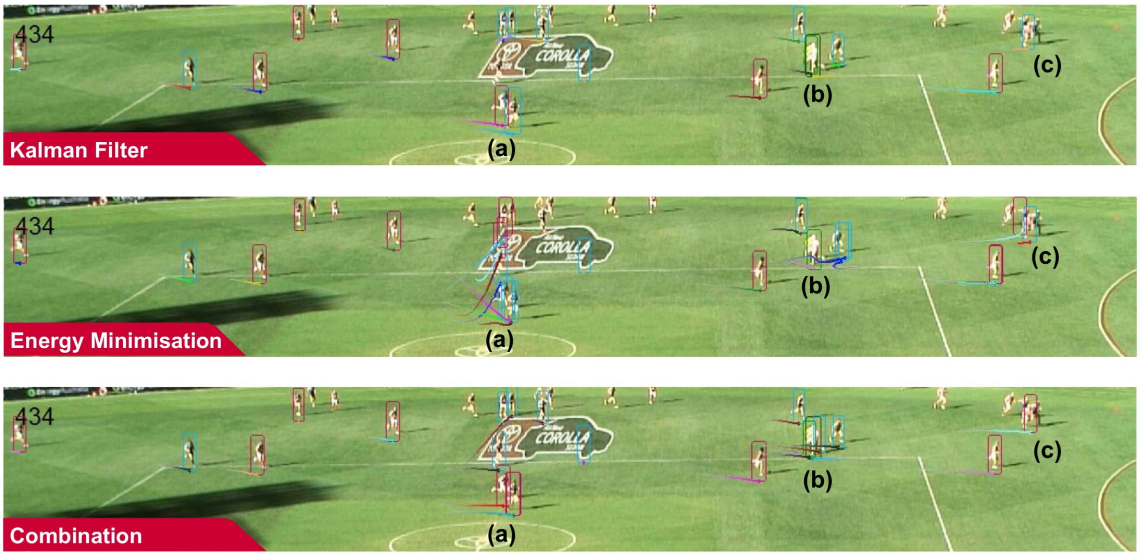



*Figure 6.2: Empirical comparison between frames from the local Kalman Filter approach, the global energy minimisation approach, and the combination of using the latter to refine the former.*

### 6.3.2 Parameter Tuning

As previously mentioned, parameters were tuned by logically guided trial and error, with each independently modified and evaluated. The final tracking configuration contains five modified parameters which penalise high numbers of tracks and attempts to separate close tracks. Figure 6.3 provides comparisons between frames from the original configuration with frames from the tuned configuration. The effects of the parameter tuning are minor and dependent on the particular scenario. For the case below there are many less false positives. With use of the Kalman Filter approach as the initial solution, the effects are even lessened.

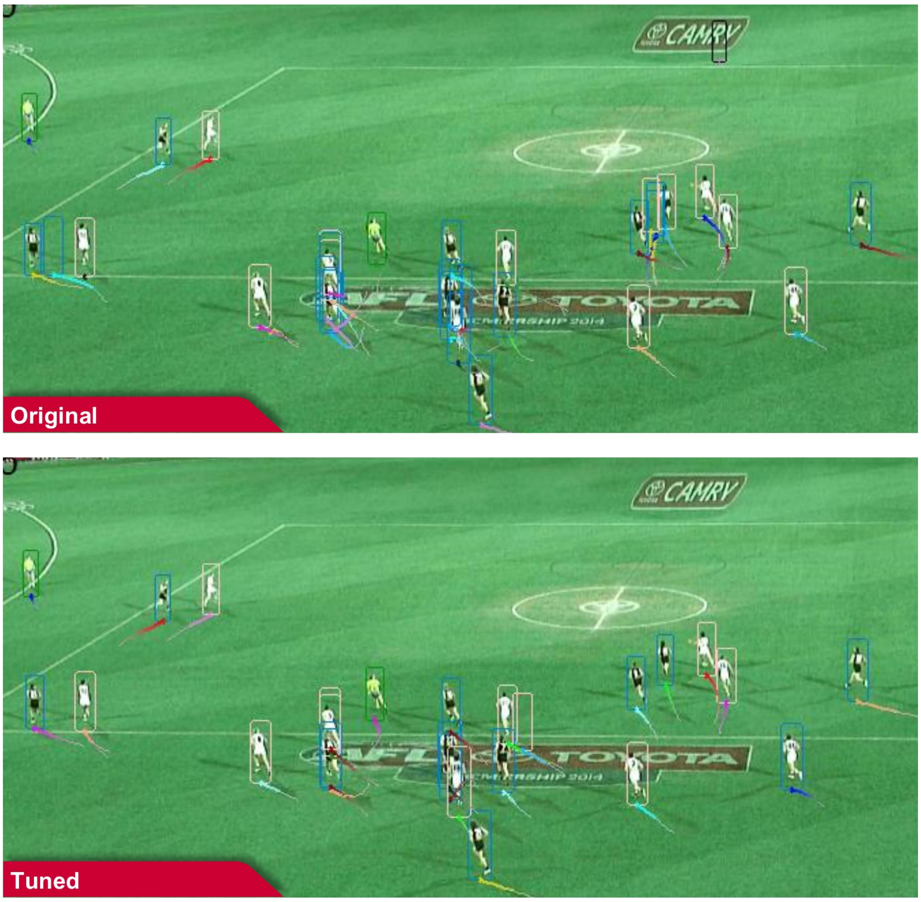



*Figure 6.3: Empirical comparison between frames from the original configuration against frames from the tuned tracker configuration.*

### 6.3.3 Runtime Analysis

The tracking runtime was dependent on which tracking approaches were utilised, the Kalman Filter or the energy minimisation approach or both. It is also dependent on the length of the sequence and the number of detections in a sequence. The tracking runtimes are relatively long compared to those of detection, and even classification for long sequences with over 10, 000 detections (Table 6.1). The Kalman Filter approach is faster than the energy minimisation approach for the results below, however it appears that as the number of frames increases the closer the Kalman Filter runtime gets to the energy minimisation. This behaviour is likely to be attributed to the Kalman Filters increasing the number of tracks needed to be estimated and compared over time, where as the energy minimisation approach handles less tracks overall, slowly building the ones it does have. The result also shows a key benefit of using the Kalman Filter to devise an initial solution, in that it lowers the time necessary in the energy minimisation process, and also in some circumstances, finds a more accurate solution in lesser time overall when compared to energy minimisation utilised on its own.

| Dataset | Frames | Detections | Kalman Filter | Energy Min. | KF + EM Comb. |
|---|---|---|---|---|---|
| **R14Q4C4** | **250** | 1788 | 16.77 | 109.9 | 16.77 + 70.95 |
| **R14Q4C4** | **500** | 3194 | 75.34 | 234.63 | 75.34 + 97.37 |
| **R14Q4C4** | **1000** | 5840 | 294.75 | 415.98 | 294.75 + 224.09 |
| **R07Q4C2** | **250** | 1437 | 14.3 | 50.24 | 14.3 + 47.04 |
| **R07Q4C2** | **500** | 1817 | 24.6 | 74.16 | 24.6 + 83.5 |
| **R07Q4C2** | **1000** | 2696 | 70.93 | 115.91 | 70.93 + 75.59 |
| **R03Q3C5** | **250** | 6944 | 409.61 | 1163.07 | 409.61 + 329.27 |
| **R03Q3C5** | **500** | 13468 | 2090.81 | 4056.33 | 2090.81 + 3153.54 |
| **R03Q3C5** | **1000** | 25232 | 8702.88 | 14306.6 | 8702.88 + 15449.75 |
| **R04Q3C2** | **250** | 5182 | 82.29 | 409.11 | 82.29 + 298.54 |
| **R04Q3C2** | **500** | 9655 | 565.63 | 1349.39 | 565.63 + 537.1 |
| **R04Q3C2** | **1000** | 21737 | 3459.66 | 8226.86 | 3459.66 + 2987.44 |

*Table 6.1: Runtimes (in seconds) of the three tracking procedures: Kalman Filter, energy minimisation, combination of both.*

## 6.4 Summary & Further Development

Two tracking approaches, a local Kalman Filter approach, and a global energy minimisation approach, were utilised in the tracking module. Using either solely on its own had its problems. The Kalman Filter was unable to handle noisy detections, occlusions and fast target speed and direction changes, however it was able to maintain one track per target at any point in time. The energy minimisation approach while better suited to handling the noisy, occluded and fast changing scenarios, often suggested too many false positives for targets, especially in areas of high occlusion. The global approach was also susceptible to generating unrealistic erratically moving tracks. The combination of the two approaches, using the energy minimisation approach to refine the Kalman Filter solution, was able to provide a middle-ground which was more realistic and accurate. As was the case for the detector with training for the AFL problem domain, the energy minimisation tracking technique also benefited from being tuned for the particular intricacies of the AFL scenario. The benefits of the tuning however are minor in comparison, with limited knowledge of what each parameter contributed, it is likely that the tuning is far from optimal.

The combined approach still has difficulty handling highly crowded areas, but this is related back to the reliance of the tracker on the detections provided by the detector, which is likely to provide noisy and inaccurate detections in such cases. It is likely necessary in such high density areas where occlusion is constant for multiple players, that the detector and tracker adapt to detecting and tracking a group rather than trying to differentiate individuals, then re-identifying individuals once the group has dispersed.

Both implemented approaches were slow for longer sequences, with runtimes on a single machine becoming unreasonable very quickly. Kalman Filter approaches can generally be implemented for online usage, suggesting the particular implementation used in this work can be rewritten for much faster performance. Experimenting with other tracking approaches such as a more complex Kalman Filter method may provide better solutions. Restricting any global methods to a short window of time is likely to be necessary, however doing so would decrease their effectiveness at finding the overall most accurate solution.

# 7. Conclusion

This work investigated detection and tracking methods for the sporting application of AFL football, a yet to be researched scenario. The AFL scenario brought about a particular set of unique challenges that require more general methods to be refined and improved.

Off-the-shelf detectors, trained on pedestrian datasets such as INRIA and CALTECH, were unusable for AFL footage, with detectors needing to be trained with specific AFL data. Using approximately 12,000 positive and 180,000 negative samples, a cascade detection framework utilising AdaBoost and HOG was able to perform relatively well for a first attempt at the problem. The improved detector is able to handle difficult pose and lighting variations, which in more generalised pedestrian systems would likely cause erroneous behaviour. Further pose variation, such as players lying or kneeling, will need to be handled by secondary detectors trained specifically for those cases. The extreme exposure problems are considered to be more of a hardware limitation, with more recent camera sensor technology likely to overcome many of the exposure problems, easing pressure on the software. Any remaining exposure challenges may require further detection model training or, if too extreme, separate models. Occlusion remains the most difficult case to account for, with the detector being susceptible to partial and full occlusions. A possible solution for partial occlusions could be special double-person detectors like that of Tang et al. [63]. Another solution for both partial and full occlusions, particular to this application, could be using cameras placed around the entire stadium, each capturing the field at different angles. Beyond these suggestions, occlusion handling is a problem that needs to be handled in the tracking framework.

Using a spatially weighted HSV colour histogram feature extraction process, with a linear SVM classification approach resulted in exceptionally reliable team classification results. The use of colour however results in the process being highly susceptible to exposure variations and occlusion. It is necessary to train very specific team classification models for different environmental conditions, suggesting a necessity to train and apply different models for different parts of the field and frame in many circumstances such as sunny conditions. Occlusion problems are almost completely overcome in the tracker with knowledge of previous team classification results for a particular target earlier in time.

The global energy minimisation tracking approach was improved for the particularities of the AFL problem by tuning some parameters, however it is still susceptible to noisy and erroneous detections. The most notable complication of the tracker is its inability to handle occlusion and player crossover, often incorrectly switching tracks between targets as well as generating many false positive tracks which move erratically over crowded areas. Implementing a simpler local Kalman Filter tracking approach to build an initial solution, which was then refined by the global optimisation was able to correct many of the erratic false positives while maintaining longer tracks that include difficult speed and direction changes of targets. Adapting the tracking framework further to enable it to follow groups of players rather than individuals when occlusions occur, and then resolving individual tracks when groups split, may allow some of the incorrect occlusion behaviour to be rectified.

By incrementally experimenting, refining and evaluating different detection, classification and tracking approaches, this project has achieved some viable and promising results for all stages throughout the pipeline. The current methods work remarkably well and handle many difficult AFL situations not common in generalised pedestrian tracking cases. However there is still room for improvement and future work, especially in reducing the runtime of the modules and experimentation with other tracking frameworks.

In this project a well tuned visual tracking framework, specifically for the problem of tracking AFL football players, has been devised and implemented. This framework provides a good foundation for future development of more refined solutions and of higher level information abstraction processes, such as those necessary for statistics and match analysis.

# A. Captured Datasets

## A.1 Round 3 : Saturday, April 05, 1:40PM, Adelaide VS Sydney


*Figure A.1: The four quarters filmed with the five cameras for the round 3 Adelaide VS Sydney match*

## A.2 Round 4 : Saturday, April 12, 1:40PM, Port Adelaide VS Brisbane

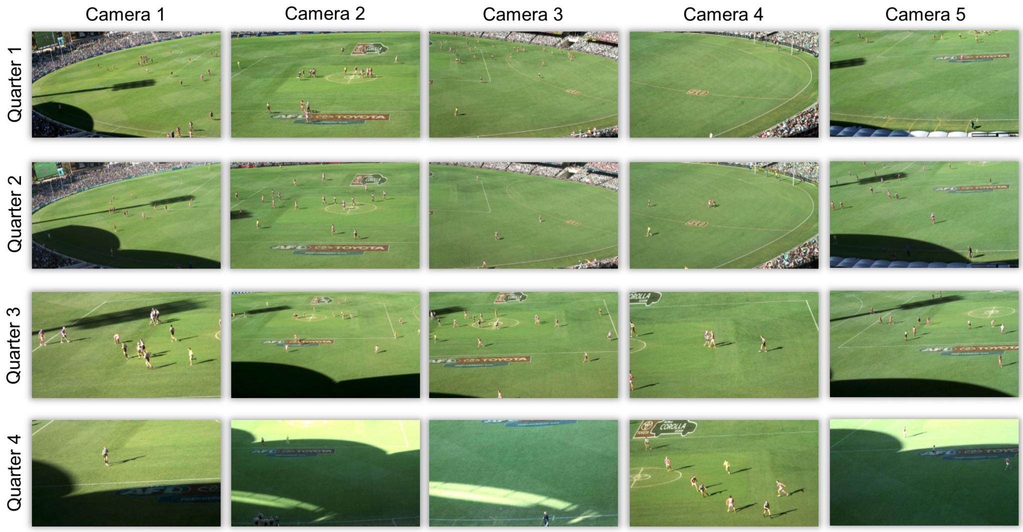



*Figure A.2: The four quarters filmed with the five cameras for the round 4 Port Adelaide VS Brisbane match*

## A.3 Round 7 : Saturday, May 03, 4:10PM, Adelaide VS Melbourne

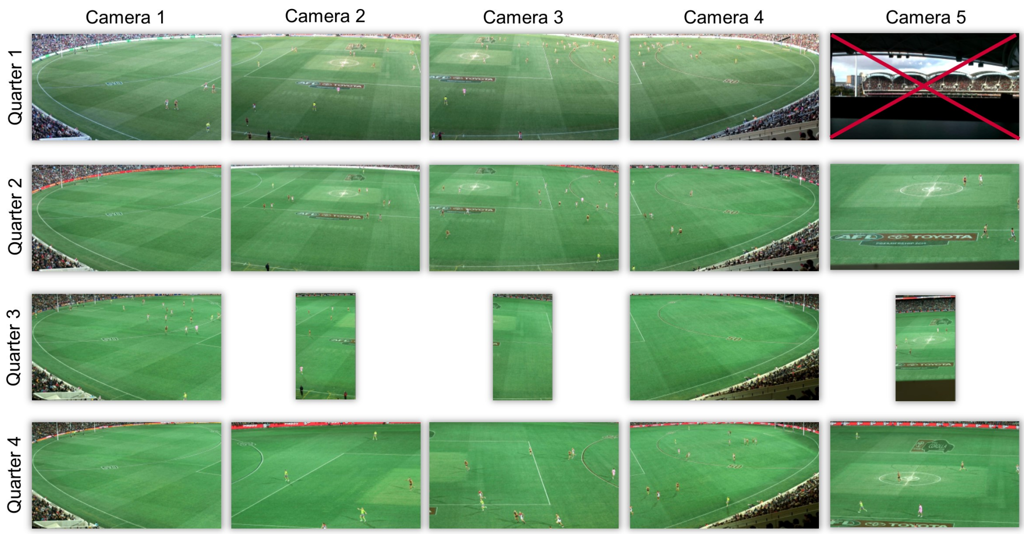



*Figure A.3: The four quarters filmed with the five cameras for the round 7 Adelaide VS Melbourne match*

## A.4 Round 12 : Saturday, June 07, 4:10PM, Port Adelaide VS St Kilda

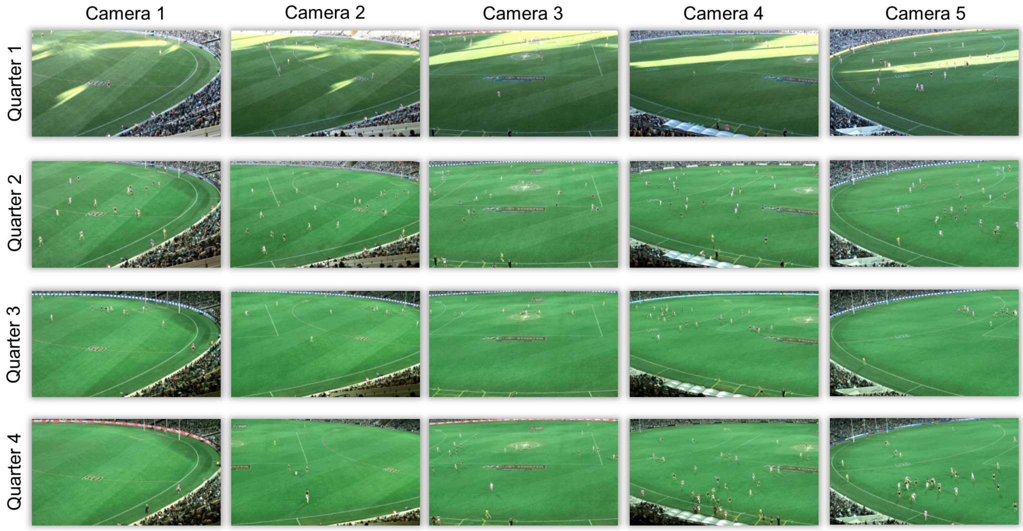



*Figure A.4: The four quarters filmed with the five cameras for the round 12 Port Adelaide VS St Kilda match*

## A.5 Round 14 : Saturday, June 21, 1:15PM, Port Adelaide VS Western Bulldogs

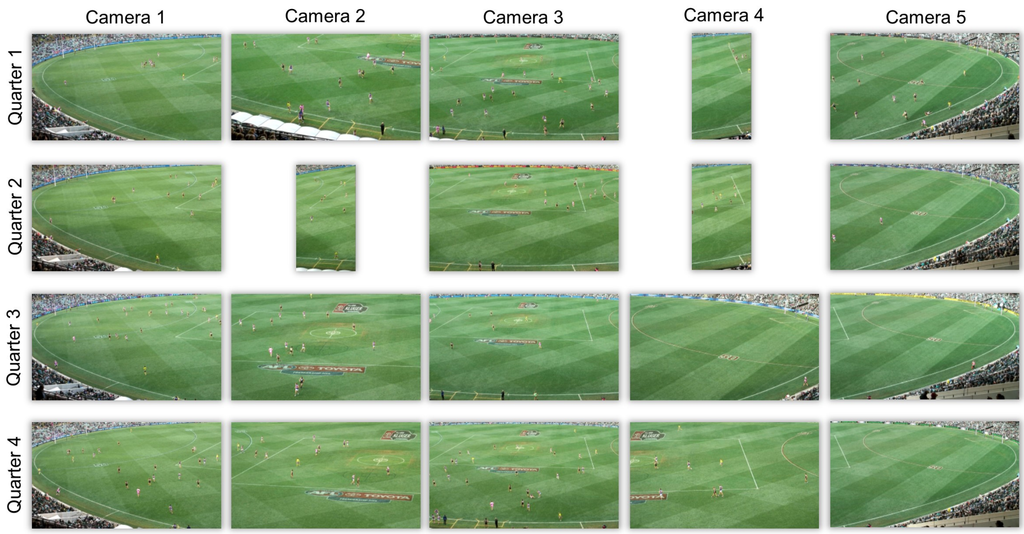



*Figure A.5: The four quarters filmed with the five cameras for the round 14 Port Adelaide VS Western Bulldogs match*

# B. Code

Listed in this appendix are the main Matlab files used throughout the pipeline. Some are written from scratch, others are modified from the vision toolbox[1] or tracking packages[2].

## B.1 Preprocessing (Pre-Detector)

### B.1.1 `extractFrames.m`

`PATH: \CODE\extractFrames.m`

Description:

- Extracts `.jpg` frames from `.avi` videos
- Can specify interval (eg. every 25th frame), starting frame, and final frame

### B.1.2 `gtAn.m` (*modified*), originally `bbLabeler.m`

`PATH: \CODE\gtAn.m`

Description:

- Frame-by-frame ground truth bounding box annotator
- Includes functionality for classes, occlusion flagging and angled boxes

Modifications:

1. Added team ID functionality
2. Changed method of drawing bounding boxes, restricting aspect ratio and speeding up annotation time

## B.2 Detector

### B.2.1 `filterTrain.m`

`PATH: \CODE\filterTrain.m`

Description:

- Accumulates all ground truth from a specified set of tests into a training directory

[1] http://vision.ucsd.edu/~pdollar/toolbox/doc/
[2] http://www.milanton.de/files/software/dctracking-v1.0.zip

- Can specify whether to include or exclude particular samples based on team ID and occlusion flag

### B.2.2 `train.m` (***modified***), originally `acfTrain.m`

PATH: `\CODE\train.m`

Description:

- Trains the cascade of classifiers with AdaBoost
  (***Note:*** *This code was run on the ACVT cluster as the large amounts of training data required large amounts of memory*).

Modifications:

1. Added LBP feature option
2. Changed degree of overlap to `.3` from `.1` tolerated for choosing negative samples that overlap with ground truth positives to push for harder negatives.

### B.2.3 `acfDetect.m` (***modified***)

PATH: `\CODE\toolbox\dollar\detector\acfDetect.m`

Description:

- The detector wrapper program from the vision toolbox

Modifications:

1. Removed `bbs=1` to allow confidence to be returned and written to output file

### B.2.4 `detect.m`

PATH: `\CODE\detect.m`

- Uses a pre-trained detector model trained with `train.m`, and a set of input images and runs the detector model on the images
- Applies mask to frames before being passed into `acfDetect()` method
- Outputs detection `.txt` files for each frame. Each line of the output file specifies a bounding box position, size and confidence: `topLeftXpos, topLeftYpos, width, height, confidence`
- Outputs detection `.jpg` image files for each frame with green bounding boxes drawn around positive samples, and with confidence written within each box

### B.2.5 `evaluate.m`

`PATH: \CODE\evaluate.m`

- Evaluates a detector on a set of ground truth
- Outputs DET and PR curves
- Outputs detection `.txt` and `.jpg` files, and can be run instead of `detect.m`

### B.2.6 `compare.m`

`PATH: \CODE\compare.m`

Description:

- Compares the results of multiple different detector models by comparing their hypothesis detections to ground truths
- Plots DET and PR graphs with for multiple detector models

## B.3 Pre-Team Classifier

### B.3.1 `filterSVMData.m`

`PATH: \CODE\filterSVMData.m`

Description:

- Assembles and filters ground truth into SVM training and testing data
- Occlusion samples and particular teams can be specified as being included or excluded
- Evenly splits ground truth into training and tested datasets by putting consecutive ground truth annotation files into alternating sets (eg. one to test, one to train, one to test, ... etc.)

## B.4 Team Classifier

### B.4.1 `teamFeatures.m`

`PATH: \CODE\teamFeatures.m`

Description:

- Calculates the weighted colour histogram descriptor vectors used for team classification

### B.4.2 `teamTrainer2.m`

`PATH: \CODE\teamTrainer2.m`

Description:

- Trains specific team classifiers on a per-dataset basis using Matlab's inbuilt SVM implementation with a quadratic kernel function
  (***Note:*** *A similar program* `teamTrainer.m` *was also implemented which builds team models for all matches combined*).

### B.4.3 `teamClassifier2.m`

`PATH: \CODE\teamClassifier2.m`

Description:

- Classifies detections into teams using provided SVM models
- All detections are tested against all models, with detections assigned to a model if and only if that model classifies as positive (if multiple classify as positive the detection is marked as unknown)
- PR and ROC curves are outputted and save in both `.jpg` and `.fig` formats
- Outputs same `.txt` detections from the input but each with team ID and team confidence
- Outputs same detection images `.jpg` with each box coloured with team colour
  (***Note:*** *Again a similar program* `teamClassifier.m` *was also implemented which builds team models for all matches combined*).

## B.5 Pre-Tracker

### B.5.1 `cnvrtDetTr.m`

`PATH: \CODE\cnvrtDetTr.m`

Description:

- Converts the comma separated `.txt` team classification output detections into an `.xml` format for input into the tracker

## B.6 Tracker

### B.6.1 `myTracker.m`

`PATH: \CODE\myTracker.m`

Description:

- A basic Kalman Filter style tracking implementation that greedily assigns detections to frames based on estimated velocities calculated from past track motion

### B.6.2 `dcTracker.m` (***modified***)

`PATH: \CODE\tracking\dctracking-v1.0\dcTracker.m`

Description:

- The main tracking program for the energy minimisation approach
- Outputs results in `.xml` similar to that of the input, as well as a `.jpg` sequence and an `.avi` video with the aid of `seq2vid.m`

Modifications:

1. Added the Kalman Filter implementation as initial solution
2. Added team functionality to tracks, where a track is assigned the team that most of its detections are classified as
3. Numerous other alterations to sub-functions

## B.7 Post-processing (Post-Tracker)

### B.7.1 `seq2vid.m`

`PATH: \CODE\seq2vid.m`

Description:

- Converts a `.jpg` image sequence into an `.avi` video for easier viewing (***Note:*** *This was called inside* `dcTracker.m` *at the end as it was always desirable to have video outputted, however it can be used on its own to create video from the detection and team classification output images*).

## B.8 Other

### B.8.1 `runAll.m`

`PATH: \CODE\runAll.m`

Description:

- A program containing all of the stages of the pipeline in a single script, allowing initial paths to be setup and the script will complete the entire pipeline from raw video to video with tracks

# Bibliography


[1] A Aizerman, Emmanuel M Braverman, and LI Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, 25:821–837, 1964.

[2] Shai Avidan. Ensemble tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(2):261–271, 2007.

[3] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(4):509–522, 2002.

[4] Ben Benfold and Ian Reid. Stable multi-target tracking in real-time surveillance video. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3457–3464. IEEE, 2011.

[5] Samuel Blackrnan and Artech House. Design and analysis of modern tracking systems. *Boston, MA: Artech House*, 1999.

[6] Michael D Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1515–1522. IEEE, 2009.

[7] Michael D Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(9):1820–1833, 2011.

[8] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 142–149. IEEE, 2000.

[9] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[10] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[11] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *Computer Vision–ECCV 2006*, pages 428–441. Springer, 2006.

[12] Piotr Dollár. Piotr's Image and Video Matlab Toolbox (PMT). http://vision.ucsd.edu/ pdollar/toolbox/doc/index.html.

[13] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection.

[14] Piotr Dollár, Boris Babenko, Serge Belongie, Pietro Perona, and Zhuowen Tu. Multiple component learning for object detection. In *Computer Vision–ECCV 2008*, pages 211–224. Springer, 2008.

[15] Piotr Dollár, Serge Belongie, and Pietro Perona. The fastest pedestrian detector in the west. In *BMVC*, volume 2, page 7. Citeseer, 2010.

[16] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: A benchmark. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 304–311. IEEE, 2009.

[17] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(4):743–761, 2012.

[18] Anna Ellis, Ali Shahrokni, and James Michael Ferryman. Pets2009 and winter-pets 2009 results: A combined evaluation. In *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*, pages 1–8. IEEE, 2009.

[19] Markus Enzweiler and Dariu M Gavrila. Monocular pedestrian detection: Survey and experiments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(12):2179–2195, 2009.

[20] Andreas Ess, Bastian Leibe, and Luc Van Gool. Depth and appearance for mobile scene analysis. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[21] Zipei Fan, Zeliang Wang, Jinshi Cui, Franck Davoine, Huijing Zhao, and Hongbin Zha. Monocular pedestrian tracking from a moving vehicle. In *Computer Vision-ACCV 2012 Workshops*, pages 335–346. Springer, 2013.

[22] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[23] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[24] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[25] J Ferryman and Anna Ellis. Pets2010: Dataset and challenge. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 143–150. IEEE, 2010.

[26] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[27] Keinosuke Fukunaga and Larry Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on*, 21(1):32–40, 1975.

[28] David Geronimo, Antonio M Lopez, Angel Domingo Sappa, and Thorsten Graf. Survey of pedestrian detection for advanced driver assistance systems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(7):1239–1258, 2010.

[29] Rafael C Gonzalez and Richard E Woods. Digital image processing, 2002.

[30] Yaowen Guan, Xiaoou Chen, Deshun Yang, and Yuqian Wu. Multi-person tracking-by-detection with local particle filtering and global occlusion handling. In *Multimedia and Expo (ICME), 2014 IEEE International Conference on*, pages 1–6. IEEE, 2014.

[31] Raffay Hamid, Ramkrishan Kumar, Jessica Hodgins, and Irfan Essa. A visualization framework for team sports captured using multiple static cameras. *Computer Vision and Image Understanding*, 118:171–183, 2014.

[32] Raffay Hamid, Ramkrishan K Kumar, Matthias Grundmann, Kihwan Kim, Irfan Essa, and Jessica Hodgins. Player localization using multiple static cameras for sports visualization. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 731–738. IEEE, 2010.

[33] Sibt Ul Hussain, William Triggs, et al. Feature sets and dimensionality reduction for visual object detection. In *British Machine Vision Conference*, 2010.

[34] W Jiang. Human feature extraction in vs image using hog. 2007.

[35] Zhengqiang Jiang, Du Q Huynh, William Moran, Subhash Challa, and Nick Spadaccini. Multiple pedestrian tracking using colour and motion models. In *Digital Image Computing: Techniques and Applications (DICTA), 2010 International Conference on*, pages 328–334. IEEE, 2010.

[36] Michael J Jones and Daniel Snow. Pedestrian detection using boosted features over many frames. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[37] Bernardin Keni and Stiefelhagen Rainer. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008, 2008.

[38] Zia Khan, Tucker Balch, and Frank Dellaert. Mcmc data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12):1960–1972, 2006.

[39] Khalil Khattab, Philippe Brunet, Julien Dubois, Johel Miteran, et al. Real time robust embedded face detection using high level description. *New Approaches to Characterization and Recognition of Faces*, 2011.

[40] Bastian Leibe, Edgar Seemann, and Bernt Schiele. Pedestrian detection in crowded scenes. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 878–885. IEEE, 2005.

[41] Shengcai Liao, Xiangxin Zhu, Zhen Lei, Lun Zhang, and Stan Z Li. Learning multi-scale block local binary patterns for face recognition. In *Advances in Biometrics*, pages 828–837. Springer, 2007.

[42] Jingchen Liu, Peter Carr, Robert T Collins, and Yanxi Liu. Tracking sports players with context-conditioned motion models. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1830–1837. IEEE, 2013.

[43] Francisco Madrigal and Jean-Bernard Hayet. Evaluation of multiple motion models for multiple pedestrian visual tracking. In *Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on*, pages 31–36. IEEE, 2013.

[44] Bangalore S Manjunath and Wei-Ying Ma. Texture features for browsing and retrieval of image data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(8):837–842, 1996.

[45] Alvin Martin, George Doddington, Terri Kamm, Mark Ordowski, and Mark Przybocki. The det curve in assessment of detection task performance. Technical report, DTIC Document, 1997.

[46] Anton Milan, Konrad Schindler, and Stefan Roth. Challenges of ground truth evaluation of multi-target tracking. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 735–742. IEEE, 2013.

[47] Anton Milan (né Andriyenko), Stefan Roth, and Konrad Schindler. Continuous energy minimization for multi-target tracking. 2013.

[48] Anton Milan (né Andriyenko) and Konrad Schindler. Multi-target tracking by continuous energy minimization. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1265–1272. IEEE, 2011.

[49] Anton Milan (né Andriyenko), Konrad Schindler, and Stefan Roth. Discrete-continuous optimization for multi-target tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1926–1933. IEEE, 2012.

[50] Yadong Mu, Shuicheng Yan, Yi Liu, Thomas Huang, and Bingfeng Zhou. Discriminative local binary patterns for human detection in personal album. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[51] Peter Nillius, Josephine Sullivan, and Stefan Carlsson. Multi-target tracking-linking identities using bayesian network inference. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2187–2194. IEEE, 2006.

[52] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

[53] Kenji Okuma, David G Lowe, and James J Little. Self-learning for player localization in sports video. *arXiv preprint arXiv:1307.7198*, 2013.

[54] Michael Oren, Constantine Papageorgiou, Pawan Sinha, Edgar Osuna, and Tomaso Poggio. Pedestrian detection using wavelet templates. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 193–199. IEEE, 1997.

[55] Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1201–1208. IEEE, 2011.

[56] Fatih Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 829–836. IEEE, 2005.

[57] Zhi Qiang Qu, Dan Tu, and Jun Lei. Online pedestrian tracking with kalman filter and random ferns. *Applied Mechanics and Materials*, 536:205–212, 2014.

[58] Donald B Reid. An algorithm for tracking multiple targets. *Automatic Control, IEEE Transactions on*, 24(6):843–854, 1979.

[59] William Robson Schwartz, Aniruddha Kembhavi, David Harwood, and Larry S Davis. Human detection using partial least squares analysis. In *Computer vision, 2009 IEEE 12th international conference on*, pages 24–31. IEEE, 2009.

[60] Rainer Stiefelhagen, Keni Bernardin, Rachel Bowers, John Garofolo, Djamel Mostefa, and Padmanabhan Soundararajan. The clear 2006 evaluation. In *Multimodal Technologies for Perception of Humans*, pages 1–44. Springer, 2007.

[61] Kah-Kay Sung and Tomaso Poggio. Example-based learning for view-based human face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(1):39–51, 1998.

[62] Xiaoyang Tan and Bill Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. In *Analysis and Modeling of Faces and Gestures*, pages 168–182. Springer, 2007.

[63] Siyu Tang, Mykhaylo Andriluka, and Bernt Schiele. Detection and tracking of occluded people. *International Journal of Computer Vision*, pages 1–12, 2012.

[64] Paul Viola and Michael Jones. Robust real-time object detection. *International Journal of Computer Vision*, 4:34–47, 2001.

[65] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[66] Paul Viola, Michael J Jones, and Daniel Snow. Detecting pedestrians using patterns of motion and appearance. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 734–741. IEEE, 2003.

[67] Stefan Walk, Nikodem Majer, Konrad Schindler, and Bernt Schiele. New features and insights for pedestrian detection. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 1030–1037. IEEE, 2010.

[68] Xiaoyu Wang, Tony X Han, and Shuicheng Yan. An hog-lbp human detector with partial occlusion handling. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 32–39. IEEE, 2009.

[69] Christian Wojek and Bernt Schiele. A performance evaluation of single and multi-feature people detection. In *Pattern Recognition*, pages 82–91. Springer, 2008.

[70] H Wold and E Lyttkens. Nonlinear iterative partial least squares (nipals) estimation procedures. *Bulletin of the International Statistical Institute*, 43(1), 1969.

[71] Hanxuan Yang, Ling Shao, Feng Zheng, Liang Wang, and Zhan Song. Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74(18):3823–3831, 2011.

[72] Qiang Zhu, M-C Yeh, Kwang-Ting Cheng, and Shai Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1491–1498. IEEE, 2006.