VIOLENCE DETECTION IN CROWD FOOTAGE: ENGINEERING STATISTICAL FEATURES USING TRANSFORMED OPTICAL FLOW

by

Mia Tai Parenteau

Submitted in partial fulfillment of the requirements
for the degree of Master of Science

at

Dalhousie University
Halifax, Nova Scotia
December 2020

Table of Contents

List of Tables

List of Figures

Abstract

Video surveillance technology is becoming increasingly common and is often used to increase safety in public spaces. However, the effectiveness of the video information on its own is questionable as it relies on someone manually reviewing the footage, in real time or retroactively. The current research intends to improve video surveillance technology by using computer vision and machine learning techniques to automatically detect a violent event within a crowded scene, in real time. Meaningful information is extracted from the raw, gray-scale pixel data. This is done through optical flow feature extraction, then a projection of those values onto a plane which approximates the ground in the reality of the scene. The projection is done as a means to account for perspective distortion affecting the optical flow calculation. The dimension of the feature set is reduced through quantizing the frame and following several statistics through time. Random forest variable ranking is leveraged to further reduce the feature set. Promising results are found using simple cutoff classifiers in the target region of the frame.

Acknowledgements

I'd like to thank my supervisor Dr. Hong Gu and the industry research director Phil Munz for their patience and guidance, my parents for their enthusiastic support and advice, and finally my fellow classmates for their company and generosity.

Chapter 1

Introduction

1.1 Background

The presence of video surveillance is becoming increasingly ubiquitous as camera technology improves. This fact combined with ever-decreasing computation times presents an opportunity for intelligent video data analysis. The overwhelming wave of video information has begun to collide with advances in statistics and computer science, generating new and exciting learning problems in the multidisciplinary field of artificial intelligence. Artificial intelligence focuses on developing a machine's ability to perform higher-order cognitive tasks. The relevant sub-field of computer vision is specific to making better use of the images and videos that permeate our digital world through processing and analyzing the large amounts of noisy data often associated with such mediums. Jay Stanley in *The Dawn of Surveillance* draws the analogy that video footage is to intelligent video analytics as the eyes are to the human brain and likens recent technological advances to an awakening of a dormant mind (Stanley, 2019). That is, models trained to automatically interpret visual data are an essential factor in our ability to process large volumes of the like in such a way that is useful, intentional, and meaningful.

There is a seemingly endless range of applications in computer vision. With moving picture data, there have been developments in the analysis of road traffic, sports, and even identifying behavioural markers for psychological diagnoses such as autism (Buch et al., 2011; Thomas et al., 2017; Hashemi et al., 2012). In fact, applications to moving picture data largely focus on the classification of human movement. This particular area of computer vision is called "human action recognition". Human

action recognition research usually develops its methods using either benchmark data sets created in an artificial setting or more naturalistic footage like sporting events, Hollywood movies, and of course surveillance footage.

As shown above, there is a great amount of diversity in the types of images as well as research questions to be had, and each objective requires a process tailored to suit its needs (Klette, 2014). There are pros and cons when extracting clear information from video surveillance footage. Having a fixed camera capturing a consistent scene eliminates the need to stabilize a shaky camera while also maintaining a naturalistic setting, ideal for real-world application. Disadvantages could be occurrences such as occlusion of the objects of interest, further distances from the camera, and the potential of needing to manually identify the response within the footage, the latter of which can translate to hours of work if dealing with multiple data sets. Nonetheless applications of surveillance footage data are extremely useful and could have a major impact with regards to the interest of public safety. For example, intelligent video analysis can be of assistance in emergency situations. Models have been trained to identify when someone falls over, a helpful tool in a place such as a retirement home where quick response to falls are essential (Feng et al., 2014). A software that detects drowning persons in swimming pools can also assist on-site lifeguards, who are responsible for overseeing many swimmers (Lu & Tan, 2004). It is clear that faster first responses to such events could be the difference between life and death.

The subject of the current work deals with a similarly urgent situation, one in which a physical fight occurs in a crowded scene. This type of event (and the others mentioned above) are a specific type of human action recognition problem sometimes referred to as "anomaly detection". Anomaly detection deals with modeling of rare or infrequent events. As Mahadevan et al. write, "...the goal is not so much to analyze normal crowd behaviour, but to detect deviations from it." (Mahadevan et al., 2010). The need for such modeling is apparent because although there is no doubt in this day and age that video surveillance could conceivably capture such events, the effectiveness of the technology in practice is questionable as it relies on humans constantly monitoring the footage and detecting anomalies by eye.

In computer vision, training an effective model typically comprises the following steps: image pre-processing, feature extraction, and statistical modeling. Pre-processing refers to cleaning and transforming data so that it is more manageable while feature extraction amplifies relevant information while discarding or muting noisy data. Both steps can have considerable conceptual overlap and are sometimes used inter-

changeably. Pre-processing of video data is necessary due to the overwhelmingly large amount of data provided, both spatially through pixels and temporally through video frames. This will ensure that the data brought to the classification step is meaningful and can be classified efficiently and effectively. Considering the diverse scope of research goals within computer vision, it is necessary to select appropriate methods at each of these junctions. Depending on the question of interest, the same data set could have wildly different methods of solution.

For the data set that will be used in the current work and in the context of human action recognition in general, conditions of the recording, the scene, actions of interest, classes to be assigned, and desired run time should all be taken into account. Pre-processing of the data requires capturing differences in perceived motion over the image frames. A variety of different techniques have been reported in the scientific literature to do this. Methods for motion-related feature extraction can be divided into local (feature-based) and global (frame-based) methods. Feature-based methods involve finding and following interest points in an image through a feature vector. Examples that fit the context of the current project are histogram of oriented gradients (HOG) and scale-invariant feature transform (SIFT). To contrast, frame-based methods extract a representation of the entire frame and then calculate local descriptors (Gao et al., 2016). Local, feature-based methods can be more precise however processing times increase with the number of features being tracked. Global methods are more naïve in their approach but require considerably less computation time.

Feature extraction using global methods for violence detection are typically predicated on the idea that people who are fighting exhibit certain motion patterns that can be identified at the pixel level. An example of this type of descriptor is Violent Flows (ViF) developed by Hassner et al. (2012). They used optical flow in conjunction with other feature extraction methods such as thresholding and quantized histograms to develop a classifier for violence that is specific to crowd footage. Optical flow is a very popular method which converts pixels into flow vector magnitudes that can be used to quantify the speed of pixels across two consecutive frames (Berthold & Schunck, 1981). Optical flow is extremely useful and therefore is extracted as a step in the current research.

Once relevant information has been extracted it is possible to train a supervised learning model to identify violent or non-violent frames in a video. Current approaches commonly use support vector machines for the classification step (Hassner et al., 2013; Laptev et al., 2008; Bilinski & Bremond, 2016; Marsden, McGuinness, Little,

& O'Connor, 2016). In contrast, the current research uses a random forest classification model. Random forest is known to be adequate in a variety computer vision classification problems, including anomaly detection (Ali et al., 2012; Primartha & Tama, 2017). In addition to using supervised learning models, a simpler alternative is optimizing some sort of classification rule. If possible this option would be preferred as the training processing time is faster.

The proposed research strives to improve video surveillance technology by using computer vision techniques and machine learning models to detect anomalous movements, specifically fighting. There are two main objectives. The first is to process the raw, gray-scale pixel data with the intention of calculating information-rich features while consequently lowering the overall dimension of the data. This is done first by taking advantage of a well-established dense optical flow extraction algorithm. Next a novel method is developed to achieve a more realistic estimate of the motion in the video. Perspective-related data transformations are derived from measurements of the real-life location in the recording. Statistically-based feature extraction methods are also applied to further reduce the dimension of the feature set. The second goal is to develop and evaluate a classifier which is able to accurately categorize a series of video frames as either violent or non-violent, both spatially and temporally. This is accomplished through thoughtful feature selection, model training and testing, and optimizing simple cutoff points.

The ultimate goal is to integrate the findings from the current research into a video surveillance system so that it may detect anomalous behaviour automatically, improving the overall effectiveness of detecting violence in crowded spaces. This will need to be done in real time and current methods used should be mindful of potential future work that would require scaling to larger data sets. The current research focuses on the overall classification of fighting and non-fighting frames to provide more general conclusions. In practice the most important information would probably be the exact instance in which a fight erupts since that would be the point where a response is necessary. Future development regarding the application of the classifier may consult security specialists as to what the priority of the classification scheme may be.

1.2 Structure of Thesis

The current chapter provides the motivation and context for the current problem. In the second chapter, we set the scene by detailing the optical flow extraction process

while also providing a comprehensive description of the data set. Once these pieces are understood, the following chapter addresses conditions and motivation leading up to a second "ground speed" data transformation which adjusts for data discrepancies caused by perspective projection. The ground speed transformation, which uses optical flow and measurements from the real-life scene, is outlined in detail. The fourth chapter discusses a statistical approach to feature extraction, and subsequent feature selection. The final two chapters present the numerical results, their interpretation, and then debrief the implications of these results as well as their relevance to future work.

Chapter 2

Defining the Data Set

2.1 Optical Flow

Optical flow is defined as a measurement of the perceived velocity between consecutive frames of motion (Horn & Schunck, 1981). As was outlined in the introduction, accurately measuring movement in video data is of the utmost importance for human action recognition, and optical flow is a widely popular approach when it comes to extracting such data. It is labeled as perceived velocity because actual pixels are not moving; indices of pixel location are concrete yet objects can be perceived as moving. If a particular pattern of luminance is followed to new pixel locations gradually across many frames, then we perceive that pattern as an object traversing through a scene. Optical flow is a calculation of these patterns of luminance and uses the brightness of pixels in an image environment. A gray-scale version of the video frames are used as is standard practice for virtually all optical flow algorithms.

The image environment is then defined through a few key assumptions regarding the objects in the frame. Because objects are being tracked through pixel brightness across time, we must assume that the brightness of objects do not change over that space and time (brightness constancy). We also assume that points do not move large distances over consecutive frames, and that points tend to move similarly to their neighbors (spatial coherence). Keep in mind that these parameters do not usually hold for the entire frame however in applying optical flow it is only important that they are true for a given local area of the image (Beauchemin & Barron, 1995).

Further considerations from Beauchemin & Barron (1995) expand upon the main constraints stated above. An object inside an image is assumed to be represented

with uniform brightness, it does not vary in its representation based on shade or light (e.g. shadows or direct sunlight). Another assumption is that the projection of reality onto the image is done so with accuracy, implying that the effects of perspective or camera lens distortion are not accounted for. Optical flow also does not account for translucent or occluded objects. As with the above assumptions, these constraints only need to be valid for neighborhoods within the image.

An optical illusion known as the aperture problem also affects optical flow calculations. This phenomena occurs when a local area of an image is partially or fully within an object (i.e. no edges or corners are represented within the area) (Klette, 2014). The consequence is that, from the viewpoint of the local area, no movement is detected. Furthermore if only a horizontal edge is included, it is impossible to detect horizontal movement while preserving detection of vertical movement and vice versa. It is only corners that are able to capture the full range of motion. Many optical flow algorithms have accounted for the aperture problem through introducing additional constraints.

Despite these pitfalls, optical flow still manages to measure object velocity within moving images in sufficient detail to be valuable. There are many different techniques that exist to implement optical flow, however they can all be connected through some foundational concepts.

2.1.1 Deriving Optical Flow (in General)

The following derivation can be found in Beauchemin & Barron (1995). In optical flow we wish to find where objects have been displaced to. As stated above, we don't actually observe movement but instead only image intensities, $I(x, y, t)$, at each location (x, y) and for each time t. If movement is happening at a low rate relative to how frequently images are garnered, then $I(x, y, t)$ can be expected to be a smooth function of x, y and t. Thus Taylor's series gives

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt + O^2 \tag{2.1}$$

where O^2 represents negligible higher order terms. For the current implementation $\frac{\partial I}{\partial x}$ is approximated through

$$\frac{I(x + 1, y, t) - I(x - 1, y, t)}{2} \tag{2.2}$$

and the same scheme is used for the y-direction intensity gradient (Bouget, 2001). $\frac{\partial I}{\partial t}$ is the difference in intensity between t and $t+dt$ at a given location (x, y). Assume now that the object at (x, y) at time t is displaced to an unknown location $(x+dx, y+dy)$ at time $t + dt$. Due to the brightness constancy assumption, we expect

$$I(x, y, t) \approx I(x + dx, y + dy, t + dt) \tag{2.3}$$

Substituting in (2.1) and dividing by dt gives

$$0 \approx I_x u + I_y v + I_t \tag{2.4}$$

$u = \frac{dx}{dt}$ and $v = \frac{dy}{dt}$ are the x-direction and y-direction velocity and I_x, I_y, and I_t are the partial derivatives of the image intensity function with respect to x, y, and t. We now have an equation that may allow us to infer where the object that was at (x, y) at time t has likely moved to by time $t+dt$. Let $\nabla(I_x, I_y)$ be the spatial intensity gradient (the partial derivatives which represent the direction of pixel intensity change), leading to the final expression

$$0 \approx \nabla(I_x, I_y) \cdot (u, v) + I_t \tag{2.5}$$

known as the *optical flow constraint equation*. The challenge of calculating optical flow lies in solving for u and v. With one equation and two unknowns further constraints are required, of which there are many options. An in-depth review of all techniques is outside of the scope of this thesis, for more information you may consult Beauchemin & Barron (1995).

Although there are numerous methods and even categories of methods when it comes to estimating optical flow, they can generally be divided into two principal types: sparse and dense. Sparse optical flow methods are defined by using a sample of points to estimate movement in the entire frame while dense optical flow makes use of all points in the frame. The Lucas-Kanade method (Lucas & Kanade, 1981) is a prime example of sparse optical flow. A new assumption is introduced which states that (u, v) is the same for n neighboring pixels around a centre point p. Using the optical flow constraint equation (2.5), a system of equations can be expressed in the form $\boldsymbol{Av} = \boldsymbol{b}$. $\boldsymbol{A}$ is an $n \times 2$ matrix of partial derivatives (I_x, I_y) whereby each row corresponds to a point in the neighborhood. $\boldsymbol{v} = (u, v)$ and $\boldsymbol{b}$ is an $n \times 1$

matrix representing $-I_t$ for each pixel in the neighborhood. (u, v) is solved using least squares. That is, $\boldsymbol{v} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b}$.

Sparse and dense optical flow methods of calculation are contrasted by a trade-off between accuracy of estimated displacements and computational efficiency. Sparse optical flow is faster with less accuracy whereas dense optical flow is slower with greater accuracy. For the current data set, dense optical flow is calculated because although efficiency is important for the current problem, the nature of the scene required a more detailed depiction of motion. Details on the specific dense optical flow method used in the context are described next.

2.1.2 Dense Optical Flow Estimation

The following method was derived by Farnebäck (2003) and provides calculations for two-frame dense optical flow estimation. As previously established, in order to solve the optical flow constraint equation for u and v, additional constraints are required. This method introduces a rather presumptuous constraint which is nonetheless shown to be useful. We assume that image intensity of some neighborhood around each pixel can be approximated through the following polynomial.

$$f(\boldsymbol{x}) \sim \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x} + c \tag{2.6}$$

where location vector $\boldsymbol{x} = (x, y)$. $\boldsymbol{A}$, $\boldsymbol{b}$ and c are matrix, vector, and scalar coefficients respectively. A detailed overview of background and implementation involved in the calculation of the coefficients can be found in Farnebäck (2002), a short summary is explained here.

Coefficients are estimated through weighted least squares, using a normalized convolution of pixel intensities in the neighborhood centred around $\boldsymbol{x}$. Let $\mathbf{f}$ represent a vector containing pixel intensities located in a neighborhood around $\boldsymbol{x}$. The polynomial basis functions are $\{1, x, y, x^2, y^2, xy\}$ and are applied to each pixel intensity in $\mathbf{f}$ which is then stored in matrix $\boldsymbol{B}$. Weights are applied for certainty of intensity and applicability of pixels. That is, we let $\boldsymbol{W}_c$ be a diagonal matrix of certainty weights which are meant to regulate the reliability of pixel intensity estimates in the neighborhood. In practice it is mostly used to indicate when the edge of the frame is reached. $\boldsymbol{W}_a$ is also a diagonal weight matrix that indicates the radial proximity of pixels in the neighborhood to $\boldsymbol{x}$. Thus a vector of coefficients, $\boldsymbol{r}$ is found through weighted least squares

$$\boldsymbol{r} = (\boldsymbol{B}^T \boldsymbol{W}_a \boldsymbol{W}_c \boldsymbol{B})^{-1} \boldsymbol{B}^T \boldsymbol{W}_a \boldsymbol{W}_c \mathbf{f} \tag{2.7}$$

where $\boldsymbol{r}$ contains six elements, corresponding to each basis function. Then, coefficients in the polynomial are set as

$$c = r_1 \qquad \boldsymbol{b} = \begin{pmatrix} r_2 \\ r_3 \end{pmatrix} \qquad \boldsymbol{A} = \begin{pmatrix} r_4 & \frac{r_6}{2} \\ \frac{r_6}{2} & r_5 \end{pmatrix} \tag{2.8}$$

Note that $\boldsymbol{A}$ is symmetric. Farnebäck (2003) asserts that fast computation of these coefficients is possible through a "...hierarchical scheme of separable convolutions".

Now, we first consider the approximation in (2.6) globally at time $t = 1$,

$$f_1(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{A}_1 \boldsymbol{x} + \boldsymbol{b}_1^T \boldsymbol{x} + c_1 \tag{2.9}$$

For time $t = 2$, we can define the global intensity through f_1 and the global displacement $\boldsymbol{d}$

$$f_2(\boldsymbol{x}) = f_1(\boldsymbol{x} - \boldsymbol{d}) = (\boldsymbol{x} - \boldsymbol{d})^T \boldsymbol{A}_1 (\boldsymbol{x} - \boldsymbol{d}) + \boldsymbol{b}_1^T (\boldsymbol{x} - \boldsymbol{d}) + c_1 \tag{2.10}$$

which through expansion and simplification can be re-expressed as

$$f_2(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{A}_1 \boldsymbol{x} + (\boldsymbol{b}_1 - 2\boldsymbol{A}_1 \boldsymbol{d})^T \boldsymbol{x} + \boldsymbol{d}^T \boldsymbol{A}_1 \boldsymbol{d} - \boldsymbol{b}_1 \boldsymbol{d} + c_1 \tag{2.11}$$

The polynomial form is achieved when setting $\boldsymbol{A}_2 = \boldsymbol{A}_1$, $\boldsymbol{b}_2 = \boldsymbol{b}_1 - 2\boldsymbol{A}_1 \boldsymbol{d}$, and $c_2 = \boldsymbol{d}^T \boldsymbol{A}_1 \boldsymbol{d} - \boldsymbol{b}_1^T \boldsymbol{d} + c_1$. Using the expressions for $\boldsymbol{A}_2$ and $\boldsymbol{b}_2$ we can solve for $\boldsymbol{d}$

$$\boldsymbol{d} = \frac{-1}{2} \boldsymbol{A}_1^{-1} (\boldsymbol{b}_2 - \boldsymbol{b}_1) \tag{2.12}$$

as long as $\boldsymbol{A}_1$ is invertible. Since velocity is the ultimate value of interest and Δt is known, the goal is to find a feasible way to estimate displacement $\boldsymbol{d}$.

The global polynomial assumption is relaxed and applied as a neighborhood approximation, offering a more realistic estimation at the cost of introducing some error (Farnebäck, 2003). Local coefficients are defined as a function of a local area, and are re-expressed as $\boldsymbol{A}(\boldsymbol{x})$, $\boldsymbol{b}(\boldsymbol{x})$, and $c(\boldsymbol{x})$. In practice, $\boldsymbol{A}(\boldsymbol{x})$ is estimated as the average of $\boldsymbol{A_1}(\boldsymbol{x})$ and $\boldsymbol{A_2}(\boldsymbol{x})$, and $\boldsymbol{\Delta b}(\boldsymbol{x}) = \frac{-1}{2}(\boldsymbol{b}_2(\boldsymbol{x}) - \boldsymbol{b}_1(\boldsymbol{x}))$. So,

$$\boldsymbol{A}(\boldsymbol{x}) \boldsymbol{d}(\boldsymbol{x}) = \Delta \boldsymbol{b}(\boldsymbol{x}) \tag{2.13}$$

To apply this equation over a local neighborhood, the spatial coherence constraint is utilized in the context of assuming that displacement $\boldsymbol{d}(\boldsymbol{x})$ is not dramatically different within points in a given neighborhood. Using $\Delta\boldsymbol{x}$ to represent the change in location over N, we wish to solve the following for $\boldsymbol{d}(\boldsymbol{x})$

$$min\left[\sum_{\Delta x\in N} w(\Delta\boldsymbol{x}) \parallel \boldsymbol{A}(\boldsymbol{x}+\Delta\boldsymbol{x})\boldsymbol{d}(\boldsymbol{x}) - \Delta\boldsymbol{b}(\boldsymbol{x}+\Delta\boldsymbol{x}) \parallel^2\right] \tag{2.14}$$

where $w(\Delta\boldsymbol{x}) = \boldsymbol{w}$ is a weight function which, like $\boldsymbol{W}_a$, regulates radial proximity to $\boldsymbol{x}$ over the points in N. Taking the partial derivative with respect to $\boldsymbol{d}(\boldsymbol{x})$ and dropping some notation to simplify, expand the quadratic and find

$$\begin{aligned} 0 &= \frac{\partial}{\partial\boldsymbol{d}(\boldsymbol{x})}\sum \boldsymbol{w}\boldsymbol{d}(\boldsymbol{x})^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{d}(\boldsymbol{x}) - 2\sum \boldsymbol{w}\boldsymbol{d}(\boldsymbol{x})^T\boldsymbol{A}^T\Delta\boldsymbol{b} + \sum \boldsymbol{w}\Delta\boldsymbol{b}^T\Delta\boldsymbol{b} \\ &= \sum \boldsymbol{w}\cdot\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{d}(\boldsymbol{x}) - 2\sum \boldsymbol{w}\boldsymbol{A}^T\boldsymbol{\Delta}\boldsymbol{b} \\ \boldsymbol{d}(\boldsymbol{x}) &= 2\left(\sum \boldsymbol{w}\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\left(\sum \boldsymbol{w}\boldsymbol{A}^T\boldsymbol{\Delta}\boldsymbol{b}\right) \end{aligned}$$

Coefficients are computed over each point in the neighborhood N, using $\boldsymbol{w}$ to average. From there, it is possible to solve for $\boldsymbol{d}(\boldsymbol{x})$ as long as "the whole neighborhood [isn't] exposed to the aperture problem" (Farnebäck, 2003). What is meant by this statement is that if the entire neighborhood represents the inside of an object (or has limited edge representation) and is subject to the brightness constancy assumption, then it is not possible to capture displacement because it is impossible to discern pixels of the same brightness from one another.

It is also easy to iteratively update this process, allowing the calculation to be done more accurately when considering the entire length of t. To account for instances in which the displacement is large enough to violate the small displacement assumption, there is a technique that can be used called the "coarse-to-fine" method. The idea is that large displacements can be computed when the image is represented with less pixels (coarsely). In the coarser image, pixel displacements are smaller since there are less pixels representing the scene. The algorithm computes the displacement between two frames at some low resolution, then uses that displacement as a prior in the calculation of those same frames at a slightly higher resolution. This is iterated until the full resolution has been calculated for the two frames and then can be repeated over all frames.

2.2 Description of the Data

Figure 2.1: A still from the video used in the analysis.

The raw data used in the current work has been collected from one clip of video surveillance footage, provided by PatriotOne Technologies. About 67 seconds of footage is extracted for the purposes of this project which, at approximately 25 frames per second, translates to exactly 1680 consecutive images in total. Each image has a dimension of 1280×1706 pixels. Gray-scale pixel data from the video frames were used to calculate dense optical flow using the Farnebäck estimation method as described in the previous section. Periodically, certain frames had extremely low optical flow values across the entire frame. This is most likely due to repeating frames in the video clip and therefore are considered extraneous information. These repeating frames were excluded from the analysis, leaving a total of 1401 frames with usable information.

There is one physical altercation in the clip. It takes place approximately 36 seconds into the clip and fully concludes around the 54 second mark, leaving a total of 512 frames of fighting and 889 frames of non-fighting. Fighting and non-fighting cutoffs

are determined by viewing the video, taking the first and last second at which fighting could be identified and converting them into the nearest whole frame number. There is what could be considered lapse in movement during the fighting period, between 48 and 52 seconds. This is due to the people falling to the ground as well as a person obscuring the fight. Movement picks back up again for the remaining two seconds however it is still somewhat obscured by another person in the frame. These aspects to the event could be a detriment during the classification step however it is important to still classify them as fighting; in a real-life situation the model would ideally want to pick up on the full range of the fight. In terms of location in the scene, the fight takes place on the platform and would be considered within the foreground of the image. It takes place entirely inside the top left quadrant of the frame, first moving towards the right and then concludes by moving left towards the starting point. From anecdotal observation other parts of the frame do not appear to have rapid or otherwise significant movement from other objects in reaction to the altercation.

The scene in the video presents a few challenges. First, there are the differences in the crowd density depending on the location in the frame. In the bottom left and top right of the frame, there is little to no movement due to a food stand in the former and the sky and football field in the latter. Then in the bottom right of the frame, people are sparsely gathered compared to a denser crowd at a distance. As a result, people at greater distances are more likely perceived as moving at a slower speed, or to be obscured compared to people in the foreground. These differences are seen to be consistent throughout the time elapsed in the clip.

The camera's field of view covers a large area, which means there is a significant real-life distance between the closest and furthest point in the scene. Due to perspective projection, objects further away from the camera are shown as smaller compared to those closer to the camera; their apparent displacement is also smaller. As mentioned, optical flow calculations do not account for this type of disparity and therefore objects that are moving in the distance are calculated as slower than closer objects traveling at the same speed in reality. Moreover, pixel size and distribution are uniform throughout an image which means that smaller objects are captured with less resolution compared to larger objects. The compounding of these issues results in optical flow estimates being less accurate and of a lower magnitude with increased distance from the camera.

Although it is difficult to account for differences in crowd behaviour at different

regions in the frame, there are options for accounting for perspective projection biases in the scene. The following chapter will outline theoretical motivation and the process of converting optical flow velocity to an estimate of the ground speed in reality as well as correcting for some tilt present in the frame.

Chapter 3

Ground Speed Transformation

3.1 Motivation

Recall that an assumption that precedes optical flow is that the objects depicted in the image are an accurate representation of the reality of the scene; factors such as tilt, perspective, and lens effects are not taken into consideration during the process of calculating optical flow. In order to train a model in detecting certain specific patterns of motion, the data should be comparable within the scene of interest. As such, some processing of the current data set is required prior to its analysis.

Optical flow is computed directly from the gray-scale video pixel data, which is a perspective projection of reality. For some applications such as object tracking, perspective is not important to consider. However for motion pattern detection, if the scale of the data set is not spatially comparable then the extent to which a model can be well-trained is limited. For example, if optical flow values further away from the camera are smaller then the calculated speed at which the objects are moving will always be smaller relative to the objects closer to the camera. In other words, a value that is large for its local area would not necessarily be considered large in the scope of the entire frame. The current data set encounters this very issue, a large distance is covered and closer areas are not comparable with farther ones.

In this chapter, the process of incorporating the distance from the camera into optical flow through a ground speed transformation is presented. In addition, the steps used to account for the tilt in the frame are addressed. The slight fish-eye lens effect seen in the video is not adjusted for because its effect is only minor and the edges of the frame are not of interest to the analysis. Finally, optical flow and ground speed, with

and without adjusting for tilt are visually compared to evaluate the success of the transformations in achieving the ultimate goal of calibrating the data.

3.2 Ground Speed Geometry

We can draw a geometric relationship between the camera, the video images, and the reality captured by the camera as depicted in the images. The camera is fixed at a point in the reality of the scene, and records an image of that scene. These images can be represented as an image plane from which the reality is being projected onto. Optical flow uses the values of pixel intensity from the known image plane to infer movement in reality. If the geometric relationship between the image plane and the reality in a ground plane is known, optical flow can be projected back onto the ground plane, resulting in values that estimate the velocity on the ground. It is expected that the ground speed transformation will result in farther away values (those higher up in the frame) becoming larger and closer values becoming smaller.

The velocity of a point $(\tilde{x}, \tilde{y})$ on ground plane can be found through a Jacobian transformation of optical flow at a corresponding point (x, y) in the image plane,

$$\begin{pmatrix} \frac{\mathrm{d}\tilde{x}}{\mathrm{d}t} \\ \frac{\mathrm{d}\tilde{y}}{\mathrm{d}t} \end{pmatrix} = \begin{pmatrix} \frac{\partial\tilde{x}}{\partial x} & \frac{\partial\tilde{x}}{\partial y} \\ \frac{\partial\tilde{y}}{\partial x} & \frac{\partial\tilde{y}}{\partial y} \end{pmatrix} \begin{pmatrix} \frac{\mathrm{d}x}{\mathrm{d}t} \\ \frac{\mathrm{d}y}{\mathrm{d}t} \end{pmatrix} \tag{3.1}$$

where optical flow is written as $\left(\frac{dx}{dt}, \frac{dy}{dt}\right)$. In constructing the Jacobian matrix, expressions for $\tilde{x}$ and $\tilde{y}$ must be defined only in terms of points on the picture plane.

Theorem 3.2.1. *The ground coordinates $(\tilde{x}, \tilde{y})$ of a point on the ground with image coordinates (x, y) are given by*

$$\tilde{x} = \frac{a(x - x_0)}{\cos\theta(b - (y - y_0))} \qquad \tilde{y} = \frac{a(y - y_0)}{b - (y - y_0)}$$

for some values a, b, angle θ, and points x_0, and y_0 on the image plane.

Proof. Suppose we have some object on the ground at the position $(\tilde{x}, \tilde{y}, 0)$. In the camera image, it will appear at position (x, y). We will solve the relation between the ground coordinates $(\tilde{x}, \tilde{y}, 0)$ and the image coordinates (x, y). Firstly, we will choose our coordinate system so that the point on the ground directly below the camera has

coordinates $(0, 0, 0)$, and the y-axis and z-axis of our coordinate system are projected into the same direction on the image plane.

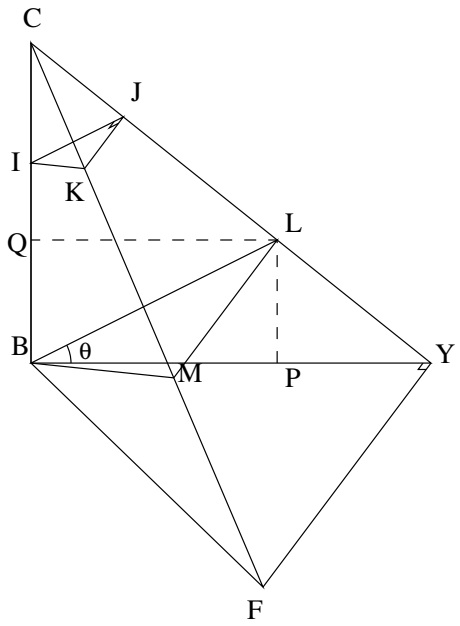

Figure 3.1: The camera is fixed at point C. Triangle IJK is within the pixel image plane p, or in other words the video data that is a representation of the scene. The triangle BYF is on the ground plane, or the reality of the scene.

Let C denote the position of the camera. Let B be the point on the ground (which we are assuming is flat and horizontal) vertically below C. Then $|CB| = h$ is the vertical height between the camera and the ground plane. Any point X on the ground can be thought of as projected onto a fixed image plane p, by taking the point of intersection of the line CX with p. Let I be the projection of B onto p.

We want to calculate the projection of a point F on the ground onto the plane p. We calculate the projection in two stages - first we calculate the projection of the point Y, which is the projection of F onto the y-axis. That is, if F has coordinates $(\tilde{x}, \tilde{y})$, then Y has co-ordinates $(0, \tilde{y})$. Then, the same is done for the x-axis, where $|YF|$ is equivalent to the value found by projecting F onto the x-axis and has coordinates $(\tilde{x}, 0)$. Let the projections of Y and F be J and K respectively. Point I in the image plane is given coordinates (x_0, y_0). J has coordinates (x_0, y) and K has coordinates (x, y) and therefore $y - y_0 = |IJ|$ and $x - x_0 = |JK|$.

We construct the triangle BLM parallel to IJK, and let $\theta = \angle LBY$ be the angle between the projection plane and the horizontal ground.

BL is parallel to IJ and LM is parallel to JK so the coordinates are proportional to BL and LM. Proportionality between triangle IJK and BLM is as follows.

$$\frac{|CI|}{|CB|} = \frac{|IJ|}{|BL|} = \frac{|JK|}{|LM|} \tag{3.2}$$

Now, let P and Q be the orthogonal projections of L onto BY and BC respectively (Figure 3.1 depicts this with the dotted lines). Then, triangle CBY can be broken into triangles CBL and LBY. Comparing total area gives

$$\frac{1}{2}|CB||BY| = \frac{1}{2}|CB||QL| + \frac{1}{2}|BY||LP| \tag{3.3}$$

Recall $y - y_0 = |IJ|$. Through equation (3.2), $|BL|$ can be expressed as

$$|BL| = \frac{|CB|}{|CI|}(y - y_0) \tag{3.4}$$

Substitute equation (3.4) into an expression for $|BY|$, the y-component at the reference point. Through θ, it is possible to represent $\tilde{y} = |BY|$ in terms of lengths relative to the camera C.

$$\tilde{y} = |BY| = \frac{|BL||CB|\cos\theta}{|CB| - |BL|\sin\theta}$$

$$= \frac{\frac{|CB|}{|CI|}(y - y_0) \cdot |CB|\cos\theta}{|CB| - \frac{|CB|}{|CI|}(y - y_0)\sin\theta}$$

Multiply the top and bottom by $\frac{|CI|}{|CB|\sin\theta}$ to assist in re-arranging, then let $a = |CB|\cot\theta$ and $b = |CI|\csc\theta$ to arrive at an expression for $\tilde{y}$

$$\tilde{y} = \frac{a(y - y_0)}{b - (y - y_0)} \tag{3.5}$$

An expression for $\tilde{x}$ is as follows. Recall that $|JK|$ is parallel to $|YF|$. So, the triangles CLM and CYF are similar and give

$$\frac{|YF|}{|LM|} = \frac{|CY|}{|CL|} = \frac{|CB|}{|CQ|} \tag{3.6}$$

which means that $|CQ| = |CB| - |QB| = |CB| - |BL| \sin\theta$. Plugging this in, as well as multiplying the top and bottom by $|CI| \csc\theta$ to simplify the expression gives

$$\tilde{x} = |YF| = |LM| \cdot \frac{|CB|}{|CB| - |BL| \sin\theta} \left(\frac{|CI| \csc\theta}{|CI| \csc\theta} \right) \tag{3.7}$$

From equation (3.2), $|LM| = \frac{(x-x_0)|CB|}{|CI|}$, also $|CB| \csc\theta = \frac{a}{\cos\theta}$. Substitute these expressions into (3.7) and get

$$\tilde{x} = \frac{a(x - x_0)}{cos\theta(b - (y - y_0))} \tag{3.8}$$

for some h, $|CI|$, θ, x_0, and y_0 where $a = h \cot\theta$ and $b = |CI| \csc\theta$. □

Corollary 3.2.1.1. *Populating the Jacobian with the partial derivatives gives*

$$\mathbf{J} = \begin{pmatrix} \frac{\partial \tilde{x}}{\partial x} & \frac{\partial \tilde{x}}{\partial y} \\ \frac{\partial \tilde{y}}{\partial x} & \frac{\partial \tilde{y}}{\partial y} \end{pmatrix} = \begin{pmatrix} \frac{a}{\cos\theta(b-(y-y_0))} & \frac{a(x-x_0)}{\cos\theta(b-(y-y_0))^2} \\ 0 & \frac{ab}{(b-(y-y_0))^2} \end{pmatrix} \tag{3.9}$$

3.2.1 Solving for unknown variables

To solve for the unknown variables set forth by Theorem 3.2.1, further constraints as well additional information are required. As an additional constraint, we choose origin of the image plane's coordinate system to be the bottom centre of the frame with coordinates $(0, 0)$ and as a result, $x_0 = 0$. The new information we are bringing in is a few real-life measurements of a reference object from the actual location. Pixel measurements of that same object as it is represented in the image are leveraged using geometry and proportionality to solve for velocity on the ground. The two points being taken from the reference object on the image plane are (x, y) and (x', y').

The reference object in question is a seating sign towards the right of the frame. In the actual location, the horizontal distance from the camera to the sign is 490 inches, the height of the camera from the ground is 158 inches and a measurement of the pole from the ground to the bottom of the sign in 116 inches. These values

Figure 3.2: Location of the points being taken from the reference object in a still of the video clip.

are used along with what we know about the geometry of the environment to build a system of equations that can be used to solve for the necessary variables to calculate speed.

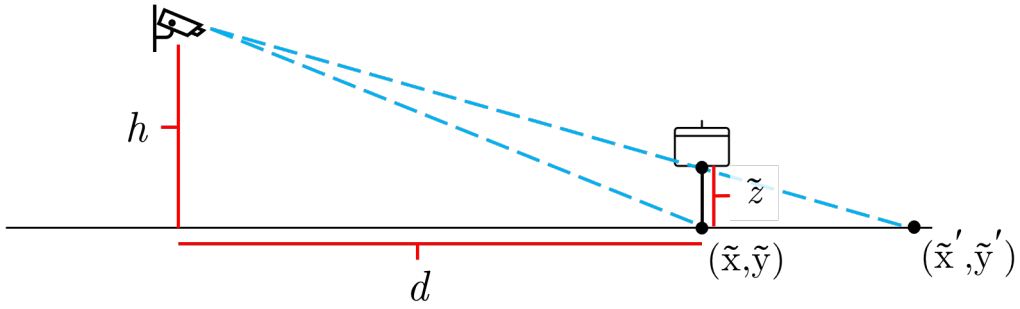

Figure 3.3: Diagram of the relationship between the camera and the reference object.

Let the coordinates of the sign on the ground be $(\tilde{x}, \tilde{y})$, the vertical height from the ground to the bottom of the sign be $\tilde{z}$, and the height of the camera $|CB| = h$. The line from the camera to the bottom of the sign (above the ground) extends to meet

the ground at $(\tilde{x}', \tilde{y}')$. By similar triangles, $\frac{\tilde{x}'-\tilde{x}}{\tilde{x}'} = \frac{\tilde{z}}{h}$ and $\frac{\tilde{y}'-\tilde{y}}{\tilde{y}'} = \frac{\tilde{z}}{h}$. Isolating for $\tilde{x}'$ and $\tilde{y}'$ gives $\tilde{x}' = \left(\frac{h}{h-\tilde{z}}\right)\tilde{x}$ and $\tilde{y}' = \left(\frac{h}{h-\tilde{z}}\right)\tilde{y}$.

The coordinates (x, y) are the point K in the image plane representing the place where the pole of the sign is on the ground. The coordinate corresponding to the bottom of the sign directly above that point is written as (x', y'). This relationship results in the following equations

$$\frac{h}{h-\tilde{z}} \cdot \tilde{x} = \frac{a(x'-x_0)}{\cos\theta(b-(y'-y_0))} \tag{3.10}$$

$$\frac{h}{h-\tilde{z}} \cdot \tilde{y} = \frac{a(y'-y_0)}{b-(y'-y_0)} \tag{3.11}$$

Also,

$$\tilde{x}^2 + \tilde{y}^2 = BF^2 = d^2 \tag{3.12}$$

where d is the horizontal distance from the camera to the seating sign.

Recall from Theorem 3.2.1 that $a = h\cot\theta$ and $b = |CI|\csc\theta$, as well as $x_0 = 0$ from the additional constraint. This leaves four unknowns; θ, $|CI|$, h, and y_0.

From (3.5) and (3.8), $\frac{\tilde{x}\cos\theta}{\tilde{y}} = \frac{x-x_0}{y-y_0}$; similarly (3.10) and (3.11) gives $\frac{\tilde{x}'\cos\theta}{\tilde{y}'} = \frac{x'-x_0}{y'-y_0}$. Thus

$$xy' - yx' = x_0(y'-y) + y_0(x-x') \tag{3.13}$$

and y_0 is isolated and found to be -8644. θ can be solved for using the above equations. Substituting equations (3.5) and (3.8) into (3.12) and isolating θ to one side results in

$$\frac{1}{\cos\theta} = \sec\theta = \frac{b-(y-y_0)}{a(x-x_0)}\sqrt{d^2 - \left(\frac{a(y-y_0)}{b-(y-y_0)}\right)^2}$$

$$\theta = \cos^{-1}\left(\sqrt{\frac{\frac{d^2}{h^2} - \left(\frac{(x-x_0)}{b-(y-y_0)}\right)^2}{\left(\frac{d^2}{h^2} + \frac{(y-y_0)^2}{(b-(y-y_0))^2}\right)}}\right) \tag{3.14}$$

Now using equations (3.5) and (3.11), it is also possible to express b in terms of known values. Let $\frac{h}{h-\tilde{z}} = H$.

$$b = \frac{(y' - y_0)(1 - \frac{1}{H})}{\left(1 - \frac{(y'-y_0)}{H(y-y_0)}\right)} \tag{3.15}$$

Recall $d = 490$ inches, $h = 158$ inches and $\tilde{z} = 116$ inches. Therefore $\frac{h}{h-\tilde{z}} = H \approx 3.761905$.

The dimensions of each frame is 1280×1706 pixels. Recall that in the image the coordinate space, $x_0 = 0$ and $y_0 = -8644$. Pixel coordinates for (x, y) and $(x^{'}, y^{'})$ were located within a still of the video. Thus, $x - x_0 = 263$, $y - y_0 = 9468$, $x' - x_0 = 267$, and $y' - y_0 = 9612$. The remaining unknowns are then solvable and calculated to be $\theta = 1.48551$ in radians, $b = 9742.536$, and $a = h \cot\theta \approx 13.50805$.

3.2.2 Computing speed

Recall equation (3.1) and (3.9).

$$\begin{pmatrix} \frac{\mathrm{d}\tilde{x}}{\mathrm{d}t} \\ \frac{\mathrm{d}\tilde{y}}{\mathrm{d}t} \end{pmatrix} = \begin{pmatrix} \frac{\partial\tilde{x}}{\partial x} & \frac{\partial\tilde{x}}{\partial y} \\ \frac{\partial\tilde{y}}{\partial x} & \frac{\partial\tilde{y}}{\partial y} \end{pmatrix} \begin{pmatrix} \frac{\mathrm{d}x}{\mathrm{d}t} \\ \frac{\mathrm{d}y}{\mathrm{d}t} \end{pmatrix}$$

$$\mathbf{J} = \begin{pmatrix} \frac{\partial\tilde{x}}{\partial x} & \frac{\partial\tilde{x}}{\partial y} \\ \frac{\partial\tilde{y}}{\partial x} & \frac{\partial\tilde{y}}{\partial y} \end{pmatrix} = \begin{pmatrix} \frac{a}{\cos\theta(b-(y-y_0))} & \frac{a(x-x_0)}{\cos\theta(b-(y-y_0))^2} \\ 0 & \frac{ab}{(b-(y-y_0))^2} \end{pmatrix}$$

The squared ground speed for a given pixel and time point is

$$\tilde{s}^2 = \left(\frac{d\tilde{x}}{dt}\right)^2 + \left(\frac{d\tilde{x}}{dt}\right)^2 = \left(\frac{dx}{dt}, \frac{dy}{dt}\right) \mathbf{J}^{'}\mathbf{J} \begin{pmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{pmatrix} \tag{3.16}$$

Substitute the known variables into $\mathbf{J}^{'}\mathbf{J}$ to get

$$\mathbf{J}^{'}\mathbf{J} = \frac{h^2}{\sin^2\theta(b-(y-y_0))^2} \begin{pmatrix} 1 & \frac{x-x_0}{b-(y-y_0)} \\ \frac{x-x_0}{b-(y-y_0)} & \frac{(x-x_0)^2+b^2\cos^2\theta}{(b-(y-y_0))^2} \end{pmatrix}$$

Now that $\mathbf{J}^{'}\mathbf{J}$ is in terms of known values, plug it back in to get $\tilde{s}^2$

$$\tilde{s}^2 = \frac{h^2}{\sin^2\theta(b-(y-y_0))^2} \cdot \left(\frac{dx}{dt}, \frac{dy}{dt}\right) \begin{pmatrix} 1 & \frac{x-x_0}{b-(y-y_0)} \\ \frac{x-x_0}{b-(y-y_0)} & \frac{(x-x_0)^2+b^2\cos^2\theta}{(b-(y-y_0))^2} \end{pmatrix} \begin{pmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{pmatrix} \tag{3.17}$$

For each time point t, $\mathbf{J}'\mathbf{J}$ is defined as a function of (x, y) and is multiplied by the optical flow vector corresponding to that coordinate as outlined in equation (3.17). The square root of this is taken as the ground speed for that particular time and location. Note that $\mathbf{J}'\mathbf{J}$ for each pixel (x, y) is a time invariant transformation, thus can be calculated once and stored as a list. With an implementation using GPUs, calculating the transformed speed can be done in real time due to the fact that the optical flow calculation itself is already able to be done in real time.

3.3 Correcting the Tilted Camera

Another issue in the representation of the scene is that the camera appears as though it is tilted. Lines in the scene that are known to be vertical in reality are not upright in the frame, particularly on the right-hand side. Since the scene is in three-point perspective, it is expected that there would be some distortion of straight lines. However with the assumption that the camera is positioned in the centre of the frame, the lines in the centre of the frame should be vertical which is not true in this case and therefore it is reasonable to assume that there is some tilt present. It may be reasonable to deduce that a crooked line in the centre of the frame could mean that the camera is not positioned in the centre of the frame but rather to the left. The assumption that $x_0 = 0$ was necessary for calculating the parameters of the transformation from the co-ordinates of the signpost. Approximating the camera as being in the centre of the frame and then rotating the image slightly counter-clockwise restores this assumption.

Determining the angle of rotation is done first through identifying objects that would have vertical lines in reality such as football goal posts, the seating signs, and upright structures. If the camera is assumed to be tilted then it would follow that all of the lines should tilt at the same angle. This logic is somewhat complicated by three-point perspective which slightly distorts all straight lines, vertical or otherwise. An additional assumption is made: the lower vanishing point is assumed to be far enough down to be treated as infinity, therefore assuming that vertical lines are parallel in the frame. The pixel coordinates of two end-points from each of the tilted objects

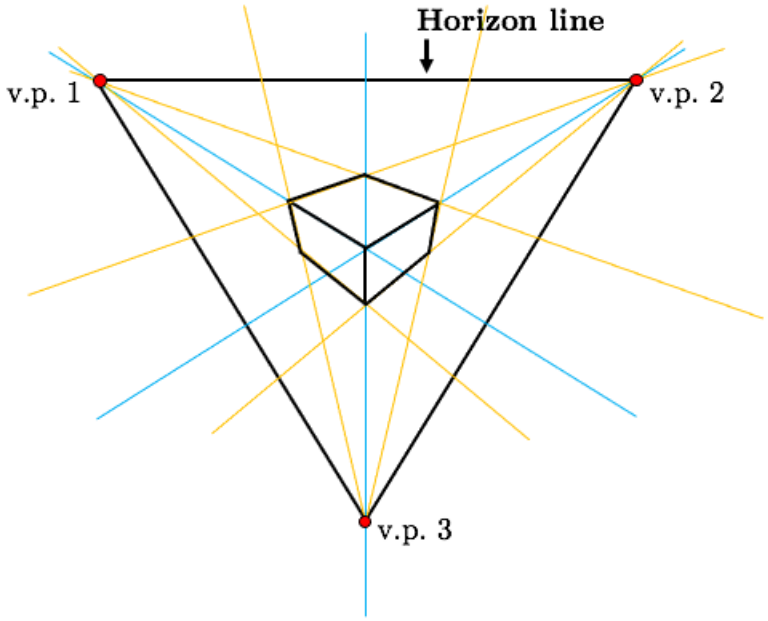

Figure 3.4: A cube rendered in three-point perspective. Three vanishing points create a triangle however it does not necessarily need to be equilateral as is seen here. The line between vanishing points one and two is the horizon line.

being used are identified. Then, the slope between those two points is calculated and used as that object's angle of rotation. The mean of all of the angles is calculated and used as the final angle of rotation to be applied to the entire frame. This value ended up being 10.33° and rotates all frames at all time points in a counter clockwise direction to reverse the tilt.

Rotating the frame is as simple as using a rotation matrix on the pixel coordinates. That is,

$$\begin{bmatrix} x_{new} \\ y_{new} \end{bmatrix} = \begin{bmatrix} \cos\gamma & \sin\gamma \\ -\sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{3.18}$$

where $\gamma = 10.33° \approx 0.1802925$ rad is the angle of rotation. Each optical flow matrix is rotated by multiplying the coordinates of each pixel by the rotation matrix. The rotation is done after the calculation of optical flow since the relationships between pixels and their values are not expected to change to a significant degree. The same cannot be said for ground speed, however. Because the frame is rotated, the coordinates of the reference points (i.e. the seating sign) are altered which results in different values being used to calculate ground speed and so the coordinates of the reference object

were found from a rotated version of Figure 3.1 and used in the above ground speed calculation.

3.4 Post-Transformation Results

A point that has not yet been noted is that the horizon line is within the field of view of the scene. A horizon line is a line in an image from which the distance appears to be infinitely far from the viewer (e.g. a sunset over the ocean). In the context of the ground speed transformation, an infinite distance from the camera would translate to an infinitely large speed. This is of course impossible and indicates that the ground speed transformation is only accurate up to a certain distance. Moreover, the projection of optical flow is onto the ground plane, which in this case corresponds to the platform in the foreground. The background areas do not depict moving objects on the ground and as such should not be taken into consideration when evaluating the efficacy of the transformation. Included in the extraneous background information is the "horizon line". We will assume that the transformation is applicable for the entire platform (which includes the location of the fight), and that the regions where the horizon line is present can be excluded. As is seen in the right-hand column of Figure 3.5, the horizon line is near the top of the frame and has extremely high values compared to the rest of the scene.

The rotation of the frame and the conversion to ground speed should both be assessed. The purpose of these two transformations is to best estimate the reality of the speed in the scene. In other words, the transformations were done to account for the camera's "misrepresentations" of perspective and tilt in an attempt to achieve a consistent estimation of the true speed throughout the entire frame. Figure 3.5 depicts a heat map of calculated speeds averaged over the first 850 frames and compares all conditions. It is examined visually as a preliminary judgement of the transformation's success.

When looking at the non-rotated optical flow image (top left in Figure 3.5), the average speed across non-fighting frames appears to be slower at further distances. The lower right of the frame shows a comparable average speed to the centre of the frame. In reality, traffic is less frequent and more sparse in the bottom right region compared to the centre. So, it is expected that this area would have a smaller optical flow on average compared to the centre of the scene. These observations support the idea that perspective is distorting the calculation of optical flow.

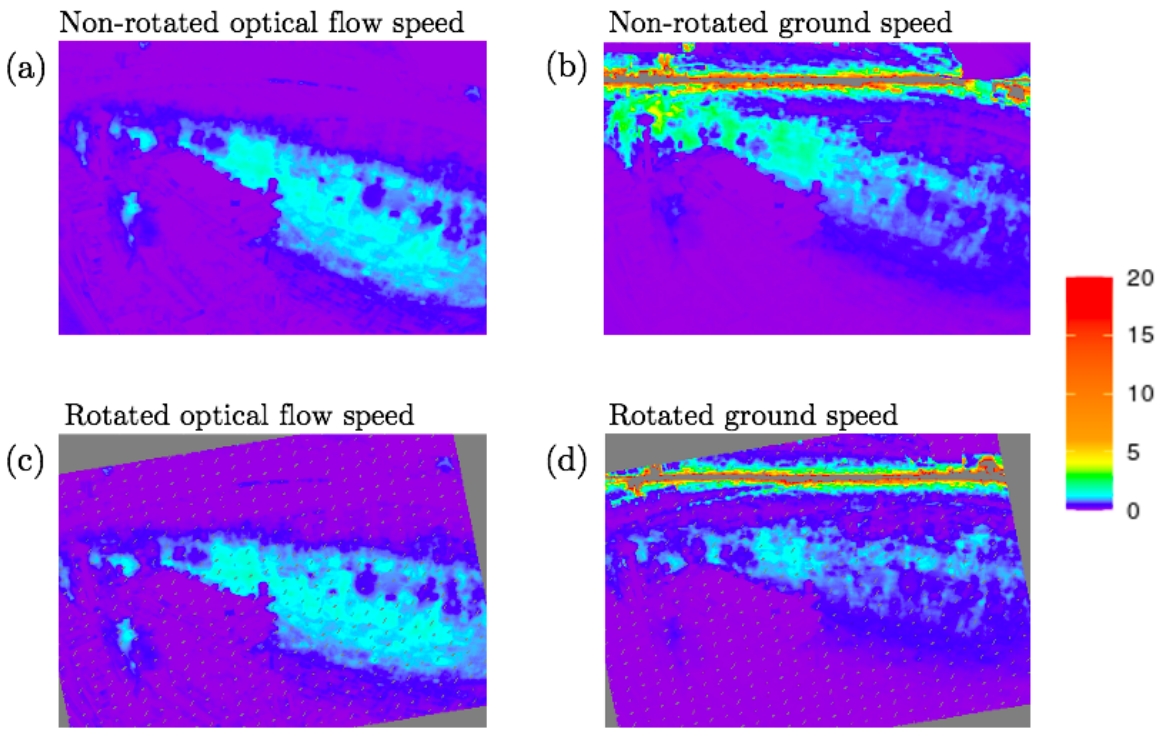

Figure 3.5: The left-hand column shows optical flow (with and without rotation) averaged across the first 850 time points. The right-hand column shows the same for the ground speed.

When looking at the ground speed calculation for rotated and non-rotated data (right-hand column of Figure 3.5), both lower right regions of the frame appear to have smaller values, and in the non-rotated data set it is clear that the upper left quadrant has much larger values. The non-rotated data set shows that although the ground speed did attempt to compensate for the distance through enlarging "further" values and shrinking closer values, it failed in the ultimate goal of the transformation which was to make the regions comparable in value. In the rotated ground speed data however, the transformation is much more reasonable. The values closer to the camera are smaller and the values further away from the camera appear to be slightly larger on average which is in keeping with expectations knowing the crowd behaviour throughout the clip.

It is concluded that the rotation was a necessary transformation to the data if the ground speed is to be used to model the data. If optical flow is used, the frame rotation would have little to no consequence seeing as the values are not affected. Only post-rotated data sets will be used moving forward.

Visual inspection of the results show evidence to suggest that the ground speed transformation was successful in achieving some balance between different areas in the scene. To finalize these conclusions, prediction results for the optical flow and ground speed data are compared in the model evaluation section to confirm relevancy. Since the background is not of interest and contains the erroneous data, the top 306 rows are immediately excluded from the scene resulting in a 974×1706 pixel image for each frame. Further refinement to eliminate irrelevant data is included in the next chapter.

Chapter 4

Finding Features

Now that gray-scale pixel data has been transformed to optical flow then again to ground speed, informative features can be extracted from the data. The field of computer vision has found many ways to extract features from large data sets. Due to fact that the data in the current project features a crowded scene with many moving objects, a global approach to feature extraction is preferred due to its high-level approach. In contrast, feature based methods are not ideal for densely crowded scenes since there are too many interest points to be tracked through time in an efficient manner (Huang & Chen, 2014). Recall that global methods are a type of feature extraction where data is extracted from the entire frame and then important information is summarized, as opposed to local methods which identify important information and then track it through time. Most feature extraction techniques (both global and local) not only require large amounts of training data but also a high level of expertise to implement. With that in mind, the typical procedure of comparing the current novel methods with popular and established methods, as well as the use of benchmark data sets are beyond the scope of the current project. Furthermore, there is an opportunity to develop a simple approach to feature extraction, one that could be generalizable to other data sets.

In addition to summarizing the data, feature extraction also aids in the reduction of the dimensionality within a data set. High dimensional data is undesirable for a number of reasons. Firstly, it often results in over-fitting of the model and in turn leads to a model which is not generalizable. As the number of features increases, the number of observations required for the model to be sufficiently generalizable increases exponentially. This is known as the *curse of dimensionality*; there is an upper bound

on how much accuracy more features can provide. This is just one of the many ways that the curse of dimensionality manifests itself (Hastie et al., 2009).

Some feature extraction has already taken place for the current data set as optical flow can be considered a form of feature extraction, one that extracts motion from gray-scale pixels. As immensely useful as this is, the data set could benefit from further work to reduce dimensionality and better summarize the movement.

The method with which features are extracted is as follows. First each frame is separated into non-overlapping local areas according to a specific layout. Then, within each local area, various statistics (features) are calculated for all time points. The features calculated are meant to capture the movement within their respective local areas. Similar methodology has been explored by Huang & Chen (2014). Once all features are calculated, they undergo feature selection using random forest in order to choose those that are most valuable to the model. The two main goals of feature selection are to reduce the feature set and find relevant features, the former ends up contributing to the reduction of the dimension of the data set.

4.1 Dividing the Frame into Local Areas

Finding some way to divide the frame into a number of regions is of the utmost importance when considering dimension reduction. If each pixel is treated as an individual feature, it implies that the data frame would have a 1280×1400 feature set. Not only would this data be high dimensional but would also be an inefficient use of information. Assembling data into features allows said features to be engineered with the specific intent to summarize information relevant to the task at hand. A simple and intuitive approach is taken to extract features from the current data set. That is instead of following individual pixels through time, they are replaced with a statistic or some other function calculated from a group of pixels confined to a local area. The size of this feature set then depends on both the number of local areas dividing the image and the number of functions chosen to be extracted from those local areas.

A naïve approach is taken to partition the frame into local areas. The frame is divided into non-overlapping cells, analogous to placing an arbitrary grid over the frame. The local areas should not be too small as to create volatile statistics which would be too variable to find meaningful patterns. At the same time local areas that are too large should be avoided because they include too many objects, resulting in non-descriptive

features. Larger local areas are desirable if possible, since it would correspond to a fewer number local areas dividing the frame, translating to less variables making up the overall feature set.

The current project seeks to recognize the movement of people fighting and so the size chosen for the local areas should be able to capture those type of movements. With the perspective differences in the scene, the number of pixels each person occupies depends on their location in the frame and sizing of the local areas also needs to be adjusted for this fact. In general, it is ideal for only one person to be within a local area at any given time so that it is not measuring separate motions by different actors. It is less of a concern whether or not a fraction of a person (e.g. only a torso) is within a local area, as long as the areas are not so small that their patterns no longer offer meaningful interpretation. These goals must be balanced by a desire for simplicity as a huge advantage of partitioning the frame is its quick and easy nature. If the signal is sufficiently large then the local area separation scheme should be robust to similar designs. Given these considerations, two separate layouts are designed. The first (Figure 4.1) divides the frame into two areas whereby the top section is divided into 50×50 pixel squares and the bottom section is 100×100 pixels. The particular dimensions were chosen through some trial and error in such a way that the smallest people underneath the grid are at least divided in half.

A second grid (Figure 4.2) is placed over the frame as a kind of additional experiment. In this scheme, the size of each cell increases with each new row. The smallest and largest dimensions are similar to the first grid size, beginning with 43×43 pixel squares and growing to 104×104 squares. The intuition here is that the local areas sizing should mimic a continuous function of distance similar to the changes in the perspective of the objects in the scene.

The first layout (grid scheme #1) allows for easier direct comparisons between local areas, while the second step-sized grid is a more intuitive option with regards to perspective distortions. If results are similar in both layouts then there is evidence to suggest that accuracy is resilient to grid layout. Grids were chosen to be non-overlapping to avoid a larger number of features. Each cell in the grid is numbered to keep track of the spatial relationships in the scene when undergoing feature selection later on. A numbered version of both grids can be found in Appendix B.

So, within each local area is a set of spatially-related data points. Although a feature is technically tied to both a statistic and a local area, the term "feature" may be used

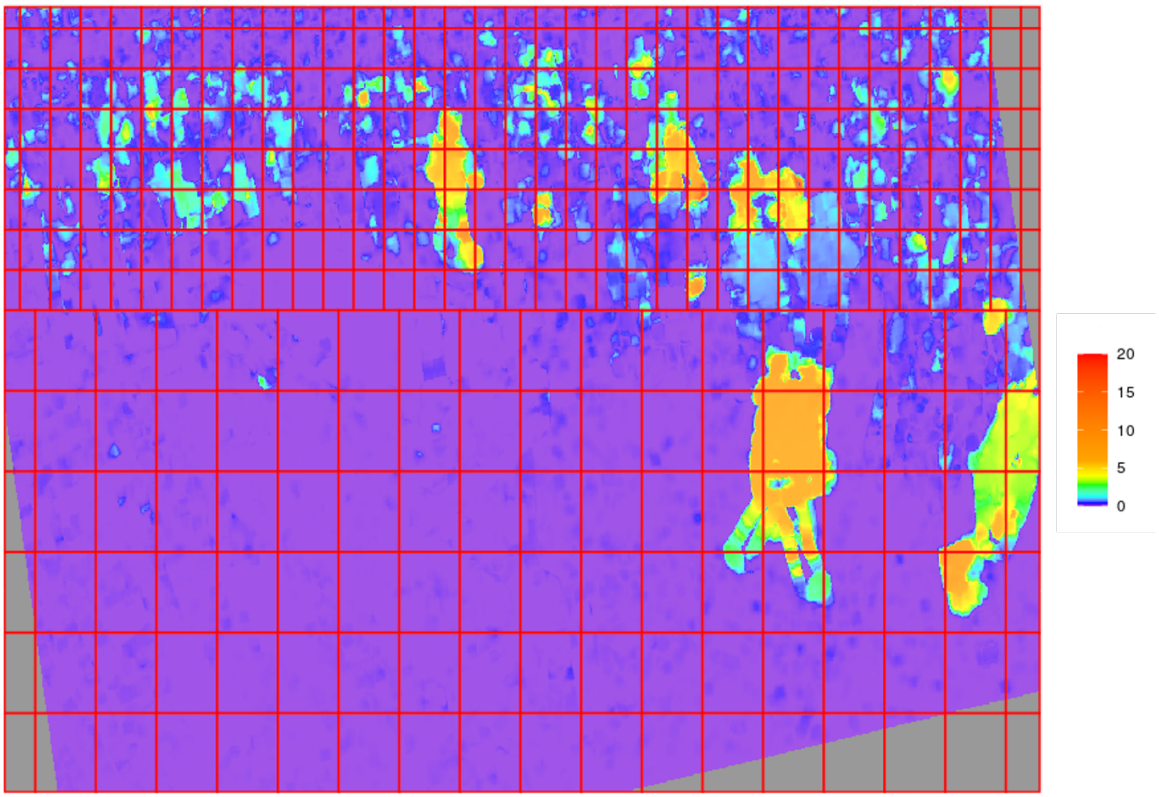

Figure 4.1: The first grid layout which uses only two different cell sizes to divide local areas.

to only describe the statistic which would be applied over all local areas. The "full" feature set consists of a number of statistics calculated in each local area. That is, if there are 5 statistics calculated over 100 local areas then there are 500 features in the full feature set.

4.2 Extracting Features

Deciding what kind of features to extract is a somewhat subjective and arduous process. There may be some intuition behind the kind of features that have the potential to be richly descriptive however it varies depending on the particular data set. Operating under the assumption that anomalous movement is markedly different in some way compared to "typical" movement inside the scene, it follows that a difference could be observed between a distribution of "normal" movement vs. a distribution of "fighting" movement. If it is correct to assume that these distributions are different, then calculating statistics is a way to characterize, describe, and compare those distributions.

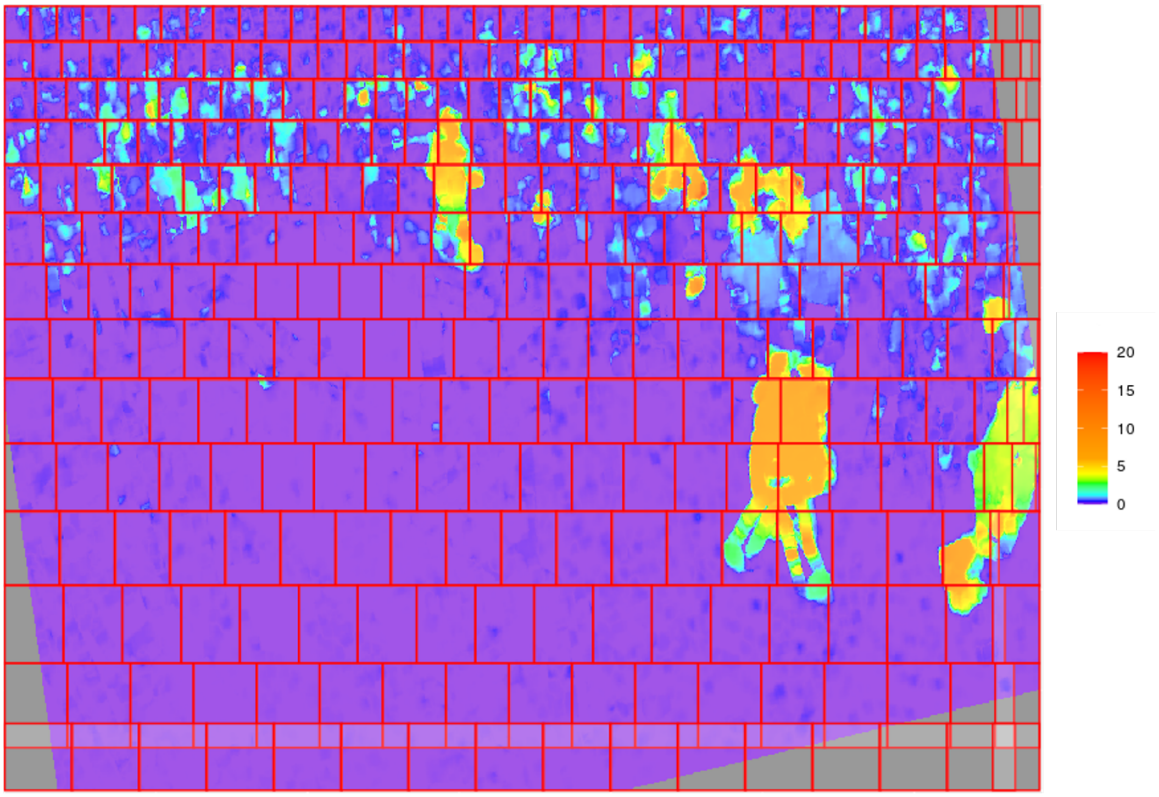

Figure 4.2: The second grid layout uses a step-sizing approach where the local areas are of larger dimension the lower they are in the frame.

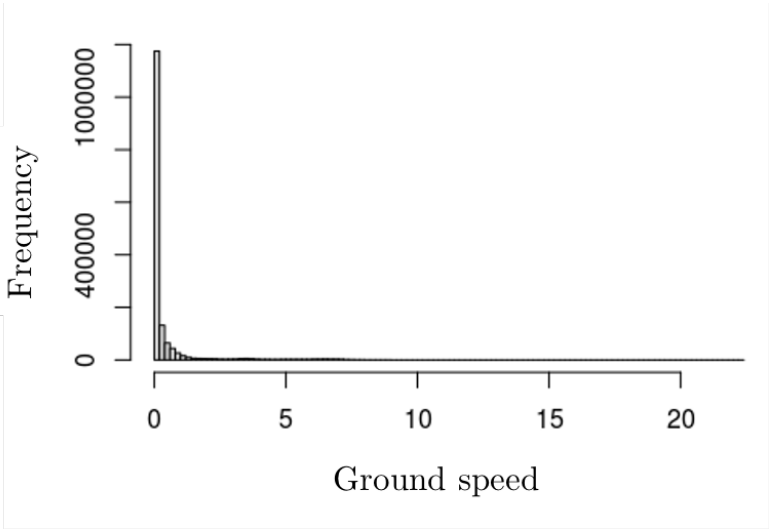

Figure 4.3: Histogram of ground speed values that took place prior to the fighting.

The overall distribution of all local areas and the frame in general is very heavily skewed to right; the majority of ground speed values are near zero (no movement) regardless of the local area in the frame, and whether or not fighting is happening.

A strong skew is without a doubt a characteristic to be mindful of during the feature extraction process and ends up generating some difficulty in finding statistics that quantify the delicate nuances between local areas. With this in mind, four different strategies for extracting our features were tried: direct statistics, the local area's relationship to its surrounding region, truncation and linear functions of statistics.

4.2.1 Direct Summary Statistics

In keeping with a preference for simplicity, the mean, variance and 95^{th} percentile are calculated for each local area. The mean and variance are measures of the centre and spread of the distribution respectively. The 95^{th} percentile is chosen because it is of interest to observe the tail of the distribution; ground speed values are heavily skewed and so the 95^{th} percentile provides some insight into the most extreme end of the distribution. If the 95^{th} percentile is very low it is likely that no movement took place in that area. If it is very high then it can be inferred that at least some proportion of the data is within the movement range.

4.2.2 Truncated Features

If a large proportion of pixels are non-moving, removing pixels with low speeds from statistical calculations could provide a more focused insight for moving pixels. A number of cutoff values are used to truncate the data (0, 0.01, 0.1, 0.3, 0.7, and 1). The mean of this truncated data is used as a feature. A risk with this method is that for certain local areas, it is possible that most if not all of the pixels could fall below the given threshold causing insufficient data. To account for this, features where there are missing values for more than 50% of the time points were excluded from the analysis.

4.2.3 Ratio between Two Statistics

In the event that the data is not generalizable across the entire frame, comparing a given local area to its surrounding pixels could prove to be useful in predicting anomalous behaviour. If a statistic calculated is large given its location in the frame then it could be considered a valuable predictor in spite of it being unremarkable in value on a global scale. To quantify such a concept, the ratio between two of the same statistic were taken such that the only difference was the number of pixels that were used in the calculation. That is, for a given local area as previously defined (and

shown in Figures 4.1 and 4.2), a square in the same location but ten times its size is also calculated. Therefore, the local area is nested within the centre of its larger square. For each time point the statistic calculated from the local area is divided by the same statistic but calculated using the larger square, creating the new feature type. If this ratio is around one then the behaviour within that local area is not particularly anomalous to the region. If it is less than one then it is particularly inactive, and if it is much greater than one then the activity would be greater than what is typical given the larger region. This scheme could also help in accounting for varying crowd densities in different areas of the frame. For example, a densely packed area usually has constant motion and its mean is higher compared to an area with less frequent motion. This higher mean doesn't necessarily imply that there is fighting taking place, as it is typical given the context of the region.

4.2.4 Multi-features

Finally, a couple of features were some combination of at least two of the above mentioned techniques. All features are specified in the following table:

#	Desc.	Type
1	95^{th} percentile	Direct
2	Means	Direct
3	Variance	Direct
4	Ratio of 95^{th} percentile	Ratio
5	Ratio of mean	Ratio
6	Ratio of variances	Ratio
7	Mean truncated at 0	Truncated
8	Mean truncated at 0.01	Truncated
9	Mean truncated at 0.1	Truncated
10	Mean truncated at 0.3	Truncated
11	Mean truncated at 0.7	Truncated
12	Mean truncated at 1	Truncated
13	Ratio of proportion above the mean	Multi
14	Ratio of the mean, truncated at the median	Multi
15	Proportion of values above the mean	Multi
16	Mean of values above the mean	Multi
17	Ratio of the mean truncated at 0	Multi
18	Ratio of the mean truncated at 0.01	Multi
19	Ratio of the mean truncated at 0.1	Multi
20	Mean of values above the 75^{th} percentile	Multi
21	Mean of values above the 75^{th} percentile	Multi

Table 4.1: List of the full set of statistics calculated within each local area at all time points. The third column tags each statistic with its corresponding feature type described in this section.

4.3 Narrowing Down the Feature Set

Feature selection is the process by which a subset of features are thoughtfully taken from the full feature set. It is an important step because it allows the dimensionality to be reduced even further by only using the most relevant features and is an interim step between extracting meaningful features and assessing the model's predictive accuracy. For the current data set the feature space is further reduced, first through excluding irrelevant local areas and then eliminating collinear features. A random forest algorithm is then used for feature selection.

Because large swaths of the frame contain little to no possibility of movement, it is in the best interest of the model to eliminate features from these local areas. Examples of non-moving areas would be the roof of the hot dog stand located left of centre in the frame, or the football field and stands which take up the upper right corner of the frame. To determine which of the local areas are in this category, the 95^{th} percentile withing each local area and averaged across time is calculated. Then all features from local areas below a 0.5 threshold are excluded from the feature set. Local areas were compared against a still frame in the video to confirm that there was little to no possibility of any movement being able to take place. The football field and seating in the background were not selected by the cutoff method but are also considered to be irrelevant to the model as these areas are not where people could reasonably be perceived to be fighting (i.e. not on the platform) and were therefore removed manually. A similar procedure could be replicated in other data sets, however it should be noted that it is not entirely necessary to remove non-moving local areas since they would be presumably be ignored during the feature selection process. Nonetheless it can be helpful as a means of dimensionality reduction.

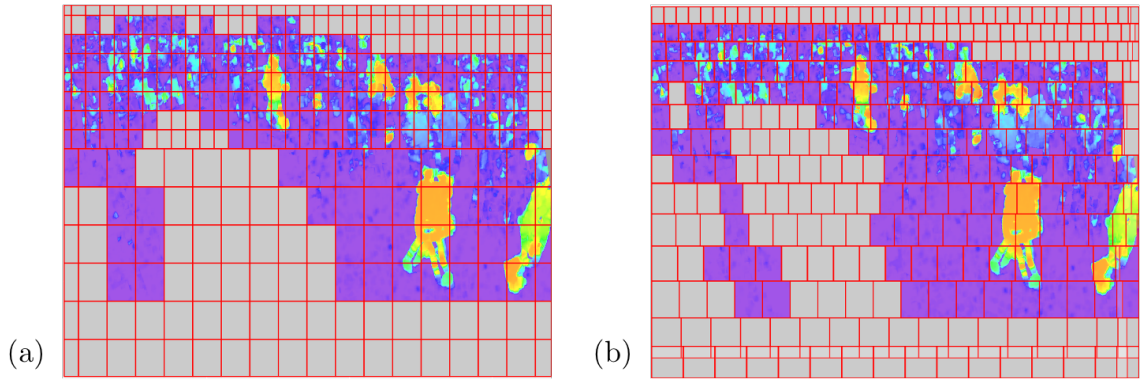

Figure 4.4: Grid layouts #1 (a) and #2 (b) with local areas in grey containing irrelevant information to the analysis. There is some variation between the two but both are mostly similar

Collinearity refers to the presence of a correlation between predictors. For random forest models, highly correlated variables tend to be ranked with similar importance. If a top ranked feature is highly correlated with other features then they are be ranked with high importance. As a consequence, lower-ranked but perhaps equally important features are taken as less important even if they provide valuable insight. Despite this draw-back for model interpretation, the presence of collinear features does not have an impact on predictive accuracy of random forest (Boulesteix et al., 2012). As such, some care is taken to eliminate highly collinear features however it is not a detriment

to the model accuracy if some collinearity is retained in the final feature set.

Determining which features are collinear is not completely straightforward. A statistic could be highly correlated to another statistic in one local area and then the two could be completely unrelated in another. Thus collinearity is adjusted for only generally. For each statistic calculated, its mean is calculated over all remaining local areas at each time point. Then a Spearman's correlation matrix is used to figure out which statistics should be excluded. If features are highly correlated then it is somewhat arbitrary which should be excluded, as theoretically either one would end up producing the same results. In keeping with Occam's Razor, among a group of highly correlated features only the simplest statistic is kept. To maintain a certain level of comparability between the two grid types, the cutting of the feature set was performed using the first grid layout and that list of features is used for both the second, step-sized grid and the optical flow data later on.

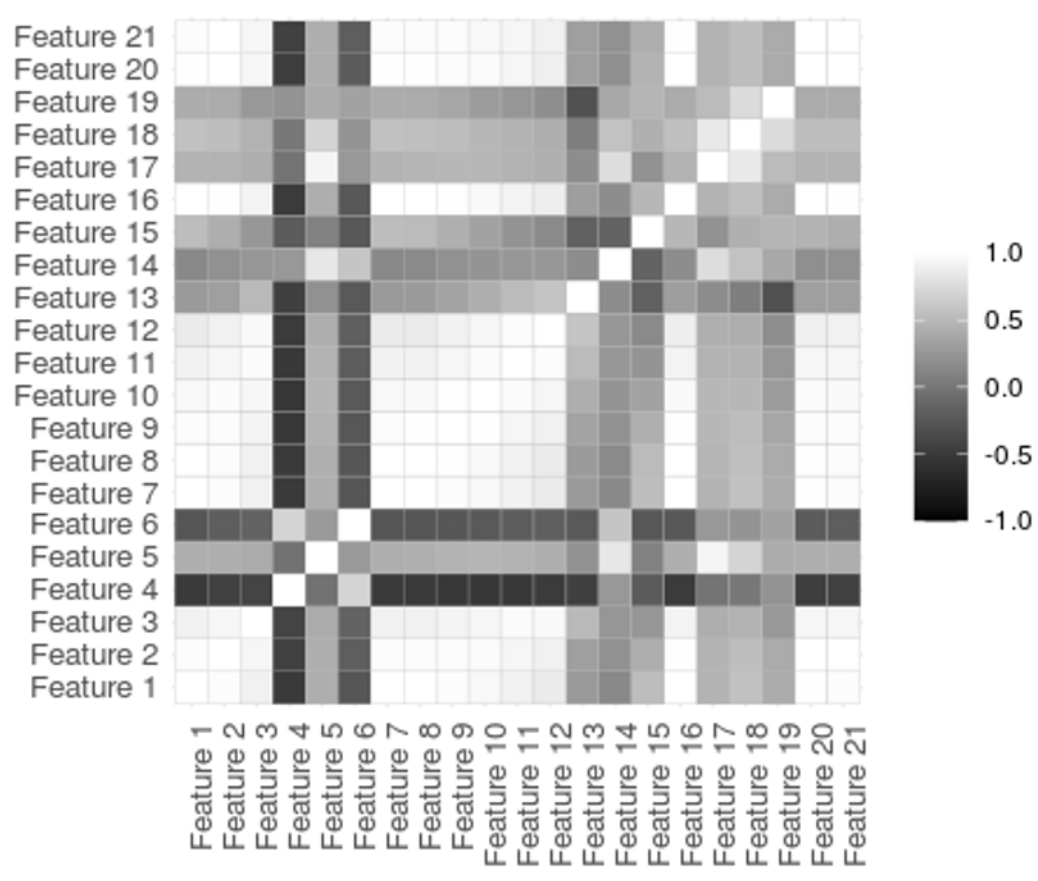

Figure 4.5: Correlation matrix including each of the statistics in Table 4.1.

The following table includes the final set of statistics after checking for collinearity.

#	Desc.	Type
1	95^{th} percentile	Direct
4	Ratio of 95^{th} percentile	Ratio
5	Ratio of mean	Ratio
6	Ratio of variances	Ratio
12	Mean truncated at 1	Truncated
13	Ratio of proportion above the mean	Multi
15	Proportion of values above the mean	Multi
19	Ratio of the mean truncated at 0.1	Multi

Table 4.2: List of the final set of statistics used in the random forest modeling.

4.4 Random Forest and Feature Selection

Feature selection is done using a random forest model. Random forest is a nonlinear, nonparametric, tree-based statistical model that can be used for both classification and regression. Tree-based methods involve dividing, or segmenting the predictor space into sub-regions, otherwise referred to as growing a decision tree (James et al., 2013). A decision tree recursively partitions the data set based on its features, optimizing for information gained. A decision tree in itself is a simple and highly intuitive method for modelling data. However results are not generalizable because they tend to over fit the data from which the model is trained. So ensemble methods like bagging, boosting, and random forest have been introduced to curb such effects. Ensemble methods use tree-averaging to address the issue of generalizability at the cost of interpretability, as an averaging over many trees does not lead to an intuitive interpretation of features the same way that a single decision tree does.

4.4.1 Random Forest Algorithm

In *The Elements of Statistical Learning*, Hastie et al. (2009) provide a clear description of the algorithm, which is re-stated and expanded upon here. The version used for classification will be presented however it can be easily be modified to suit a regression situation:

Because it is nonparametric, random forest is invariant to monotonic transformations and therefore scaling the features is not necessary as would be the case in some other classification methods. Another advantage is that there are reasonably few

Algorithm 1 Random Forest

In a training data set with N observations and p predictors,

1. Iterating over $b = 1$ to B,
   a) Draw a bootstrapped sample of size N from the training data
   b) On this data set, grow a random forest tree, T_b. That is,
      (i) Select m variables at random among the p available
      (ii) Determine the best split among the subset m
      (iii) Split the data set according to (ii)
      Repeat until there are no longer any splits available, or the pre-specified minimum node size has been reached.
2. Now with the resulting B trees, $\{T_b\}_1^B$, let $\hat{C}_b(x)$ be the estimated class of the bth random forest. Then, for a new point x_{new}, $\hat{C}_{rf}(x_{new})$ = majority vote among $\hat{C}_1^B(x_{new})$.

hyperparameters that require tuning (i.e., m and minimum node size) compared to other models. m is typically chosen to be $\sqrt{p}$ and a minimum node size of 1 in the case of classification although further tuning may be required (Breiman, 2001). The random forest algorithm also ranks features based on information loss when a certain variable is excluded. In the current data, this is leveraged to perform feature selection through the mean decrease in Gini index.

The Gini index is an alternative to the classification error rate, as it tends to not be "...sufficiently sensitive to tree growing." (James et al., 2013). In *An Introduction to Statistical Learning*, James et al. (2013) define both terms. The classification error rate E is

$$E = 1 - \max_k(\hat{p}_{rk})$$

where k is the class, r is the region considered, and $\hat{p}_{rk}$ is the proportion of observations in region r assigned to class k. The Gini index G is defined as

$$G = \sum_{k=1}^{K} \hat{p}_{rk}(1 - \hat{p}_{rk})$$

and is a "... measure of total variance across classes" (James et al., 2013). It is also a measure of node purity, the level of agreement of a classifier. The Gini index lies between 0 and 1 and smaller values correspond to a higher node purity (larger proportion of votes going to the predicted class). The mean decrease in the Gini index is a measure of variable importance which takes the mean of decrease in Gini index for a given variable at each split. Thus, a higher mean decrease in Gini index corresponds to that variable being of greater value to the model prediction.

Another excellent attribute of random forest is out-of-bag (OOB) error which was described by Breiman (2001). Consider a particular observation $z_i = (x_i, y_i)$. When re-sampling the data set in step one of the random forest algorithm, it is the case that not every sample will have included z_i. An out-of-bag classifier is created for z_i by taking the majority vote of all trees that do not contain z_i. The out-of-bag error is the error rate derived from this classifier as it is applied to the entire training set. OOB error can replace cross-validation which would normally be standard when evaluating data sets. Hastie et al. (2009) assert that using out-of-bag error is equivalent to cross-validation with sufficient B.

Random forest is suitable for the current data set because it is a highly robust and resilient model. The current data is high dimensional, which random forest handles well. A logistic regression with LASSO variable selection was initially implemented with less success compared to the random forest, suggesting that the model could have nonlinear relationships to the response.

Random forest is implemented both for feature selection and model training. To perform feature selection, two models are trained. The first is the full model which includes all features calculated, and the second is a reduced model that only uses the statistics chosen from the top three statistics from the variable ranking in the full model. For example, if the top three features happened to be statistics 4, 5 and 6, then the reduced model would include the calculations associated for features 4, 5, and 6 from all local areas. This is done because an important part of the evaluation is to judge whether or not the model not only predicts the correct points in time in which the fighting occurs but also selects spatially relevant features. If the top three features from the full model only include one or two unique statistical features, then only those features are selected. For example, if the top three features were calculated from statistics 3, 3, and 4, then only 3 and 4 would be included in the reduced model.

Chapter 5

Model Evaluation

There are a number of results discussed in this chapter. First, random forest models are trained on both grid schemes. Using the same feature extraction and model evaluation procedure allows us to draw a direct comparison between the two options. To assess the impact of the ground speed transformation, the random forest model is also evaluated on the optical flow data set and compared to ground speed results; for the sake of mitigating redundancy only the first grid option (two unique sizes) is used to do this. Finally, the most important statistic from the random forest variable ranking is selected and tracked through time in local areas where fighting took place. Three different classification cutoff strategies are also tested and subsequently evaluated. Models are evaluated on their ability to accurately classify fighting and non-fighting correctly in time as well as their ability to prioritize relevant local areas during the feature selection process.

Due to the difficult task of judging the exact frame where a fight begins and ends, some data is discarded to remove any ambiguity during the random forest model training and evaluation, leaving a total of 1178 frames, 777 non-fighting and 401 fighting. For random forest modeling the data set is separated into training and testing sets, a standard practice for model evaluation. A random forest is trained on the training data and then evaluated by comparing the predicted and actual responses under the test data. Due to variables being in a time series format, the training and testing sets are not separated randomly but instead cut off at a certain point in time. That is Care was taken to make sure that similar proportions of fighting were in each set so that the classification predictions remain balanced.

Each grid scheme is divided into quadrants to help interpret the context of the local

areas. The top left quadrant (Q1) is where the fighting is located which means that all other quadrants (i.e., Q2, Q3, and Q4) are not relevant to the objective. Preliminary results were less than satisfactory when local areas from all four quadrants were included in the models. That is, predictions using all of the relevant local areas do not show favourable results under any condition (i.e., ground speed using grid schemes #1 and 2, and grid scheme #1 with optical flow data). Models had a tendency to be over-sensitive with a trend towards false positive predictions. Furthermore, top features selected appear as though they are biased towards the bottom right corner of the frame (Q4) and in general do not do a good job at selecting spatially relevant local areas. A detailed presentation of these results can be found in Appendix B.

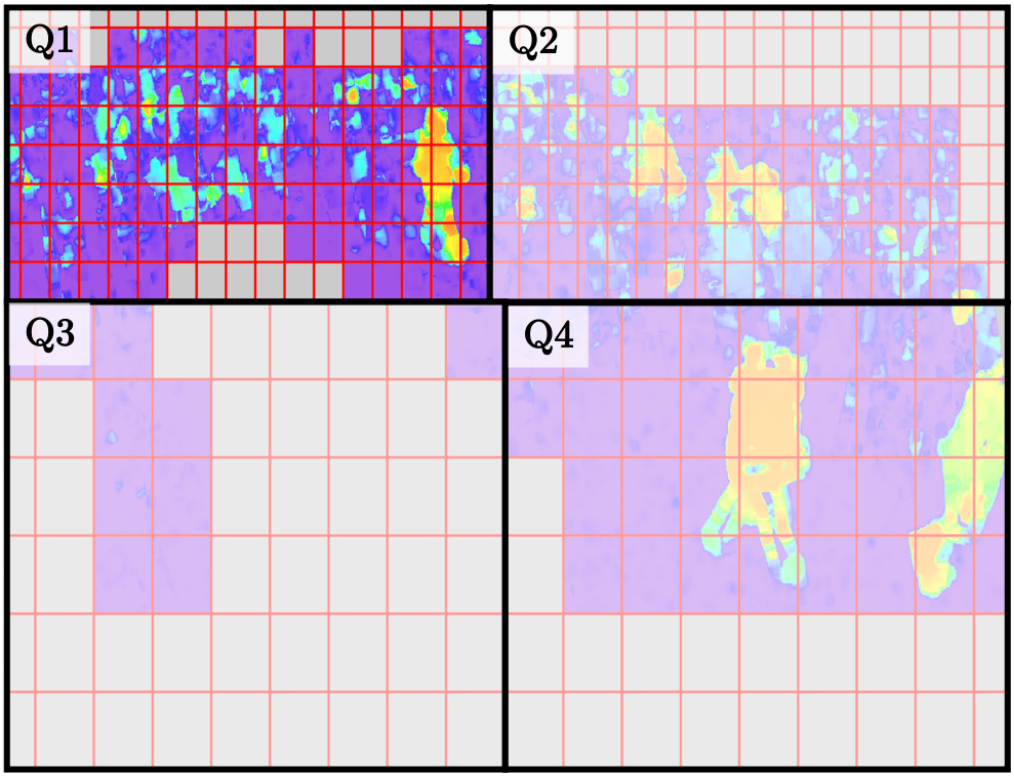

Figure 5.1: Guide to interpret local areas for the first grid scheme. The frame is separated into four distinct quadrants.

The initial findings suggest that the ground speed transformation was not successful in its objective to standardize speed values for the entire scene. Since the features selected appear to be a main issue, a narrowed-down selection of local areas is used to re-calculate the random forest model. With knowledge of where exactly fighting takes place, a subset of cells from the previous model are chosen. The specific region is the top left quadrant (Q1) of the frame which contains a total of 82 unique local areas and is depicted in figure 5.1. Using this subset of local areas, the random

forest prediction accuracy is assessed for both grid schemes on the ground speed. The same is done for optical flow data, except that only the first grid scheme is evaluated to spare redundancy. First the random forest feature selection is assessed, then the models' predictive ability. Comparing ground speed and optical flow under the first grid scheme allows for some insight into the success of the ground speed transformation.

An overview of the formulas used for classification evaluation is given to provide context to the results reported.

5.1 Measuring Effectiveness of a Classifier

Out-of-bag (OOB) error is recorded to measure the generalizability of the model, as was discussed in the previous chapter. In addition to OOB error, final conclusions for each model are drawn from multiple calculations to capture the full scope of the quality of the classifiers. In classification problems, there are four possible outcomes when comparing predicted and actual observations. They are described using the table below

		Actual Response	
		Positive	Negative
Predicted Response	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Table 5.1: Confusion matrix for classification, defines all possible prediction outcomes

True positive (TP) and true negative (TN) outcomes represent correct predictions. A false positive (FP) or false negative (FN) prediction corresponds to a misclassification of the observation. Overall accuracy is calculated using the following formula

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

Accuracy measures the overall proportion of correctly classified predictions among all responses. Although accuracy is a good measure for the general efficacy of the classifier, it fails to determine how exactly a model is under-performing. Sensitivity and specificity provide further insight into the classifier. Both are necessary because they each describe the positive and negative predictions respectively. Sensitivity, otherwise known as the true positive rate (TPR) represents the fraction of the total number of relevant instances that were actually retrieved. In other words, it measures

the proportion of positive (fighting) cases which were correctly classified.

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{5.2}$$

If sensitivity is very low then it means that there are many false negatives predicted and the model is under-performing in its ability to detect the target. If sensitivity is high then that means that out of all the cases that were supposed to be classified as positive, there is a high proportion which are being classified correctly. In contrast, specificity (also known as the true negative rate, TNR) measures the rate of truly negative cases among those which are classified as negative. That is,

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{5.3}$$

A lower specificity corresponds to a greater amount of false positives, meaning that the model is either over-classifying positive values (overly-sensitive) and/or tends to misclassify negative values as positive. A high true negative rate would represent accuracy among true negative responses. Precision is also calculated to add further insight. Precision is the proportion of positive predictions which are truly positive.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5.4}$$

It is useful in determining the quality of the positive predictions. It is different from sensitivity in that sensitivity is meant to be used in evaluating whether or not truly positive values were overlooked by the classifier whereas precision is about the reliability of the positively predicted values. In fact, sensitivity and precision are often contrasted against each other and can be used together to create a score (F_1 score).

Evaluating a classification model using all of these measures provides good insight into how a model is performing. However, it is misleading to interpret them solely by looking at their proportions, baseline accuracy rates must be taken into account. For example, if all cases were classified as positive then the sensitivity would be 1, regardless of the actual proportion of true positives in the response. The same can be said for specificity. To further expand on this thought we consider three theoretical scenarios for the current training and testing data; predictions are assigned randomly, all predictions are negative, all predictions are positive.

Scenario	Accuracy	TPR	TNR	Precision
Randomly Predicted	50%	50%	50%	33%
All Negative Prediction	67%	0%	100%	-
All Positive Predictions	33%	100%	0%	33%

Table 5.2: Percentage results for three theoretical scenarios of prediction. Note that approximately 33% of the test data is fighting and approximately 67% of it is non-fighting. Precision cannot be calculated when all values are predicted as non-fighting as there are no TP or FP.

These three baseline scenarios should be taken into account when interpreting the results. They also highlight the importance of looking beyond merely the accuracy. For example, an accuracy rate of 67% does not necessarily imply that all responses are negative. To draw that conclusion would require looking at the true positive and true negative rates as well. With these points in mind, we begin by presenting the feature selection results.

5.2 Random Forest Feature Selection

For the current research the random forest model ranks variables' importance via mean decrease in Gini impurity. These variables are statistics calculated over time for a given local area. Therefore, each variable has a given location and statistic. It is expected that in a successful variable ranking, the top features come from local areas where fighting occurs. For the current data there are 14 such areas. The following section shows the top ten features ranked under the full random forest model, to assess both the spatial relevance (and by extension effectiveness) as well as exploring which statistics are seen to be the most important. Then the subset of unique statistics from the top three variables ranked in the full model are carried on to be used in the reduced model. Only top ranking features for the full models are reported in this chapter because the reduced model variable ranking results are similar (their results can be found in Appendix B).

Ground Speed, Grid Scheme #1

Rank	Local Area #	Feature	Mean Decr. Gini	Contains fighting?
1	80	95^{th} percentile	12.12	yes
2	114	Mean of values > 1	11.83	yes
3	81	95^{th} percentile	10.75	yes
4	144	Ratio of the mean	10.41	no
5	179	Ratio of the mean	10.17	no
6	183	95^{th} percentile	9.71	no
7	179	95^{th} percentile	9.27	no
8	114	95^{th} percentile	8.04	yes
9	82	95^{th} percentile	7.99	yes
10	144	95^{th} percentile	6.92	no

Table 5.3: In grid scheme #1, top features selected by the full model after only using local areas contained in Q1.

The 95^{th} percentile and mean of all values above 1 are the features to be used in the reduced model. Five of the top ten features are from local areas containing fighting. The 95^{th} percentile is the most common in the table, making up seven of the top ten features. The same results are now presented, except under grid scheme #2. Note that because local area indexing is different in this scheme the fighting local areas have different IDs (consult Appendix B for a guide to local area indexing in either grid scheme).

Ground Speed, Grid Scheme #2

Rank	Local Area #	Feature	Mean Decr. Gini	Contains fighting?
1	146	Ratio of the mean	15.45	no
2	86	Mean of values > 1	11.70	yes
3	175	Ratio of the mean	10.76	no
4	87	Ratio of the mean	9.80	yes
5	114	Ratio of the mean > 0.1	9.51	no
6	114	Ratio of the mean	9.43	no
7	146	95^{th} percentile	8.76	no
8	88	95^{th} percentile	8.66	yes
9	53	Ratio of the mean	8.49	yes
10	119	Mean of values > 1	7.50	yes

Table 5.4: Using grid scheme # 2, top features selected by a full model that only considers local areas from Q1.

The mean values above 1 is selected for the reduced model as in the first grid scheme, along with the ratio of the mean. Only four of the eight possible statistics are ranked in the top ten in both grid schemes. The results above also show that random forest variable ranking did manage to rank relevant locations in the frame as among the top features. We now compare the results from the ground speed transformation to the optical flow data. If transforming data to ground speed values is not necessary then it would be expected that a variable ranking from the random forest has fighting local areas among the top features selected, as seen in the ground speed.

Optical Flow, Grid Scheme #1

Rank	Local Area #	Feature	Mean Decr. Gini	Contains fighting?
1	80	95^{th} percentile	17.09	yes
2	179	Ratio of the mean	14.12	no
3	183	95^{th} percentile	9.63	no
4	82	95^{th} percentile	8.80	yes
5	179	Ratio of the mean truncated at 0.1	8.13	no
6	80	Ratio of the mean	7.32	yes
7	144	Ratio of the mean	7.23	no
8	214	Ratio of the mean	6.67	no
9	214	Ratio of the variance	6.52	no
10	46	Ratio of the variance	6.28	no

Table 5.5: Top ranking features from the full random forest model using optical flow data with local areas from Q1.

Using optical flow data, the most important variable selected is indeed relevant to the fighting. However, compared to the ground speed results there are only three fighting features in the top ten. In this case, the reduced model contains the 95^{th} percentile along with the ratio of the mean between a local area and its surrounding local region. Results appear to be worse for the optical flow however it can be confirmed through comparing model accuracy against the ground speed.

5.3 Random Forest Model Prediction

The following section shows and debriefs the random forest classifier for full and reduced models under the three conditions seen above. This is done so for all random forest models and using only features within Q1. Again, the full model considers all features from Table 4.2 while the reduced model only includes statistics that are ranked in the top three from its full model.

Data	Grid	Model	OOB error	Accuracy	TPR	TNR	Precision
Ground Speed	1	Full	0.0069419	74.34%	86%	68.63%	57.33%
		Reduced	0.0048430	78.29%	94%	70.59%	61.04%
	2	Full	0.00627943	69.08%	80%	63.73%	51.95%
		Reduced	0.0078596	70.39%	88%	61.76%	53.01%
Optical Flow	1	Full	0.0045807	48.03%	42%	50.98%	29.58%
		Reduced	0.0026910	51.97%	44%	55.88%	32.84%

Table 5.6: Random forest prediction results on the test data for all conditions.

To begin with assessing the ground speed transformation, we compare the first two and last two rows. The ground speed data clearly outperforms optical flow for all classification measures, both in the full and reduced models. In fact, optical flow results are close to the random chance condition in Table 5.2. Comparison between grid schemes depends on whether or not you are looking at the full or reduced model. In the first grid scheme, all classification measures improve, implying that the full feature set contains erroneous and/or noisy features. In contrast, the quality of the classifier in grid scheme #2 is actually worse in the reduced model compared to the full. This could suggest that feature selection is not as effective in this arrangement compared to the first one and that some important features from the full model failed to be captured in the reduced model.

Overall it can be said that the random forest models achieve better results using the ground speed data conditions. Top features using ground speed data are more likely to include local areas where the fighting occurred and although the model prediction has a tendency for false positives, higher percentages are observed for accuracy, specificity, and precision calculations. In contrast the optical flow data performs very poorly, prediction results are similar to random chance. The difference in predictive quality between ground speed and optical flow data sets indicate that although it is ultimately not possible to successfully generalize results to the entire frame, the ground speed

transformation did successfully generalize motion over a larger region of the image than could be achieved with optical flow and therefore does find success in combating a moderate level of regional discrepancies.

The dissimilarity between the OOB error results and the accuracy in Table 5.6 can be attributed to the time series dependency inside the training and testing data.

5.4 Designing a Simple Classifier

As previously mentioned the features extracted are time series data, yet this aspect of the data has not been explored in depth thus far. Examining the time series of features where fighting specifically occurs could add a great deal of insight into the quality of these features and their ability to identify fighting frames. When only considering local areas where fighting is known to occur, is it possible to construct a simple yet effective classifier from the most valuable statistical features? A random forest variable importance ranking is leveraged as a means to determine the most important statistical features. Local areas are harvested from the first grid scheme and so they all have the same dimension. There are a total of 14 local areas that contain fighting in grid scheme # 1. Their IDs are 79-83, 113-118, and 150-152. The data is not divided into training and testing sets since no model is being trained; weights are also not used as that in the random forest so as not to add any prior assumptions.

Ground speed for fighting areas only, Grid Scheme #1

Rank	Local Area #	Feature	Mean Decr. Gini
1	81	95^{th} percentile	34.85
2	82	95^{th} percentile	27.27
3	114	Mean of values > 1	25.76
4	80	95^{th} percentile	23.97
5	114	95^{th} percentile	21.57
6	81	Ratio of the mean	13.71
7	81	Ratio of a mean truncated at 0.1	13.42
8	114	Ratio of the 95^{th} percentile	12.68
9	82	Mean of values > 1	12.09
10	79	Mean of values > 1	11.96

Table 5.7: Top ten variables as ranked by the random forest model. Recall that only the 14 local areas in which fighting occur are considered.

The random forest variable ranking shows that the most important variable is the 95^{th} percentile. It was also the most common statistic selected among the top ten features. Thus, a decision was made to only use the 95^{th} percentile in designing the time series classifier. Moreover the second highest ranked feature in Table 5.7, the mean of values above 1 in local area 114, has a correlation of 0.89 with the 95^{th} percentile of the same local area so there is little incentive to examine that statistic separately in this context.

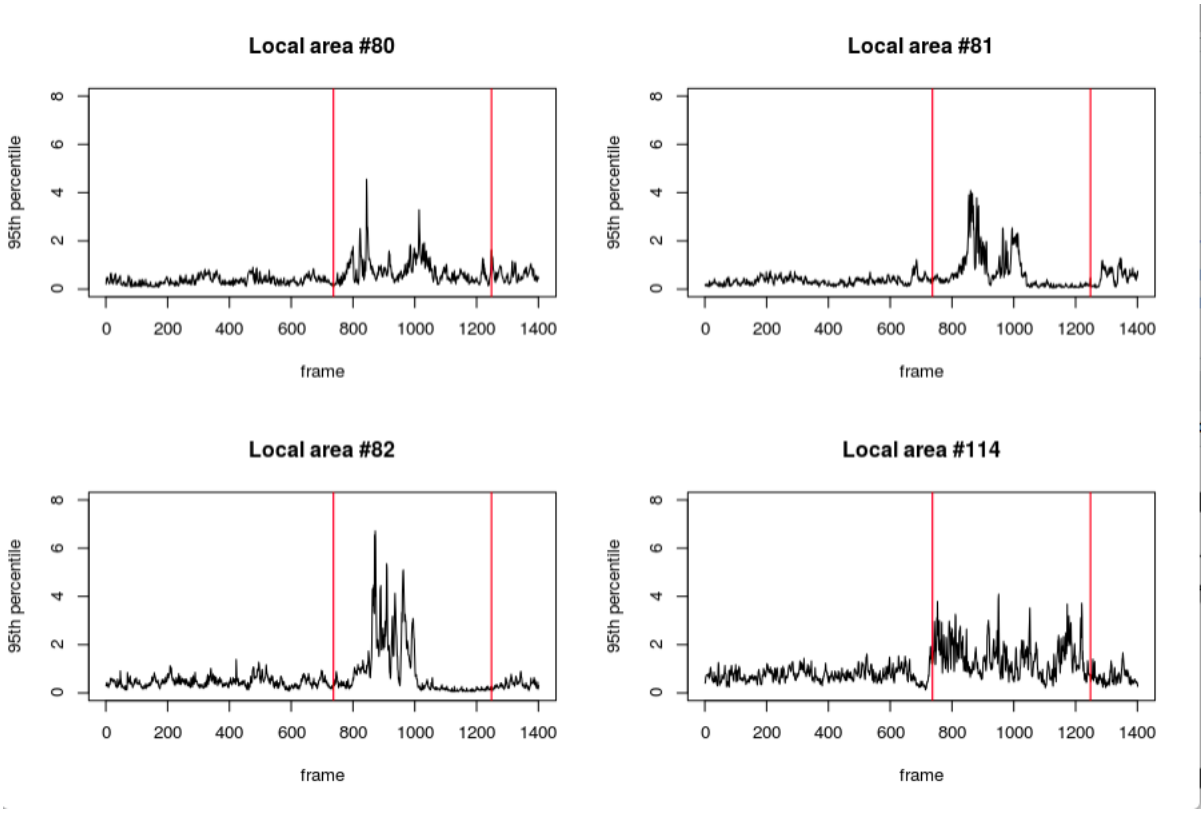

Figure 5.2: Time series plots of the 95^{th} percentile for the top four important local areas as selected from the random forest. Within the two red lines are the frames where the response is classified as fighting. It is clear that at least some of the fighting is being picked up in the calculation.

The next step is to determine a classification rule that is calibrated to either all of the fighting local areas, or each area separately. Three rules are tested. Since these classifiers do not require model training, all 1401 frames were used in the model evaluation calculations. The first classifier is a single cutoff, applied to all local areas. The cutoff is optimized using Youden's J statistic ($J = \text{sensitivity} + \text{specificity} - 1$) which places equal weighting on sensitivity and specificity. The other two classifiers calculate a cutoff for each local area separately, using the first 500 time points (a

period of time when no fighting is taking place). The second classifier calculates a cutoff by taking the mean of each local area plus four times its standard deviation ($\bar{x} + 4 \cdot s$). The final classifier takes the maximum value among the first 500 points as a cutoff for each local area.

Classifier 1	One optimized cutoff that is applied to all local areas.
Classifier 2	For the first 500 points, $\bar{x} + 4 \cdot s$. Calibrated for each local area separately.
Classifier 3	For the first 500 points, the max value of each local area.

Table 5.8: Summary of each classification rule. The standard deviation was found to be relatively stable for the first 500 time points in the local areas specific to the fighting.

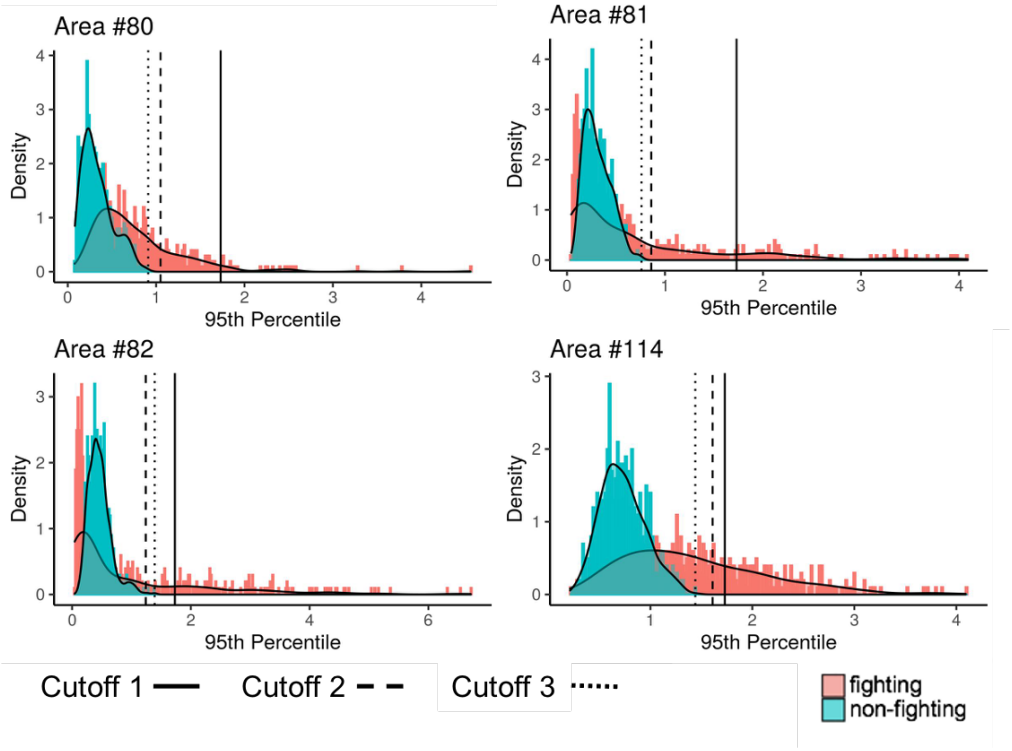

Figure 5.3: Distribution of the first 500 time points, contrasted with 500 time points during the fighting period, for the top four variables according to the random forest feature selection. The relationship between the cutoff values and the distributions are also shown.

For all classifiers, the entire region is classified as fighting at a particular time point if *any* of the features in the local areas are above their determined threshold. Each of these three classifiers are evaluated under two conditions. The first uses features extracted using ground speed data as has been done thus far, and the second data set

is a smoothed version of ground speed features using a 15-point moving average. The ROC curve in Figure 5.4 shows that for classifiers where only one value is used (such as in Classifier 1) a smoother time series yields greater accuracy. In the interest of maintaining a relatively fast classifier, a 15-point moving average is selected because it is quite smooth while still having less than a second of delay (future work can consult security experts to figure out the maximum amount of delay that is reasonable).

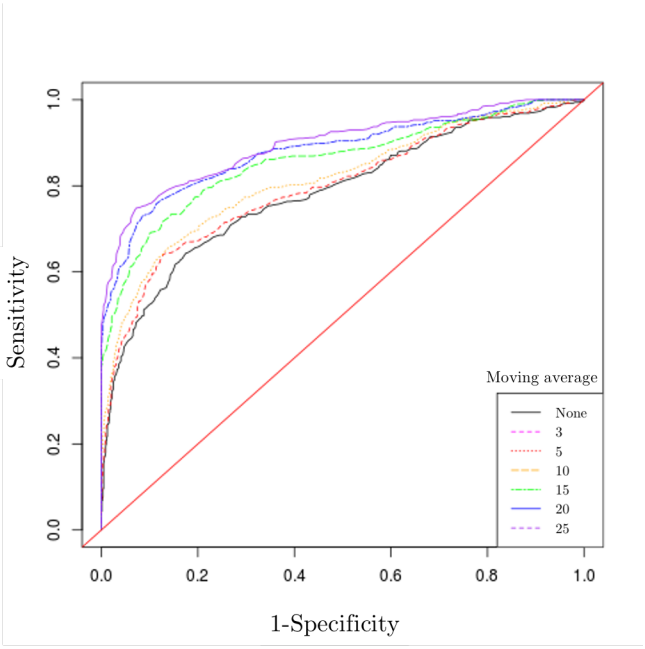

Figure 5.4: ROC curve displaying different moving average lags. The closer to the top left corner, the better the cutoff. It appears as though moving averages with longer lag tend to be more successful. The optimal point on the curve is determined using Youden's J statistic.

Classifier	Moving Avg.	Accuracy	TPR	TNR	Precision
Classifier 1	None	75.87%	64.26%	82.56%	67.98%
	15	84.73%	73.05%	91.45%	83.11%
Classifier 2	None	83.87%	70.31%	91.68%	82.95%
	15	89.29%	84.96%	91.79%	85.63%
Classifier 3	None	84.08%	74.80%	89.43%	80.29%
	15	81.37%	89.84%	76.49%	68.76%

Table 5.9: Results for all three classifiers under both data conditions. Cutoff values can be found in Appendix B.

All classifiers perform well, especially compared to the random forest classification. If no moving average is applied, then Classifier 3 performs the best. However, the

15-point moving average condition with Classifier 2 works best overall, reaching an accuracy of 89.29%.

Due to the simplicity and efficiency of cutoff classifiers like the ones above, they are most certainly preferred over a more complex random forest classification. However, keep in mind that the results found in Table 5.9 are only applicable for the 14 local areas containing fighting and do not necessarily generalize to the first quadrant let alone the entire frame. In order to apply one cutoff to several local areas such as in Classifier 1, it is crucial that all local areas across that region have similar values to describe the same types of motion. Classifiers 2 and 3 are readily applicable to the whole frame because their cutoffs are calibrated separately for each local area. Figure 5.5 shows how Classifiers 2 and 3 would perform in four local areas in the bottom right of the frame. Similar to results in Table 5.9, results improve after smoothing with a 15-point moving average. Results here are promising in terms of generalizing to other parts of the frame, especially with more training data calibrating each cutoff.

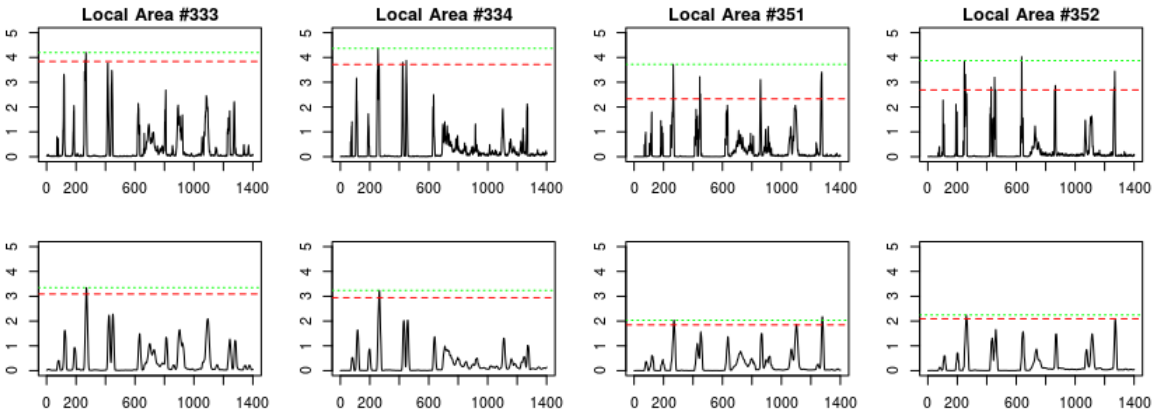

Figure 5.5: Each column shows a different local area that is located in the bottom right corner of the frame. The top row is the 95^{th} percentile and the bottom row is the smoothed 95^{th} percentile through the 15-point moving average. In red is Classifier 2 and Classifier 3 is in green.

Chapter 6

Discussion

When using local areas from the top left section of the frame, the ground speed data is shown to have more relevant features selected compared to optical flow in both grid schemes. Furthermore, the random forest classification model also performs better under both ground speed conditions compared to optical flow. These results suggest that the ground speed transformation was at least partially successful in standardizing the motion in the frame.

There are similar results between the two grid schemes indicating that the conditions are reasonably robust to the location of the grid placements used. In the interest of generalizing results over the entire frame, future iterations could consider employing a step-sized style of local area grouping, or spatially overlapping windows when calculating statistical features. In general grid layouts have the potential to be optimized to achieve greater accuracy.

Three cutoff classifiers were applied to the local areas containing fighting. All three methods are shown to be successful on smoothed and non-smoothed versions of the features. The results from Classifier 1 indicate that the ground speed transformation did well as only one value applied to all areas was needed to garner a decent accuracy, implying that the areas are similar in range to each other. The most successful method uses the smoothed data to calculate the mean plus four times the standard deviation for each feature, however Classifiers 2 and 3 are very comparable. Examples in the future using other data could identify which among the three is the most generalizable. For example, it is possible that Classifier 3, which uses the max value, could prove to be more susceptible to outliers in other contexts.

Many statistical features were calculated however most were found to be correlated with at least one other statistic in the feature set. There was an attempt to remove highly correlated features at a general level yet collinearity is still present in the feature set used in the random forest model. Because of this, the interpretation of which statistics are the more relevant than others is limited. The 95^{th} percentile is consistently chosen as an important feature across multiple different conditions. Furthermore, tracking the 95^{th} percentile through time over local areas where fighting occurred yields a good classifier. Thus it would appear as though the 95^{th} percentile is adequate in performing the objectives of the current research, whether or not it is definitively the most important.

There is further opportunity to analyze which statistic is best among all those calculated. It is also possible for more useful features to be extracted. Because ground speed was used over velocity in an effort to incorporate both horizontal and vertical directions into one value, information about the direction of the movement is lost. It is possible that this type of information could create prominent features, as was seen in Riberio & Victor (2005) and Huang & Chen (2014). Ground speed was initially extracted instead of individual velocities in the interest of reducing the dimensionality of the feature set by using only one set of data instead of x- and y-directions separately. If directionality is shown to be highly useful then it may be of interest to calculate features separately for each direction of the velocity.

There are a number of factors that could have contributed to the transformation from optical flow to ground speed being unsuccessful in moderating all distance-related issues in the frame. First, distance from the camera affects the pixel resolution of a given object which in turn affects the quality of the signal. That is, an object at a farther distance from the camera is represented with fewer pixels than if that same object was closer to the camera. This is a problem for the current data set since the fighting event is indeed far from the camera and so it is more difficult to detect compared to that same event taking place closer to the camera. Time series plots do show that there is at least a detectable signal present. Nonetheless, the disparity in resolution could have limited the quality of that signal.

In addition, estimating a 3D reality with a 2D image plane creates some barriers to perfect estimation. Points on the screen are assumed to be points on the ground which is not always true and creates some error in the calculation. For example, consider a person walking. Their head is a certain height above their feet; in reality the person's entire body is more or less moving at the same speed (excluding the speed of their

legs and feet). If every point on the image is taken to be a point projected onto the ground, a person's head would be measured as a further distance away from the camera compared to their feet, resulting in the head having a faster speed than the lower half of the body. This creates some error, especially in closer objects as their images are larger.

The data set itself also only contains one fighting event. Finding results that were able to be generalized to the entire frame would have been easier to achieve if multiple anomalous events took place at various spots in the scene and were available for comparison. In the current scene it is questionable whether such other events took place at all let alone were recorded and made available for analysis.

It is also important to note that the model built in this project was not assessed for generalizability over other pieces of footage and that the current training data was very limited. The concepts and techniques used could be applied to similar videos. In fact the model might perform even better, considering that the circumstances set by this particular video proved to be challenging ones. There is an opportunity to assess and improve the generalizability of the results to other videos through applying current methods to benchmark data sets such as those created by Hassner et al. (2012). This could also act not only as an assessment of generalizability but to also provide more training data for the model. Similarly, the random forest method used for classification could be compared to other classification models, especially support vector machines (SVM) as this is a common method used in the literature.

In addition to building upon methods used in the current research, there is an opportunity to explore other avenues of statistical analysis. For example, time series analysis could be beneficial. In viewing the time series plots, time series change point detection could be implemented to identify the exact point at which anomalous behaviour begins. A suitable time series method for the current data could be hidden Markov models (HMM), which in essence use observations related to a process or system to draw inference on the state of the system itself. It is a well-established method applied in human action recognition, where "...hidden states that correspond to different phases in the performance of an action, [and] model state transition and observation probabilities." (Poppe, 2010). A pointed application in the literature comes from Andrade et al. (2006), who applied HMMs to optical flow extracted from crowd video data to classify "emergency events" such as a person falling over.

Unsupervised models are also an option to explore with the introduction of more

data sets. Unsupervised (also known as generative) models are models in which information about the response is not factored into its training. This is particularly useful for online detection as in a real-life application there will be no prior knowledge as to when a fight could occur. Not using a response vector will circumvent the issues experienced when trying to define exact frames in which the fighting occurred manually.

To summarize, after applying both well-established and novel methods to gray-scale video imaging data, statistical features were extracted from various relevant local areas within the scene. With these features, a random forest model was trained to predict the time at which "fighting" occurs on a test data set. Although satisfactory results were not achieved using the entire frame, narrowing down the region of interest improved model accuracy as well as demonstrated that the novel ground speed transformation did help improve the feature set. Furthermore, simple cutoff classifiers were shown to be successful when restricted to the fighting parts of the frame. There is opportunity for further application to other similar video data sets to work towards a generalized version of the model. Future research could be done to improve the current methods as well as explore new topics such as unsupervised modeling.

What is clear is that we have only scratched the surface of what there is to explore with not only just the methods applied to the current data set but also the multitude of other approaches offered for accurate detection of violence in crowd surveillance video footage.

References

Ali, J., Khan, R., Ahmad, N., & Maqsood, I. (2012). Random forests and decision trees. International Journal of Computer Science Issues (IJCSI), 9(5), 272.

Andrade, E. L., Blunsden, S., & Fisher, R. B. (2006, August). Hidden markov models for optical flow analysis in crowds. In 18th international conference on pattern recognition (ICPR'06) (Vol. 1, pp. 460-463). IEEE.

Beauchemin, S. S., & Barron, J. L. (1995). The computation of optical flow. ACM computing surveys (CSUR), 27(3), 433-466.

Bouguet, J. Y. (2001). Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. Intel corporation, 5(1-10), 4.

Boulesteix, A. L., Janitza, S., Kruppa, J., & König, I. R. (2012). Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2(6), 493-507.

Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

Buch, N., Velastin, S. A., & Orwell, J. (2011). A review of computer vision techniques for the analysis of urban traffic. IEEE Transactions on Intelligent Transportation Systems, 12(3), 920-939.

Farnebäck, G. (2002). Polynomial expansion for orientation and motion estimation (Doctoral dissertation, Linköping University Electronic Press).

Farnebäck, G. (2003, June). Two-frame motion estimation based on polynomial expansion. In Scandinavian conference on Image analysis (pp. 363-370). Springer, Berlin, Heidelberg.

Feng, W., Liu, R., & Zhu, M. (2014). Fall detection for elderly person care in a

vision-based home surveillance environment using a monocular camera. signal, image and video processing, 8(6), 1129-1138.

Gao, Y., Liu, H., Sun, X., Wang, C., & Liu, Y. (2016). Violence detection using oriented violent flows. Image and vision computing, 48, 37-41.

Hashemi, J., Spina, T. V., Tepper, M., Esler, A., Morellas, V., Papanikolopoulos, N., & Sapiro, G. (2012, November). A computer vision approach for the assessment of autism-related behavioral markers. In 2012 IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL) (pp. 1-7). IEEE.

Hassner, T., Itcher, Y., & Kliper-Gross, O. (2012, June). Violent flows: Real-time detection of violent crowd behavior. In 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (pp. 1-6). IEEE.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media.

Horn, B. K., & Schunck, B. G. (1981, November). Determining optical flow. In Techniques and Applications of Image Understanding (Vol. 281, pp. 319-331). International Society for Optics and Photonics.

Huang, J. F., & Chen, S. L. (2014, August). Detection of violent crowd behavior based on statistical characteristics of the optical flow. In 2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD) (pp. 565-569). IEEE.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.

Klette, R. (2014). Concise computer vision. Springer, London.

Liu, J., Shahroudy, A., Perez, M. L., Wang, G., Duan, L. Y., & Chichung, A. K. (2019). Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. IEEE transactions on pattern analysis and machine intelligence.

Lucas, B. D., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision.

Mahadevan, V., Li, W., Bhalodia, V., & Vasconcelos, N. (2010, June). Anomaly detection in crowded scenes. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (pp. 1975-1981). IEEE.

Poppe, R. (2010). A survey on vision-based human action recognition. Image and

vision computing, 28(6), 976-990.

Primartha, R., & Tama, B. A. (2017, November). Anomaly detection using random forest: A performance revisited. In 2017 International conference on data and software engineering (ICoDSE) (pp. 1-6). IEEE.

Ribeiro, P. C., Santos-Victor, J., & Lisboa, P. (2005, September). Human activity recognition from video: modeling, feature selection and classification architecture. In Proceedings of International Workshop on Human Activity Recognition and Modelling (pp. 61-78).

Thomas, G., Gade, R., Moeslund, T. B., Carr, P., & Hilton, A. (2017). Computer vision for sports: Current applications and research topics. Computer Vision and Image Understanding, 159, 3-18.

Appendix A

Derivation of the Ground Speed Transformation

The following is a more expanded version of the ground speed transformation.

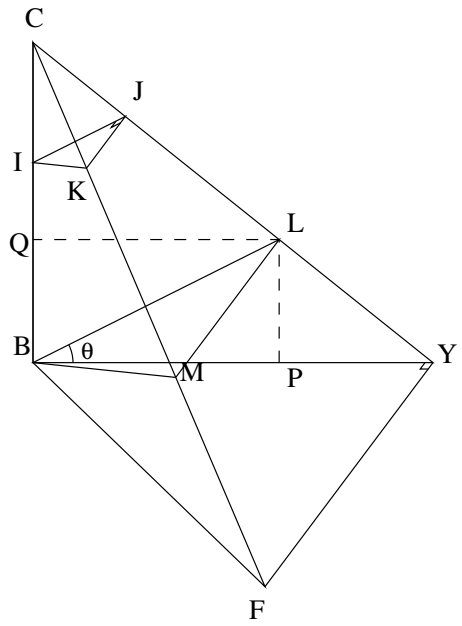

Figure A.1: The camera is fixed at point C. Triangle IJK represents the pixel image plane of the scene, or in other words the video data that is a representation of the scene. The triangle BYF represents the ground plane, or the reality of the scene.

Triangle BLM is parallel to IJK and, through θ, can be used to find the values for points in triangle BYF. The point K is the point in the image plane corresponding to point F on the ground and is where the seating sign is located. $|CB|$ is the height

of the camera from the ground and is known. We assume the x-coordinate of the origin in the picture plane is in the centre of the video frame, i.e. $x_0 = 0$. Point I represents the origin in the image plane and is alternatively written with coordinates (x_0, y_0). Similarly J has coordinates (x_0, y) , K has coordinates (x, y) and therefore $y - y_0 = |IJ|$ and $x - x_0 = |JK|$. Point F (the base of the reference object) is given the coordinates $(\tilde{x}, \tilde{y})$. $\tilde{x}$ and $\tilde{y}$ can also be written as $|YF|$ and $|BY|$ respectively.

The following aims to calculate the ground speed for the point $(\tilde{x}, \tilde{y})$ in such a way that it only depends on the coordinates of the picture plane, (x, y) and the x- and y-components of optical flow. Once this is achieved, the same function can be applied to all coordinates on the picture plane.

The velocity (and then speed) of the ground plane can be found through a Jacobian transformation of the image plane,

$$\begin{pmatrix} \frac{\mathrm{d}\tilde{x}}{\mathrm{d}t} \\ \frac{\mathrm{d}\tilde{y}}{\mathrm{d}t} \end{pmatrix} = \begin{pmatrix} \frac{\mathrm{d}\tilde{x}}{\mathrm{d}x} & \frac{\mathrm{d}\tilde{x}}{\mathrm{d}y} \\ \frac{\mathrm{d}\tilde{y}}{\mathrm{d}x} & \frac{\mathrm{d}\tilde{y}}{\mathrm{d}y} \end{pmatrix} \begin{pmatrix} \frac{\mathrm{d}x}{\mathrm{d}t} \\ \frac{\mathrm{d}y}{\mathrm{d}t} \end{pmatrix} \tag{A.1}$$

where optical flow is written as $\left(\frac{dx}{dt}, \frac{dy}{dt}\right)$. In constructing the Jacobian matrix, expressions for $\tilde{x}$ and $\tilde{y}$ must be defined only in terms of points on the picture plane. To achieve this, attention is paid to the proportional relationship between the planes (defined through the triangles mentioned above) while solving for any unknowns that crop up along the way.

BL is parallel to IJ and LM is parallel to JK so the coordinates are proportional to BL and LM. Proportionality between triangle IJK and BLM is as follows.

$$\frac{|CI|}{|CB|} = \frac{|IJ|}{|BL|} = \frac{|JK|}{|LM|} \tag{A.2}$$

Now, let P and Q be the orthogonal projections of L onto BY and BC respectively (Figure (3.2) depicts this with the dotted lines). Then, triangle CBY can be broken into triangles CBL and LBY. Comparing total area gives

$$\frac{1}{2}|CB||BY| = \frac{1}{2}|CB||QL| + \frac{1}{2}|BY||LP| \tag{A.3}$$

Recall $y - y_0 = |IJ|$.

Through equation (A.2), $|BL|$ can be expressed as

$$|BL| = \frac{|CB|}{|CI|}(y - y_0) \tag{A.4}$$

which can be re-arranged to find an expression for $|BY|$

$$\begin{aligned} |CB||BY| &= |CB||BP| + |BY||LP| \\ |BY| &= \frac{|CB||BP|}{|CB| - |LP|} \\ |BY| &= \frac{|BL||CB|\cos\theta}{|CB| - |BL|\sin\theta} \end{aligned}$$

Now, using the proportional relationships as outlined in equation (A.2) to re-express $|BY|$ as the y-component speed at the reference point ($\tilde{y}$) and in terms of lengths relative to the camera C. Proportionality gives

$$|BL| = \frac{|CB|}{|CI|}(y - y_0)$$

which can substituted into the expression for $|BY|$

$$\tilde{y} = |BY| = \frac{\frac{|CB|}{|CI|}(y - y_0) \cdot |CB|\cos\theta}{|CB| - \frac{|CB|}{|CI|}(y - y_0)\sin\theta}$$

then multiply the top and bottom by $\frac{|CI|}{|CB|\sin\theta}$ to assist in re-arranging

$$\begin{aligned} \tilde{y} &= \frac{(|CB|^2(y - y_0)\cos\theta) \cdot \frac{1}{|CI|}}{|CB| - \frac{(y-y_0)|CB|}{|CI|\sin\theta}} \left(\frac{|CI|/|CB|\sin\theta}{|CI|/|CB|\sin\theta}\right) \\ &= \frac{|CB|(y - y_0)\frac{cos\theta}{\sin\theta}}{\frac{|CI|}{\sin\theta} - (y - y_0)} \\ &= \frac{|CB|\cot\theta(y - y_0)}{|CI|\csc\theta - (y - y_0)} \end{aligned}$$

Let $a = |CB|\cot\theta$ and $b = |CI|\csc\theta$.

$$\tilde{y} = \frac{a(y - y_0)}{b - (y - y_0)} \tag{A.5}$$

θ and $|CI|$ are unknown and therefore must be solved for, requiring more equations. An expression for $\tilde{x}$ is derived. Recall that $|JK|$ is parallel to $|YF|$. So, the triangles

CLM and CYF are similar and give

$$\frac{|YF|}{|LM|} = \frac{|CY|}{|CL|} = \frac{|CB|}{|CQ|} \tag{A.6}$$

which means that $|CQ| = |CB| - |QB| = |CB| - |BL| \sin\theta$. Plugging this in, as well as multiplying the top and bottom by $|CI| \csc\theta$ to simplify the expression gives

$$\begin{aligned}
\tilde{x} = |YF| &= |LM| \cdot \frac{|CB|}{|CB| - |BL| \sin\theta} \left(\frac{|CI| \csc\theta}{|CI| \csc\theta} \right) \\
&= |LM| \cdot \frac{|CB||CI| \csc\theta}{|CB||CI| \csc\theta - |BL||CI| \sin\theta \csc\theta} \\
&= |LM| \cdot \frac{|CB||CI| \csc\theta}{|CB||CI| \csc\theta - \left[\frac{(y-y_0)|CB|}{|CI|} \right] |CI|} \\
&= |LM| \cdot \frac{|CI| \csc\theta}{|CI| \csc\theta - (y - y_0)}
\end{aligned}$$

We know from equation (A.2) that $|LM| = \frac{(x-x_0)|CB|}{|CI|}$ and that $|CB| \csc\theta = \frac{a}{\cos\theta}$. So we can substitute those values in and re-arrange to get

$$\tilde{x} = \frac{a(x - x_0)}{\cos\theta(b - (y - y_0))} \tag{A.7}$$

Recall that the coordinates of the sign on the ground are $(\tilde{x}, \tilde{y})$. Let the vertical height from the ground to the bottom of the sign be $\tilde{z}$ and the height of the camera $|CB| = h$, both of which are known. The line from the camera to the bottom of the sign (above the ground) extends to meet the ground at $(\tilde{x}', \tilde{y}')$. By similar triangles, $\frac{\tilde{x}' - \tilde{x}}{\tilde{x}'} = \frac{\tilde{z}}{h}$ and $\frac{\tilde{y}' - \tilde{y}}{\tilde{y}'} = \frac{\tilde{z}}{h}$. Isolating for $\tilde{x}'$ and $\tilde{y}'$ gives $\tilde{x}' = \frac{h}{h - \tilde{z}}\tilde{x}$ and $\tilde{y}' = \frac{h}{h - \tilde{z}}\tilde{y}$.

Recall that $a = h \cot\theta$, $b = |CI| \csc\theta$, and $x_0 = 0$. This leaves four unknowns; θ, $|CI|$, x_0, and y_0.

$$\frac{h}{h - \tilde{z}} \cdot \tilde{x} = \frac{a(x' - x_0)}{\cos\theta(b - (y' - y_0))} \tag{A.8}$$

$$\frac{h}{h - \tilde{z}} \cdot \tilde{y} = \frac{a(y' - y_0)}{b - (y' - y_0)} \tag{A.9}$$

Also,

$$\tilde{x}^2 + \tilde{y}^2 = BF^2 = d^2 \tag{A.10}$$

where $d = |BF|$ is horizontal distance from the camera to the seating sign and is known.

Recall that $a = |CB| \cot\theta = h \cot\theta$ and $b = |CI| \csc\theta$. This leaves four unknowns; θ, $|CI|$, x_0 and y_0. With the previously stated assumption that (x_0, y_0) is at the bottom centre of the frame, $x_0 = 0$ in the coordinate space of the image plane.

From (A.5) and (A.7), we get $\frac{\tilde{x}\cos\theta}{\tilde{y}} = \frac{x - x_0}{y - y_0}$; similarly from (A.8) and (A.9), we get $\frac{\tilde{x}\cos\theta}{\tilde{y}} = \frac{x' - x_0}{y' - y_0}$. Thus

$$xy' - yx' = x_0(y' - y) + y_0(x - x')$$

and from these expressions y_0 is found to be -8644.

Next θ can be solved for using the above equations. Substituting equations (A.5) and (A.7) into (A.10) to get

$$\Big(\frac{a(x - x_0)}{\cos\theta(b - (y - y_0))}\Big)^2 + \Big(\frac{a(y - y_0)}{b - (y - y_0)}\Big)^2 = d^2$$

Isolate θ to one side

$$\Big(\frac{a(x - x_0)}{\cos\theta(b - (y - y_0))}\Big)^2 = d^2 - \Big(\frac{a(y - y_0)}{b - (y - y_0)}\Big)^2$$

$$\frac{a(x - x_0)}{\cos\theta(b - (y - y_0))} = \sqrt{d^2 - \Big(\frac{a(y - y_0)}{b - (y - y_0)}\Big)^2}$$

$$\frac{1}{\cos\theta} = \sec\theta = \frac{b - (y - y_0)}{a(x - x_0)} \sqrt{d^2 - \Big(\frac{a(y - y_0)}{b - (y - y_0)}\Big)^2}$$

$$\sec\theta = \frac{b - (y - y_0)}{a(x - x_0)} \sqrt{d^2 - \Big(\frac{a(y - y_0)}{b - (y - y_0)}\Big)^2}$$

Bring the denominator into the radical.

$$\sec\theta = \frac{b - (y - y_0)}{(x - x_0)} \sqrt{\frac{d^2}{a^2} - \Big(\frac{(y - y_0)^2}{(b - (y - y_0))^2}\Big)}$$

$$\sec\theta = \frac{b-(y-y_0)}{(x-x_0)}\sqrt{\frac{d^2}{h^2\cot^2\theta} - \Big(\frac{(y-y_0)^2}{(b-(y-y_0))^2}\Big)}$$

Bring the $\sec\theta$ into the expression.

$$1 = \frac{b-(y-y_0)}{\sec\theta(x-x_0)}\sqrt{\frac{d^2}{h^2\cot^2\theta} - \Big(\frac{(y-y_0)^2}{(b-(y-y_0))^2}\Big)}$$
$$1 = \frac{b-(y-y_0)}{(x-x_0)}\sqrt{\frac{d^2}{h^2\cot^2\theta\sec^2\theta} - \Big(\frac{(y-y_0)^2}{\sec^2\theta(b-(y-y_0))^2}\Big)}$$

Recall $\frac{1}{\cos^2\theta} = \sec^2\theta$, $\frac{1}{\tan^2\theta} = \cot^2\theta$ and $\frac{1}{\cos^2\theta\tan^2\theta} = \frac{1}{\sin^2\theta} = \csc^2\theta$

$$1 = \frac{b-(y-y_0)}{(x-x_0)}\sqrt{\frac{d^2}{h^2\csc^2\theta} - \Big(\frac{(y-y_0)^2}{\sec^2\theta(b-(y-y_0))^2}\Big)}$$
$$1 = \frac{b-(y-y_0)}{(x-x_0)}\sqrt{\frac{d^2\sin^2\theta}{h^2} - \Big(\frac{(y-y_0)^2\cos^2\theta}{(b-(y-y_0))^2}\Big)}$$

Recall $1 = \sin^2\theta + \cos^2\theta$

$$1 = \frac{b-(y-y_0)}{(x-x_0)}\sqrt{\frac{d^2}{h^2}(1-\cos^2\theta) - \Big(\frac{(y-y_0)^2}{(b-(y-y_0))^2}\cos^2\theta\Big)}$$
$$1 = \frac{b-(y-y_0)}{(x-x_0)}\sqrt{\frac{d^2}{h^2} - \frac{d^2}{h^2}(\cos^2\theta) - \Big(\frac{(y-y_0)^2}{(b-(y-y_0))^2}\cos^2\theta\Big)}$$
$$1 = \frac{b-(y-y_0)}{(x-x_0)}\sqrt{\frac{d^2}{h^2} - \cos^2\theta\Big(\frac{d^2}{h^2} + \frac{(y-y_0)^2}{(b-(y-y_0))^2}\Big)}$$
$$\frac{d^2}{h^2} - \Big(\frac{(x-x_0)}{b-(y-y_0)}\Big)^2 = \cos^2\theta\Big(\frac{d^2}{h^2} + \frac{(y-y_0)^2}{(b-(y-y_0))^2}\Big)$$

Isolate θ.

$$\cos^2\theta = \frac{\frac{d^2}{h^2} - \Big(\frac{(x-x_0)}{b-(y-y_0)}\Big)^2}{\Big(\frac{d^2}{h^2} + \frac{(y-y_0)^2}{(b-(y-y_0))^2}\Big)} \tag{A.11}$$

$$\theta = \cos^{-1}\left(\sqrt{\frac{\frac{d^2}{h^2} - \Big(\frac{(x-x_0)}{b-(y-y_0)}\Big)^2}{\Big(\frac{d^2}{h^2} + \frac{(y-y_0)^2}{(b-(y-y_0))^2}\Big)}}\right) \tag{A.12}$$

Now, using equation (A.5) and (A.7), it is possible to express b in terms of known values. Let $\frac{h}{h-\tilde{z}} = H$

$$\begin{aligned}
\tilde{y} = \frac{a(y'-y_0)}{b-(y'-y_0)}\Big(\frac{1}{H}\Big) &= \frac{a(y-y_0)}{b-(y-y_0)} \\
\frac{(y'-y_0)}{b-(y'-y_0)} &= \frac{H(y-y_0)}{b-(y-y_0)} \\
\frac{b-(y-y_0)}{b-(y'-y_0)} &= \frac{(y'-y_0)}{H(y-y_0)} \\
b-(y'-y_0) &= \frac{(y'-y_0)b}{H(y-y_0)} - \frac{(y-y_0)(y'-y_0)}{H(y-y_0)} \\
b - \frac{b(y'-y_0)}{H(y-y_0)} &= (y'-y_0) - \frac{(y'-y_0)}{H} \\
b\Big(1 - \frac{(y'-y_0)}{H(y-y_0)}\Big) &= (y'-y_0)(1-\frac{1}{H})
\end{aligned}$$

and finally an expression for b in terms of known values

$$b = \frac{(y'-y_0)(1-\frac{1}{H})}{\Big(1-\frac{(y'-y_0)}{H(y-y_0)}\Big)} \tag{A.13}$$

Recall $d = 490$ inches, $h = 158$ inches and $\tilde{z} = 116$ inches. Therefore $\frac{h}{h-\tilde{z}} = H \approx 3.761905$.

The dimensions of each frame is 1280×1706 pixels. Recall that in the image the coordinate space, $x_0 = 0$ and $y_0 = -8644$. Pixel coordinates for (x, y) and $(x^{'}, y^{'})$ were located within a still of the video. Thus, $x - x_0 = 263$, $y - y_0 = 9468$, $x' - x_0 = 267$, and $y' - y_0 = 9612$. The remaining unknowns are then solvable and calculated to be $\theta = 1.48551$ in radians, $b = 9742.536$, and $a = h\cot\theta \approx 13.50805$.

Recall equation (A.1).

$$\begin{pmatrix} \frac{\mathrm{d}\tilde{x}}{\mathrm{d}t} \\ \frac{\mathrm{d}\tilde{y}}{\mathrm{d}t} \end{pmatrix} = \begin{pmatrix} \frac{\mathrm{d}\tilde{x}}{\mathrm{d}x} & \frac{\mathrm{d}\tilde{x}}{\mathrm{d}y} \\ \frac{\mathrm{d}\tilde{y}}{\mathrm{d}x} & \frac{\mathrm{d}\tilde{y}}{\mathrm{d}y} \end{pmatrix} \begin{pmatrix} \frac{\mathrm{d}x}{\mathrm{d}t} \\ \frac{\mathrm{d}y}{\mathrm{d}t} \end{pmatrix}$$

populating the Jacobian with the partial derivatives gives

$$\mathbf{J} = \begin{pmatrix} \frac{\mathrm{d}\tilde{x}}{\mathrm{d}x} & \frac{\mathrm{d}\tilde{x}}{\mathrm{d}y} \\ \frac{\mathrm{d}\tilde{y}}{\mathrm{d}x} & \frac{\mathrm{d}\tilde{y}}{\mathrm{d}y} \end{pmatrix} = \begin{pmatrix} \frac{a}{\cos\theta(b-(y-y_0))} & \frac{a(x-x_0)}{\cos\theta(b-(y-y_0))^2} \\ 0 & \frac{ab}{(b-(y-y_0))^2} \end{pmatrix} \tag{A.14}$$

The ground speed for a given pixel at a given time point is given by

$$\tilde{s}^2 = \left(\frac{d\tilde{x}}{dt}\right)^2 + \left(\frac{d\tilde{x}}{dt}\right)^2 = \left(\frac{dx}{dt}, \frac{dy}{dt}\right) \mathbf{J}'\mathbf{J} \begin{pmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{pmatrix} \tag{A.15}$$

Now we substitute the known variables into $\mathbf{J}'\mathbf{J}$.

$$\begin{aligned}
\mathbf{J}'\mathbf{J} &= \begin{pmatrix} \frac{a}{\cos\theta(b-(y-y_0))} & 0 \\ \frac{a(x-x_0)}{\cos\theta(b-(y-y_0))^2} & \frac{ab}{(b-(y-y_0))^2} \end{pmatrix} \begin{pmatrix} \frac{a}{\cos\theta(b-(y-y_0))} & \frac{a(x-x_0)}{\cos\theta(b-(y-y_0))^2} \\ 0 & \frac{ab}{(b-(y-y_0))^2} \end{pmatrix} \\
&= \begin{pmatrix} \frac{a^2}{\cos^2\theta(b-(y-y_0))^2} & \frac{a^2(x-x_0)}{\cos^2\theta(b-(y-y_0))^3} \\ \frac{a^2(x-x_0)}{\cos^2\theta(b-(y-y_0))^3} & \frac{a^2(x-x_0)^2}{\cos^2\theta(b-(y-y_0))^4} + \frac{a^2b^2}{(b-(y-y_0))^4} \end{pmatrix} \\
&= \frac{a^2}{\cos^2\theta(b-(y-y_0))^2} \begin{pmatrix} 1 & \frac{x-x_0}{b-(y-y_0)} \\ \frac{x-x_0}{b-(y-y_0)} & \frac{(x-x_0)^2+b^2\cos^2\theta}{(b-(y-y_0))^2} \end{pmatrix}
\end{aligned}$$

Recall $a = h\cot\theta$ and $\frac{\cot\theta}{\cos\theta} = \frac{1}{\sin\theta}$

$$= \frac{h^2}{\sin^2\theta(b-(y-y_0))^2} \begin{pmatrix} 1 & \frac{x-x_0}{b-(y-y_0)} \\ \frac{x-x_0}{b-(y-y_0)} & \frac{(x-x_0)^2+b^2\cos^2\theta}{(b-(y-y_0))^2} \end{pmatrix}$$

Now that we have $\mathbf{J}'\mathbf{J}$ in terms of known values, we plug it back in to get an expression for $\tilde{s}^2$.

$$\tilde{s}^2 = \frac{h^2}{\sin^2\theta(b-(y-y_0))^2} \cdot \left(\frac{dx}{dt}, \frac{dy}{dt}\right) \begin{pmatrix} 1 & \frac{x-x_0}{b-(y-y_0)} \\ \frac{x-x_0}{b-(y-y_0)} & \frac{(x-x_0)^2+b^2\cos^2\theta}{(b-(y-y_0))^2} \end{pmatrix} \begin{pmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{pmatrix} \tag{A.16}$$

The square root is the ground speed $\tilde{s}$ for all pixels (x, y) for any time point t.

Appendix B

Supplemental Tables and Figures

B.1 Local Area Maps

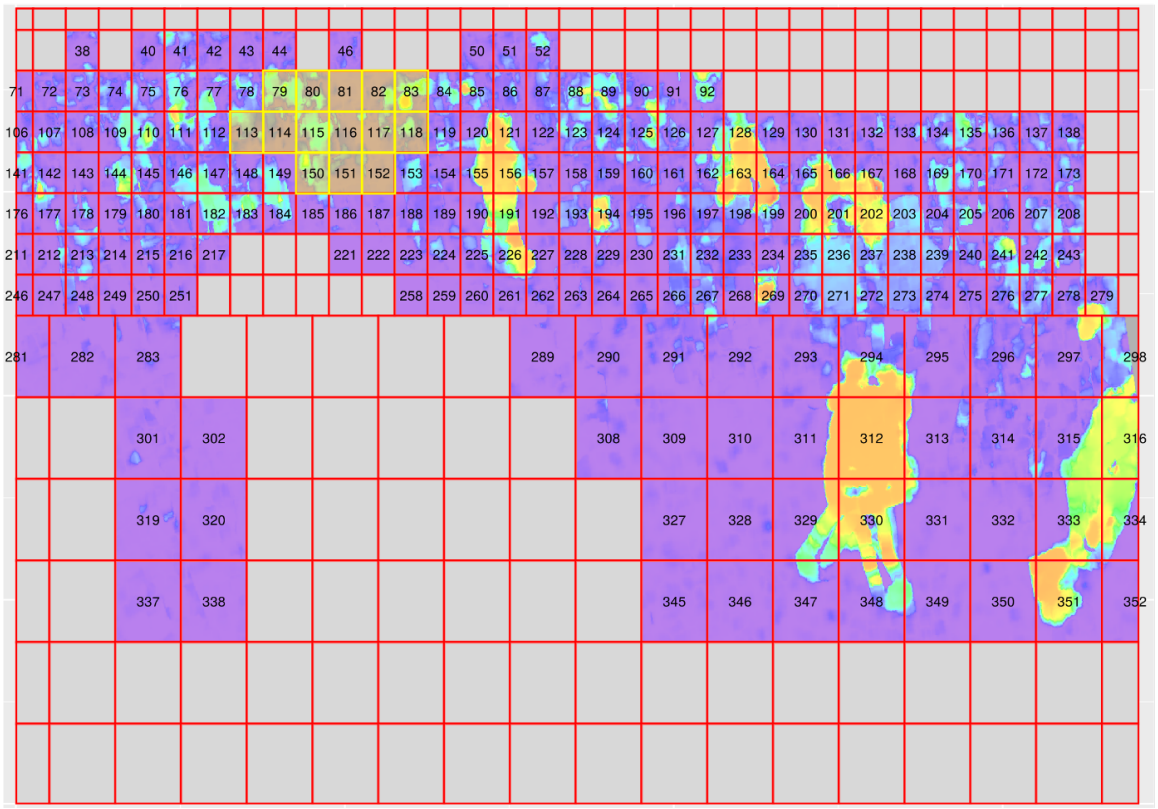

Figure B.1: Local area indexing for grid scheme # 1. Fighting regions are in yellow.

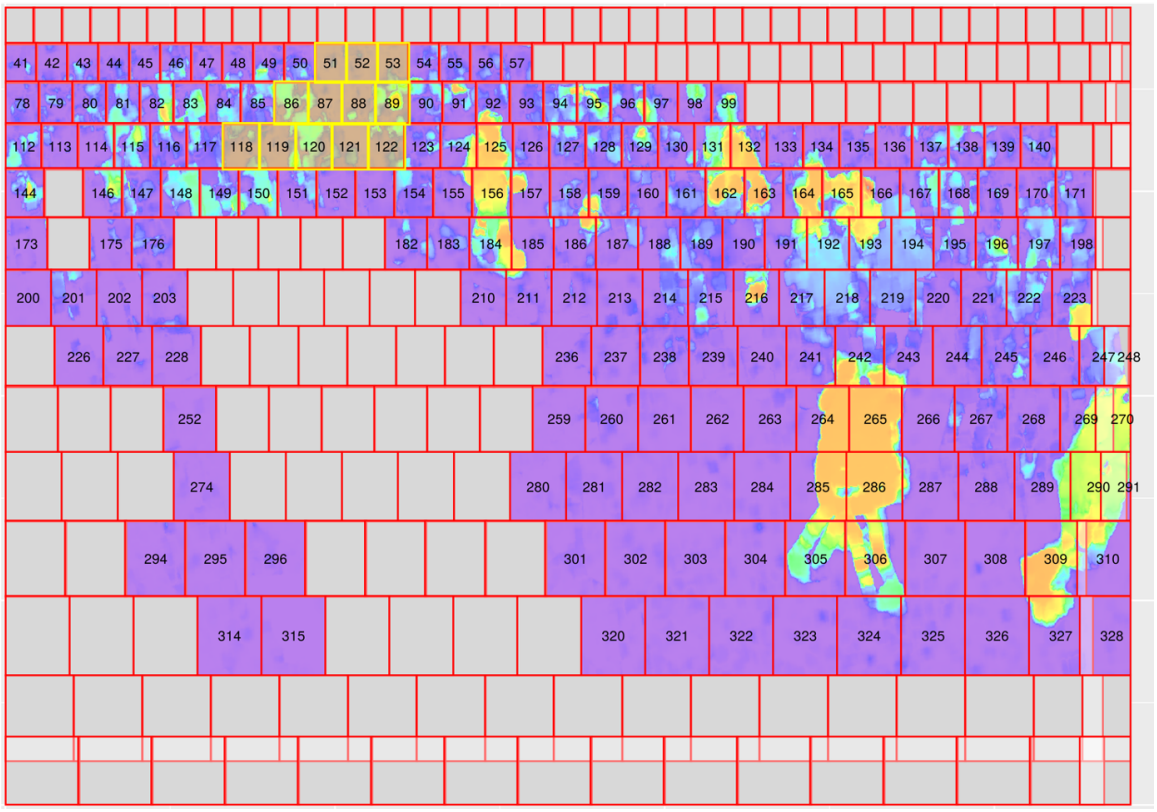

Figure B.2: Numbered local areas for grid scheme # 2. Fighting regions are in yellow.

B.2 Quadrant Divisions

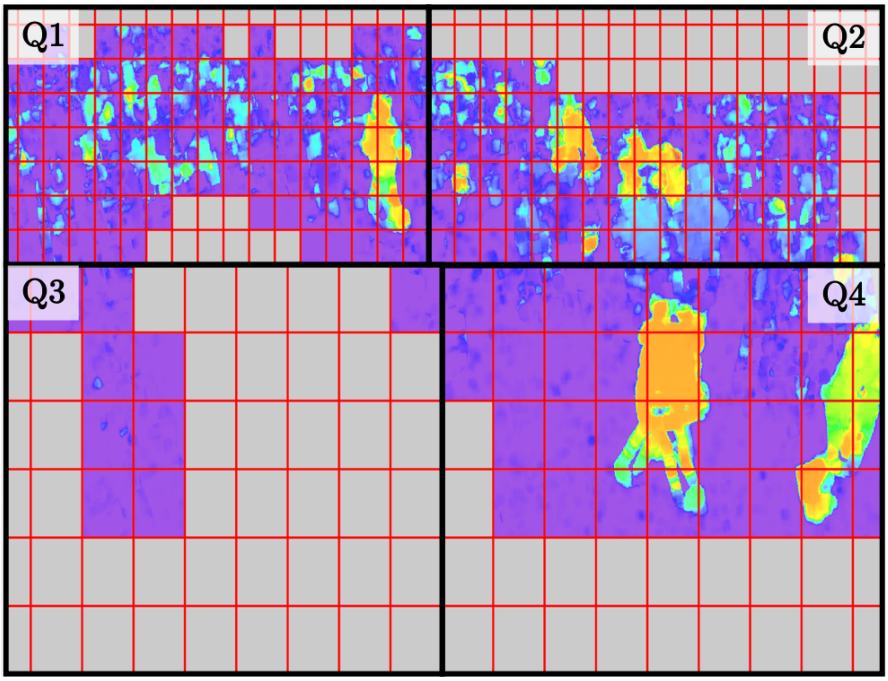

Figure B.3: The first grid scheme, separated into four distinct quadrants. These act as a guide to interpretation when considering variables in the whole frame.

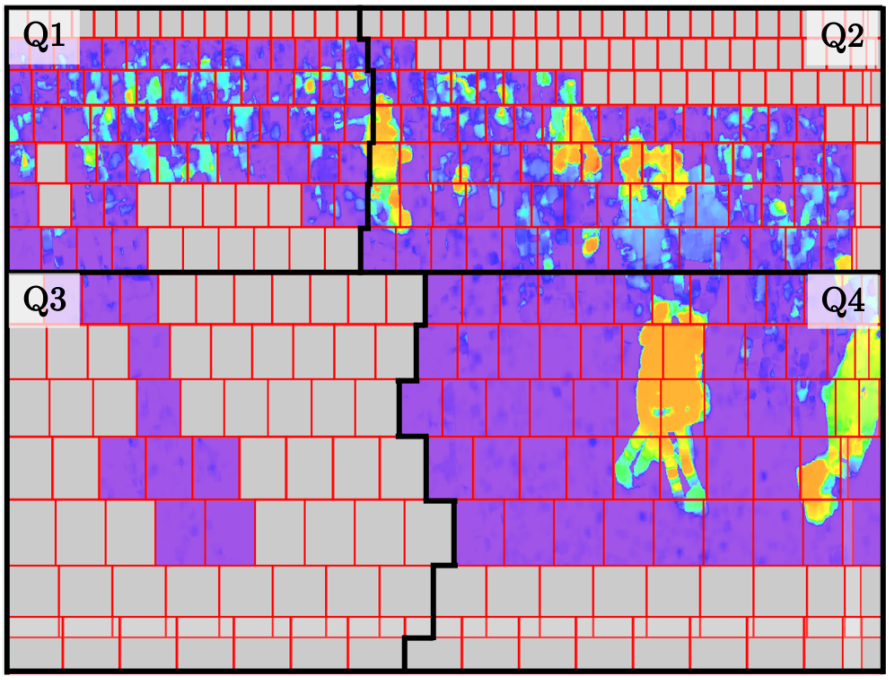

Figure B.4: Quadrants used in interpreting local areas in the second grid scheme.

B.3 Whole Frame Results

Results are presented for the random forest model when the entire frame is considered. Refer to Figures B.3 and B.4 when interpreting the "Quadrant" column.

B.3.1 Feature Selection

Ground Speed, Grid Scheme #1

	Rank	Area ID	Feature	Mean Decr. Gini	Quadrant
Full Model	1	334	95^{th} percentile	16.83	Q4
	2	351	95^{th} percentile	13.08	Q4
	3	352	95^{th} percentile	12.55	Q4
	4	352	Ratio of variances	10.88	Q4
	5	334	Ratio of 95^{th} percentile	10.68	Q4
	6	334	Ratio of variances	10.58	Q4
	7	334	Ratio of the mean	10.56	Q4
	8	301	95^{th} percentile	8.98	Q3
	9	132	95^{th} percentile	8.86	Q2
	10	352	Ratio of the mean	7.76	Q4
Reduced Model	1	334	95^{th} percentile	47.86	Q4
	2	352	95^{th} percentile	41.57	Q4
	3	351	95^{th} percentile	22.88	Q4
	4	301	95^{th} percentile	21.01	Q3
	5	132	95^{th} percentile	20.93	Q2
	6	167	95^{th} percentile	15.85	Q2
	7	80	95^{th} percentile	15.34	Q1*
	8	316	95^{th} percentile	14.25	Q4
	9	319	95^{th} percentile	13.00	Q3
	10	133	95^{th} percentile	12.49	Q2

* local area contains fighting

Table B.1: Top features selected by the full and reduced random forest models in grid scheme # 1 according to the mean decreased Gini index. Since only the 95^{th} percentile is used in the top three, it was the only feature considered in the reduced model.

Ground Speed, Grid Scheme #2

	Rank	Area ID	Feature	Mean Decr. Gini	fighting?
Full Model	1	328	95^{th} percentile	15.44	Q4
	2	310	95^{th} percentile	13.88	Q4
	3	327	95^{th} percentile	12.53	Q4
	4	310	Ratio of the mean	12.40	Q4
	5	310	Ratio of variances	11.94	Q4
	6	328	Ratio of the mean	11.85	Q4
	7	328	Ratio of variances	11.11	Q4
	8	290	95^{th} percentile	9.61	Q4
	9	310	Ratio of the 95^{th} percentile	9.48	Q4
	10	291	95^{th} percentile	9.45	Q4
Reduced Model	1	310	95^{th} percentile	42.46	Q4
	2	328	95^{th} percentile	42.05	Q4
	3	291	95^{th} percentile	41.56	Q4
	4	327	95^{th} percentile	31.02	Q4
	5	290	95^{th} percentile	23.93	Q4
	6	135	95^{th} percentile	21.73	Q2
	7	136	95^{th} percentile	14.48	Q2
	8	248	95^{th} percentile	12.91	Q4
	9	146	95^{th} percentile	9.50	Q1
	10	86	95^{th} percentile	8.89	Q1*

* local area contains fighting

Table B.2: Top features selected by the full and reduced models in scheme # 2, according to the mean decreased Gini index. Just as in Table B.1, the 95^{th} percentile is the only statistic in the reduced feature set.

Optical Flow, Grid Scheme #1

	Rank	Area ID	Feature	Mean Decr. Gini	fighting?
Full Model	1	334	95^{th} percentile	16.78	Q4
	2	352	95^{th} percentile	16.23	Q4
	3	351	95^{th} percentile	10.57	Q4
	4	132	95^{th} percentile	8.95	Q2
	5	352	Mean of values > 1	8.94	Q4
	6	301	Ratio of the mean	8.79	Q3
	7	301	95^{th} percentile	8.36	Q3
	8	334	Mean of values > 1	8.33	Q4
	9	80	95^{th} percentile	6.14	Q1*
	10	352	Ratio of the mean	5.56	Q4
Reduced Model	1	334	95^{th} percentile	46.00	Q4
	2	352	95^{th} percentile	35.25	Q4
	3	351	95^{th} percentile	27.26	Q4
	4	132	95^{th} percentile	21.74	Q2
	5	301	95^{th} percentile	19.09	Q3
	6	316	95^{th} percentile	17.61	Q4
	7	80	95^{th} percentile	13.88	Q1*
	8	133	95^{th} percentile	12.86	Q2
	9	167	95^{th} percentile	11.67	Q2
	10	319	95^{th} percentile	11.64	Q3

* local area contains fighting

Table B.3: Top features using optical flow data instead of ground speed.

B.3.2 Model Evaluation

Data	Grid	Model	OOB error	Accuracy	TPR	TNR	Precision
Ground Speed	1	Full	0.0030368	40.79%	94%	14.71%	35.07%
		Reduced	0.0034934	45.39%	86%	25.4%	36.13%
	2	Full	0.0017061	44.08%	92%	20.59%	36.22%
		Reduced	0.0024034	47.37%	92%	25.49%	37.70%
Optical Flow	1	Full	0.0013927	50.66%	90%	31.37%	39.13%
		Reduced	0.0012172	42.76%	74%	27.45%	33.33%

Table B.4: Random forest prediction results under all conditions where features in the entire frame were included in the feature sets.

Perhaps due to the fact that the top-ranked features are mostly irrelevant to the local areas of interest, it is clear that the models above are under-performing. Specifically, they are too sensitive and over-classify frames as "fighting". This is clear due to the discrepancy between sensitivity and specificity; positive frames are identified most of the time however looking at the specificity we see that the same cannot be said for negative results. Moreover, precision shows that meager levels of positively predicted values were truly positive. The model predicts well within the training data, but it doesn't generalize to the test data. This is mainly because of the high correlations among the frames around the same time. Had we split the data randomly for training and test data, the test accuracy would be much higher.

B.4 Q1-only Results

This section presents all results when only the first quadrant is considered in the random forest model. The full model variable ranking results were already presented in Chapter 5. Here they are shown along with their reduced model counterparts.

B.4.1 Feature selection

Ground Speed, Grid Scheme #1

	Rank	Area ID	Feature	Mean Decr. Gini	fighting?
Full Model	1	80	95^{th} percentile	12.12	yes
	2	114	Mean of values > 1	11.83	yes
	3	81	95^{th} percentile	10.75	yes
	4	144	Ratio of the mean	10.41	no
	5	179	Ratio of the mean	10.17	no
	6	183	95^{th} percentile	9.71	no
	7	179	95^{th} percentile	9.27	no
	8	114	95^{th} percentile	8.04	yes
	9	82	95^{th} percentile	7.99	yes
	10	144	95^{th} percentile	6.92	no
Reduced Model	1	80	95^{th} percentile	32.76	yes
	2	114	Mean of values > 1	21.80	yes
	3	81	95^{th} percentile	21.56	yes
	4	183	95^{th} percentile	21.15	no
	5	144	95^{th} percentile	19.96	no
	6	82	95^{th} percentile	18.29	yes
	7	179	95^{th} percentile	16.50	no
	8	114	95^{th} percentile	15.55	yes
	9	80	Mean of values > 1	12.82	yes
	10	115	Mean of values > 1	12.53	yes

Table B.5: Top ten features in the full and reduced ground speed models using grid scheme #1. The quadrant is not reported because all features included are restricted to Q1.

Ground Speed, Grid Scheme #2

	Rank	Area ID	Feature	Mean Decr. Gini	fighting?
Full Model	1	146	Ratio of the mean	15.45	no
	2	86	Mean of values > 1	11.70	yes
	3	175	Ratio of the mean	10.76	no
	4	87	Ratio of the mean	9.80	yes
	5	114	Ratio of the mean truncated at 0.1	9.51	no
	6	114	Ratio of the mean	9.43	no
	7	146	95^{th} percentile	8.76	no
	8	88	95^{th} percentile	8.66	yes
	9	53	Ratio of the mean	8.49	yes
	10	119	Mean of values > 1	7.50	yes
Reduced Model	1	146	Ratio of the mean	34.91	no
	2	114	Ratio of the mean	25.82	no
	3	86	Mean of values > 1	23.71	yes
	4	175	Ratio of the mean	22.38	no
	5	87	Mean of values > 1	22.03	yes
	6	119	Mean of values > 1	18.47	yes
	7	53	Ratio of the mean	17.54	yes
	8	123	Ratio of the mean	14.16	no
	9	85	Mean of values > 1	11.25	no
	10	88	Mean of values > 1	10.97	yes

Table B.6: Full and reduced random forest variable ranking using the second grid scheme.

Optical Flow, Grid Scheme #1

	Rank	Area ID	Feature	Mean Decr. Gini	fighting?
Full Model	1	80	95^{th} percentile	17.09	yes
	2	179	Ratio of the mean	14.12	no
	3	183	95^{th} percentile	9.63	no
	4	82	95^{th} percentile	8.80	yes
	5	179	Ratio of truncated mean at 0.1	8.13	no
	6	80	Ratio of mean	7.32	yes
	7	144	Ratio of mean	7.23	no
	8	214	Ratio of mean	6.67	no
	9	214	Ratio of variances	6.52	no
	10	46	Ratio of variances	6.28	no
Reduced Model	1	80	95^{th} percentile	27.86	yes
	2	179	Ratio of the mean	26.79	no
	3	183	95^{th} percentile	19.90	no
	4	82	95^{th} percentile	16.12	yes
	5	180	Ratio of the mean	13.92	no
	6	80	Ratio of the mean	13.78	no
	7	144	Ratio of the mean	13.41	no
	8	81	95^{th} percentile	13.28	yes
	9	214	Ratio of the mean	13.21	no
	10	215	Ratio of the mean	12.62	no

Table B.7: Top features selected in full and reduced models when using the optical flow data set.

B.4.2 Model Evaluation

Data	Grid	Model	OOB error	Accuracy	TPR	TNR	Precision
Ground Speed	1	Full	0.0069419	74.34%	86%	68.63%	57.33%
		Reduced	0.0048430	78.29%	94%	70.59%	61.04%
	2	Full	0.0045436	78.95%	80%	78.43%	64.52%
		Reduced	0.0076216	72.37%	90%	63.73%	54.88%
Optical Flow	1	Full	0.0045807	48.03%	42%	50.98%	29.58%
		Reduced	0.0026910	51.97%	44%	55.88%	32.84%

Table B.8: Random forest prediction results under all conditions that only consider features from the first quadrant. This table is identical to the model evaluation results presented in Chapter 5 (Table 5.6).

These results show that although it was previously shown that the model was not signifcantly improved through use of the ground speed data, there was some success if we focused on a certain region in the frame. Thus, the ground speed transformation does not result in the ability to generalize the model to the entire frame but does and success in combating a moderate level of regional discrepancies. In other words, although the ground speed transformation was not able to fully balance out the distance-related differences due to optical flow it did manage to improve discrepancies to a useful extent.

B.5 Classifier Cutoff Values

Recall that classifiers were calibrated using the first grid scheme, therefore the local area indices correspond to Figure B.1. Table 5.8 is shown here as a refresher.

Classifier 1	One optimized cutoff that is applied to all local areas.
Classifier 2	For the first 500 points, $\bar{x} + 4 \cdot s$. Calibrated for each local area separately.
Classifier 3	For the first 500 points, the max value of each local area.

Area ID	Classifier 1	Classifier 2	Classifier 3
79	1.73	1.26	1.05
80	1.73	1.05	0.91
81	1.73	0.86	0.76
82	1.73	1.24	1.39
83	1.73	1.90	3.99
113	1.73	2.43	4.51
114	1.73	1.61	1.44
115	1.73	2.31	2.21
116	1.73	2.85	2.34
117	1.73	2.50	2.99
118	1.73	3.20	2.98
150	1.73	2.60	2.78
151	1.73	1.53	1.87
152	1.73	1.66	1.64

Table B.9: Cutoffs used for each local area under all classifiers for the un-smoothed data.

Area ID	Classifier 1	Classifier 2	Classifier 3
79	1.59	1.13	0.77
80	1.59	0.89	0.73
81	1.59	0.74	0.55
82	1.59	0.98	0.87
83	1.59	1.42	1.12
113	1.59	1.85	1.69
114	1.59	1.26	1.13
115	1.59	2.01	1.58
116	1.59	2.61	1.69
117	1.59	2.26	1.95
118	1.59	2.80	1.84
150	1.59	2.23	1.87
151	1.59	1.29	1.01
152	1.59	1.41	1.07

Table B.10: Cutoffs used for each local area under all classifiers for data that are smoothed using a 15-point moving average.