Detecting Crowd Characteristics that are Indicative of Disorder and Violence in City Centre Locations from Video Data

Author: Kaelon Lloyd
Supervisor: Prof. Dave Marshall
Moderator: Prof. Paul Rosin

CM3203 – 40 Credits

Abstract

Surveillance cameras are placed around city centre locations in order to provide a method of identifying scenes of undesirable behaviour in real-time, if such behaviour is spotted then local law enforcement personnel can be called in to stop it. The job of the human observer can be very demanding as they are tasked with monitoring multiple live streams, it is inevitable that some instances of disorderly behaviour will not be caught. This project proposes the use of computer vision techniques to develop a method of action recognition that could be used to aid in the violence detection process. The abilities of multiple state-of-the-art action recognition algorithms will be compared alongside a newly proposed action descriptor referred to as GEP. Three widely different violence datasets are used to gauge the overall performance of each method; different datasets represent different types of violence seen within city street environments. Once results have been evaluated, efforts are made to increase action recognition performance of each implemented method beyond their original capabilities; this is accomplished by identifying points of description failure and improving upon them. Through the use of empirical testing methods it is shown that multiple action recognition algorithms provide great overall performance at detecting scenes of disorderly behaviour that are indicative of violence in city centre locations.

Acknowledgements

I would like to express deep gratitude to both Professor Dave Marshall and Professor Paul Rosin for providing advice and guidance throughout this project.

TABLE OF CONTENTS

TABLE OF FIGURES

INTRODUCTION

Around most major city environments a large multitude of closed circuit television (CCTV) cameras are placed strategically around city centre locations in order to detect undesirable behaviour; one of the main forms of undesirable behaviour in public areas is violence. In many cases, acts of violence can be cut short before any major damage is dealt providing that the situation is identified within a reasonable time-frame; constant real-time surveillance provides an ideal way of identifying such acts.

A CCTV observer is tasked with watching over a large array of monitors and acting appropriately when anything of interest is depicted; each monitor may display multiple video streams at once. The real-time surveillance observers must act on visual information alone as no audio is recorded. Monitoring a large number of live video streams requires the users' attention be spread out; this inevitably results in the occasional failure to identify scenarios of undesirable behaviour caught on camera. It is imperative that identification failure doesn't occur to ensure public safety.

In the research area of computer vision many methods of action recognition have been developed and show great performance at recognizing human actions using popular action datasets such as the Weizmann and KTH datasets; the actions depicted in these datasets are mostly of comprised of a single human performing one action at a time such as walking or waving. To evaluate the true effectiveness of action recognition within the context of this project a different set of data that has a heavy focus on violence is required; the hockey violence and Violent Flows datasets are available for violence recognition testing. These two datasets have previously been classified to a high degree of accuracy using various methods motion description methods. South Glamorgan Police has also provided Cardiff University with a dataset that contains scenes of violence in and around Cardiff city centre.

The overall goal of the project is to use computer vision techniques to develop and implement methods of action recognition that will be able to distinguish between scenes of violence and non-violence found in city centre environments. The purpose for doing so is to aid real-time surveillance observation personal in identifying scenes of violence that take place within city centre environments. The ultimate purpose for creating such software is to reduce the number of unidentified scenes of violence that may be missed by human personal and their limited capacity at being attentive at all times.

Background

K-Means Clustering

The aim of K-means clustering is to partition a set of vectors into *k* groups. Each group is represented by a centroid vector that is equal to the partition mean; a data point is assigned to the cluster which has the shortest data point to centroid distance (Tapas et al, 2002).

Lloyds K-Means clustering algorithm is outlined as a simple two stage iterative algorithm. The first stage assigns each point in the dataset a group; this is accomplished by using the Euclidean distance metric to find the closest centroid. The second stage re-assigns each centroid to the average of all points within their cluster. This two stage process will repeat until no further centroid changes take place or until an arbitrary iteration limit is reached.

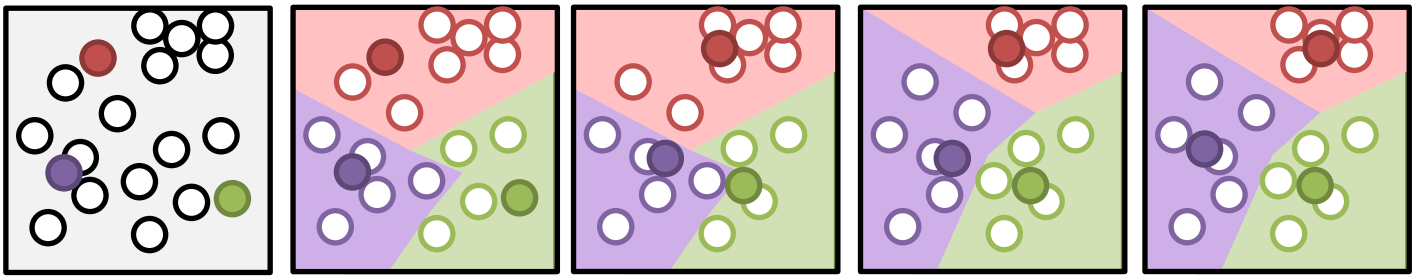


Figure 1: Stage 1: Centroid Selection, Stage 2: Find nearest cluster, Stage 3: Re-align centroids, Stage 4: Re-cluster data, Stage 5: Re-align centroids (Again).

The initial centroid positions are selected randomly, this does not guarantee good results; to combat this, k-means is executed multiple times and the solution that has the maximal difference between each centroid is chosen.

The implementation of K-means contained within the VL_feat toolbox (Vedaldi et al, 2008) is used exclusively for all clustering requirements. VL_feat is coded in C and wrapped in Matlab, it was chosen as it provides much faster clustering with a lower memory usage than inbuilt Matlab functions.

Principal Component Analysis

Principal component analysis is a multivariate analysis method that identifies the Principal components of data; this is accomplished by performing a linear transformation that projects a set of standardised data points into a new orthogonal co-ordinate system whose axes are placed such that the variance in the data is maximized; these new axes are the Principal components.

$$\text{Standardised}\ (n_i) = n_i - \frac{1}{N}\sum_{j=1}^{N} n_j$$

Mathematically, Principal components are computed by first standardising a set of data, this creates a zero mean and centres the data. After standardisation a covariance matrix is created by calculating the covariance between each dimensional pair; this will result in an array that describes how each dimension changes with respect to another.

$$cov(X,Y) = \sum_{i=1}^{N} \frac{(x_i - \bar{x})(y_i - \bar{y})}{N}$$

From the co-variance matrix both the Eigen values and Eigen vectors are calculated; the collection of Eigen pairs is sorted in descending order of Eigen value. The Eigen vector with the greatest corresponding Eigen value is the most important Principal component; all principal components are ranked in descending order of importance.

Higher ranking Principal components hold more information as they represent greater data variance. Using the fact that Eigen vectors are ordered based on the amount of information they represent, it is possible to reduce data dimensionality by omitting low ranking principal components that describe a very low amount of data. This is achieved by simply removing a set of Eigen vectors whose corresponding Eigen value falls below a threshold or doesn't meet set criteria; the criteria used throughout this project is to keep the most important principal components that collectively describe at least 90% of data variance.

Dimension reduction will be used on all descriptor data as it greatly reduces the computation time of K-means clustering while inflicting little to no reduction of descriptor effectiveness. Reducing data dimensionality also makes it possible to visualize data that would have been difficult to plot in Euclidian space otherwise.

Bag of Words Model

A bag of words model uses a collection of words that can be used to describe the contents of a document. The document representation is a histogram of word occurrences based on the words available in the model.

Given the codebook:

{1| The} {2| Apple} {3| Quick} {4| Lazy} {5| Jumps} {6| Over} {7| Fox} {8| Brown} {9| Dog}

The document "The Quick Brown Fox Jumps Over The Lazy Dog" can be formulated as the following vector of word occurrences:

{2 0 1 1 1 1 1 1 1}

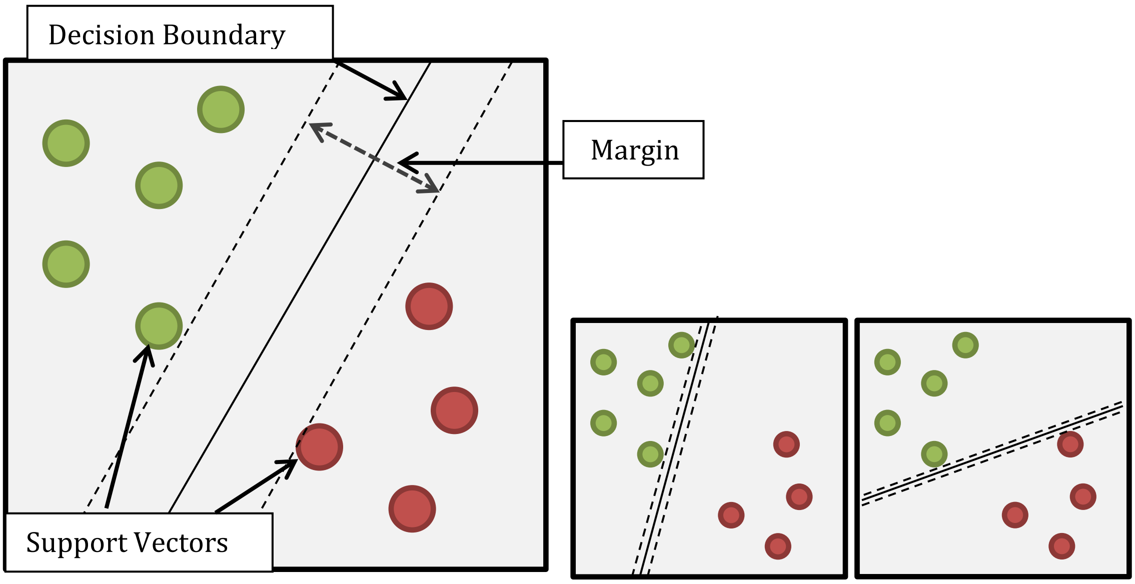
The idea of the bag of words model is adapted so that the model is a collection of features rather than English words; a scene or document can then be described as a collection of feature occurrences.

A bag of words code book is generated by performing K-means clustering on a subset of all features, the centroids returned from clustering will act as the vocabulary. When we wish to use the bag of words representation on new features, a search is executed in order to find the word in the vocabulary that best matches the new feature. The matching word index is then used to increment a counter within a histogram as demonstrated above. At every instance of vocabulary generation within this project, 200000 randomly selected features will be used.

Using word/feature frequency removes all geometric correlation between features within the same scene; this is both beneficial and detrimental to the description process. When analysing features found in multiple images there is no guarantee that the same feature that exists in many images will occur at the same position, in this case reduced spatial information is beneficial if we wish to match images based on their contents. Alternatively a bag of words approach wouldn't suit an algorithm that is trying to classify books that contain a specific sentence as the sentence structure wouldn't be captured.

SUPPORT VECTOR MACHINE (SVM)

A Support Vector Machine is a supervised learning method that when provided with a training set which is split into two classes will identify the optimal hyperplane of separation between them. The class separation hyperplane is obtained by projecting data points into a higher (potentially infinite) dimension and then maximizing the margin between support vectors (Chang et al, 2010).

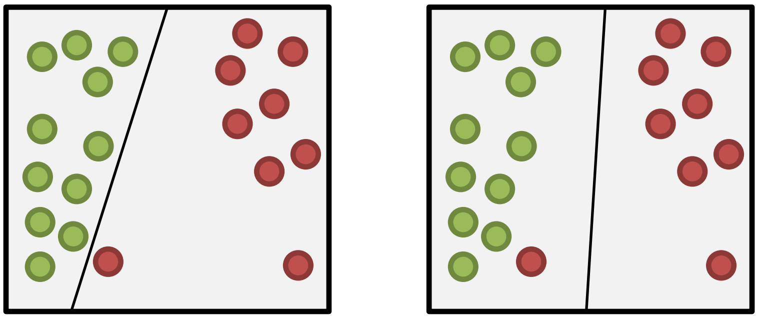


The margin between support vectors is maximized; this is a good boundary assignment

These are bad examples of hyperplane assignment.

FIGURE 2: EXAMPLES OF GOOD AND BAD SVM BOUNDARY SELECTION

All methods with be classified using C-Support Vector Machines. C-SVM uses a regularization parameter *C* that dictates the nature of group separation; the aim of the *C* parameter is to adjust boundary to avoid over fitting the data.



Large *C* value, Narrow Margin

Small *C* value, Wide Margin

FIGURE 3: HOW THE C-PARAMETER EFFECTS SVM

The rightmost hyperplane in the above figure, while failing to separate all points from each class, may achieve better results in general than a narrow margin boundary that could exist; for this reason when performing classification a grid search will be completed in order to find an optical regularisation parameter.

When data cannot be described using a linear hyperplane it is common to apply a Kernel function to describe the similarity between points. The Radial Basis Function (RBF) is one of the most common kernels for describing non-linear separation:

$$K(x_i, x_j) = \exp\left(-\gamma\|x_i - x_j\|^2\right), \gamma > 0$$

The kernel parameter γ is identified using a grid search on a set of training data. The LibLinear and LibSVM libraries are used for support vector machine learning (Chang et al, 2001).

Random Forests

Decision trees are simple data structures that are "grown" using a recursive method. In order to generate a tree, training data with associated class assignments must be provided. Growth is completed by partitioning data into two groups; this is accomplished by comparing values from a random subset of variables found within the supplied feature vectors. The newly partitioned groups are then subject to further partitioning based on a different variable subset extracted from feature vectors; each point that dictates data separation is known a "node", these nodes recursively branch off to other nodes. This recursive partitioning continues until all members of a partition belong to a single class, once a partition achieves class dominance it will not be partitioned again; these final partitions are known as leaf nodes.

A random forest is an ensemble classification method that is comprised of multiple decision trees; a predefined number of trees are grown using a random subset (with replacement) of all training data. Classification is achieved by sending new data through all decision trees, each tree will output a class that it assigned to the new data; the class assignments from each tree are treated as votes, the class with the most votes is the final classification.

Optical Flow

Optical flow describes the appearance of motion between two consecutive images by identifying features that exist in the preceding image and matching them to the similar features found within the following image; the spatial change of these features are expressed as a vector pair *u* and *v*. The magnitude signifies motion strength/velocity and vector orientation describes perceived motion direction.

Two methods of optical flow estimation have been implemented for this project; these are SIFT Flow and the Lucas-Kanade method. The major difference between these two methods is that the Lucas-Kanade method provides sub-pixel motion estimation.

SIFT Flow generates optical flow fields by matching Scale Invariant Feature Transform (SIFT) features across two frames and outputting each features vector of change. SIFT features describe image gradients as a histogram of gradient orientations. At every pixel position a 16x16 neighbourhood of pixels is split into 4x4 cells which are described using an 8 bin SIFT feature. As flow generation is accomplished by comparing detailed descriptors across two frames the results should contain less noise than other methods which use less robust features.

Optical flow can be described as the motion of pixel I(*x, y, t*) moves a distance of (*dx, dt*) over *dt* time, this can be expressed as:

$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$

The optical flow equation is derived from the above equation and is as follows:

$$f_x u + f_y v + f_t = 0$$

Where f_x, f_y are pixel intensity gradients and f_t is the first temporal derivative; solving this equation will provide the optical flow vector (u, v).

The Lucas-Kanade method aims to solve the optical flow equation by making two assumptions, the first is that the brightness intensity of moving objects stays constant; this is known as the Brightness Constancy Constraint.

The second assumption is that pixels within a neighbourhood have equal motion vectors; because of this the optical flow equation must hold for all neighbouring pixels, using a 3x3 neighbourhood we can generate a system of linear equation to solve the optical flow equation. The system of linear equations can considered over determined as we have more equations (9) than unknowns (2), to get around this, the linear least square principle can be applied to obtain the final equation (Bradski, 2000)

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_i f_{x_i} f_{x_i} & \sum_i f_{x_i} f_{y_i} \\ \sum_i f_{x_i} f_{y_i} & \sum_i f_{y_i} f_{y_i} \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i f_{x_i} f_{t_i} \\ -\sum_i f_{y_i} f_{t_i} \end{bmatrix}$$

Histogram of Oriented Flow (HOF)

To create a Histogram of Oriented Optical Flow, both *u* and *v* optical flow fields must be generated for a given area within two consecutive images. Flow orientations are computed and rounded into $360/n$ evenly spaced angular directions where *n* is the number of histogram bins used to represent the Histogram of Oriented Flows. Each histogram bin is incremented by the magnitude of respective flow.

Histogram of Oriented Gradients (HOG)

The creation of a Histogram of Oriented Gradients is accomplished by calculating the first *x* and *y* spatial derivatives L_x and L_y; these derivatives are used to obtain gradient magnitude and orientation.

$$|G| = \sqrt{L_x^2 + L_y^2} \qquad \theta = arctan\left(\frac{L_x}{L_y}\right)$$

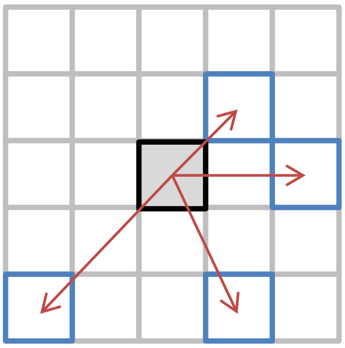
Gradient orientations are rounded into $360/n$ directions where *n* is the number of histogram bins. Each histogram bin is incremented by the magnitude of gradient intensity.

Typically an image is split into evenly sized cells and HOG performed on each segment; each HOG vector will then be concatenated together.

HOG differs from the aforementioned SIFT features in the way gradient magnitude is counted towards histogram bin values, SIFT uses weightings to describe edge contents across cell boundaries.

Grey Level Co-Occurrence Matrix (GLCM)

A GLCM is computed by counting the frequency that a pixel with a grey level value *i* occurs in conjunction with a second pixel intensity *j* given a pre-defined spatial relationship between the two. An offset map dictates the spatial relationship between a pixel of interest and its neighbours using vectors.



This figure represents pixel intensity co-occurrence using the offset map that is comprised of the following vectors.

(1,1) (2, 0) (-2, -2) (1, -2)

The correlation between each pixel pair is used to generate the correlation matrix.

Figure 4: GLCM Offset Vector

All pixel intensities are scaled to *n* different grey level values, in my usage of the Matlab in-built GLCM method the *n* parameter is set to 16; this results in a 256 element co-occurrence matrix.

Once a co-occurrence matrix is created, several statistics can be derived to describe the nature of image texture. Throughout the project I use only two texture measures, these are Energy and Contrast. Energy is a measure of texture uniformity; the maximum measure value of 1 is obtained when each element in the GLCM is equal which means that the image is constant.

$$\sum_{i,j} p(i,j)^2$$

Texture Contrast, also known as the sum of square variance, measures the intensity similarity between a pixel and its neighbours across the entire image. Soft textures are expected to have a low contrast value whereas hard textures will have high contrast values.

$$\sum_{i,j} |i-j|^2\, p(i,j)$$

APPROACH

The ultimate goal of the project is to identify methods of action recognition that allow for computer systems to differentiate between scenes of violence and non-violence within city centre environments. A discussion shall be made in order to evaluate whether or not the developed solutions are suitable for use in real world applications.

Through research I have found four different action recognition methods that aim to describe a wide range of action depicted in visual media.

- **Violent Flows**: Motion texture descriptor that uses optical flow fields.
- **Motion Binary Pattern**: Motion texture descriptor that estimates motion intensity using local changes in pixel intensity.
- **STIP+HOG+HOF**: An interest point detector with HOG and HOF descriptors.
- **Trajectory Histogram of Flows**: Describes long term temporal description of motion trajectories and their surrounding motions.

Both MBP (Baumann et al, 2014) and STIP (Laptev et al, 2008) have proven to be effective action description methods as they both score 91.83% and 91.10% respectively on the popular KTH action dataset which contains footage of different people performing a wide variety of actions, one of which is fast punching movements. STIP has also been previously compared against other local feature descriptors to determine the best method for violence detection between two individuals; it achieved 92 % classification accuracy on the Hockey violence dataset in a descriptor comparison paper (Bermejo et al, 2011). Although STIP performs well at describing scenes with few participants the local nature of its features will not be suited to more crowded environments; because of this, other, more global descriptors have been researched and have shown to perform well on violence that occurs in heavily populated situations. Violent Flows is an example of one such method and holds 81.30% classification accuracy in a dataset dedicated to densely populated scenes (Hassner et al, 2012).

It is important to investigate different methods as city street violence can form in many different ways with such variety; the expectation is that a single method cannot describe them all. In order to fully evaluate the performance of above-mentioned action recognition methods I must create a set of suitable tests that cover a wide range of violence types. Data used for testing must be indicative of scenes found within city centre locations; using scenes of violence from action movies for example, wouldn't suffice.

Once I have performed testing and evaluated the results I will propose an extension with the intention of improving classification performance; the proposed extension method will be developed by analysing results from initial testing; the test data will also be analysed so to identify any patterns or data measures that clearly describe the difference between violent and non-violent scenes. The change in performance between methods in their original form and their extended form will be evaluated and a final conclusion discussing whether or not my proposed extension improves classification will be made.

Each action recognition method implemented focuses on describing different types of data; because of this no one action recognition method it is expected to dominate all others when supplied with a variety of different violence datasets; combining different methods may show that a certain combination of descriptors can achieve high performance across all datasets.

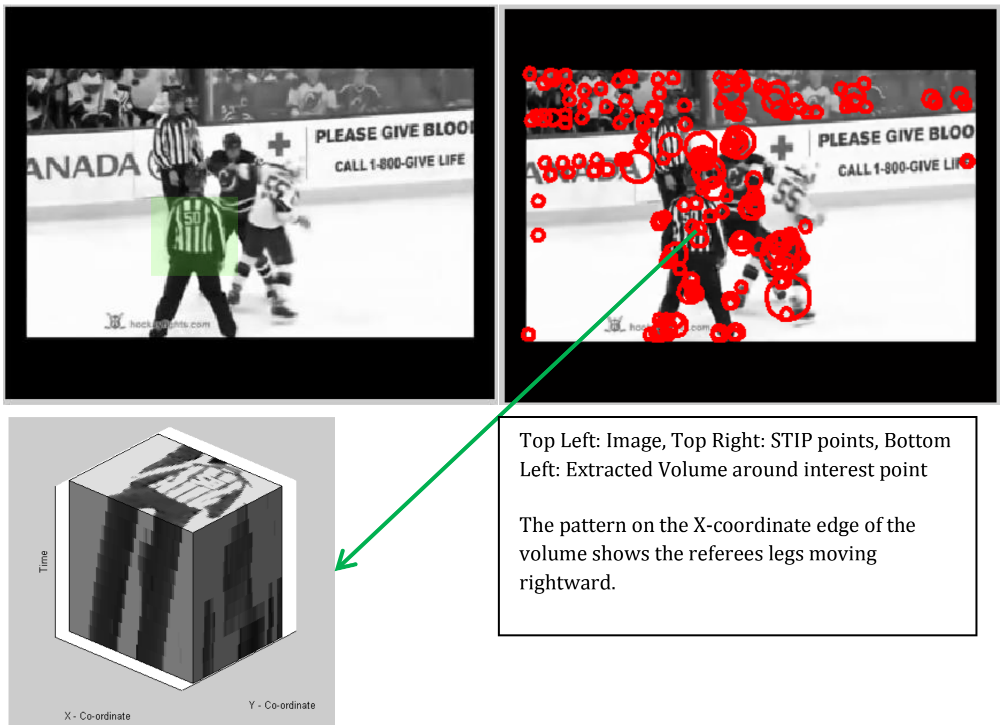
Violence Detection Methods

Space Time Interest Point with HOG and HOF

A corner is a feature found within an image that holds large intensity changes in multiple directions, these points of high pixel intensity variance contain large amounts of information; these points are known as spatial interest points and are translation and orientation invariant.

A Space Time Interest Point (STIP) is an extension of the Harris-Stephens corner detection method; alterations have been made so that interest points are identified by analysing spatial and temporal variations. STIP features just like the method they are based upon are not scale invariant; the original papers (Laptev, 2005) proposes a method of performing automatic scale estimation to best describe a point, this is computationally difficult and doesn't offer much, if any advantage over the alternative method of performing STIP detection over multiple scales (Bermejo et al, 2011).

Around each space time interest point a 3 dimensional volume is extracted; the volume shows how a 2D image segment evolves over time. The size of the 3D volume is based on the detected features scale. The spatial size of the volume is determined by $2k\sigma$ where σ is the feature scale determined by the STIP detector. The temporal scale is determined using $k\tau$ where τ is the temporal duration of a feature. The parameter k is assigned the value of 9. (Laptev, 2005)



Top Left: Image, Top Right: STIP points, Bottom Left: Extracted Volume around interest point

The pattern on the X-coordinate edge of the volume shows the referees legs moving rightward.

Figure 5: How STIP extracts frame volumes

The appearance of the 3D cube is identified using a Histogram of Oriented Gradients and the motion is described using a Histogram of Oriented Flows.

As suggested by Laptev (2008), the histogram of gradients is formed by splitting the volume into 3x3x2 cells with each section describing edges using a four bin histogram. Generating the histogram of oriented flows is also performed on a 3x3x2 volume split but motions are placed into 5 bins. The final descriptor is therefore a concatenation of both histograms resulting in a 162 element vector.

The STIP detection algorithm is a piece of closed source software written in C++, it was obtained from the creator's website (Laptev, 2008). HOG and HOF generation algorithms are included in the STIP detection package.

Modified Motion Binary Pattern

Motion Binary Pattern is a motion texture descriptor; it starts by making the assumption that motion can be detected from the change in pixel intensities. MBP requires three frames to describe motion.

Two difference binary maps are created by comparing frames n with n+1 and n+1 with n+2. At each pixel a value of 1 is assigned if the pixel intensity on the first frame is greater than the corresponding pixel intensity on the next frame. A 0 (zero) is assigned otherwise.

An 'exclusive or' function is applied to the two binary patterns which produces a third combination binary pattern depicting areas of perceived motion across three frames. At each pixel in this new binary pattern the L-1 norm is generated using a 3 by 3 grid.

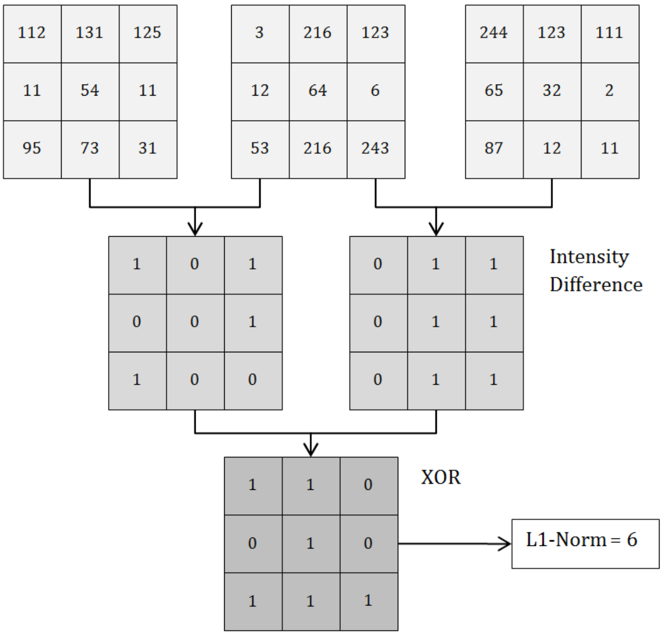


Figure 6: MBP Representation

The final binary motion pattern is created by assigning 1 where the L-1 Norm of each 3x3 window is greater than a set threshold and 0(zero) otherwise. Once the final binary pattern has been computed for a set of frames they are added together along the temporal plane to produce a single texture. The original MBP paper suggests re-arranging this texture into a 1 dimensional vector and performing classification upon it; the problem with this is that it has no spatial invariance whatsoever which limits its application.

The method proposed in the original paper has no spatial invariance whatsoever; this will be detrimental as important events within city locations can take place at different areas within the camera frame. To solve this problem I proposed splitting the final texture into M by N cells and introducing a bag of words models to quantize each cell; doing so will bring spatial invariance which will result in greater performance on dataset with large inter-sample spatial variance.

The above process will only have the ability to describe micro motions that due to the 3 consecutive frame description. The MBP paper proposes a method to extend the above processes ability to describe features over a longer temporal window through the use of a time step. As stated, comparisons are made between 3 adjacent frames; what a time step does is introduce an offset between these frames so rather than compare frames *n*, *n*+1 and *n*+2 you compare frames *n* , *n* + *t* and *n*+ 2*t* where *t* is the time step. *t =1* is equivalent to consecutive frames. The final descriptor is a combination of MBP descriptors taken at different time steps which are concatenated together.

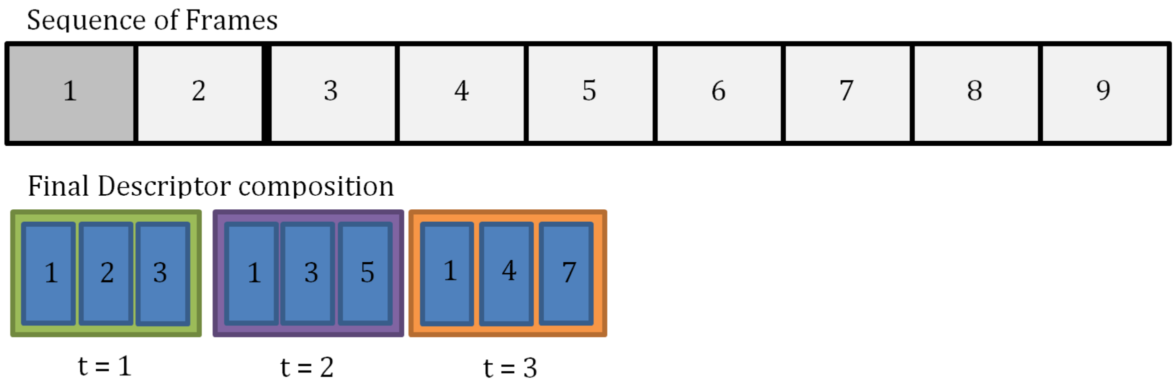


Figure 7: Visual representation of MBP time-step

Violent Flows

Violent Flows is a motion texture descriptor that uses optical flow for motion estimation. Given a sequence of frames, we must first perform optical flow estimation between adjacent frames. Optical flow will produce two flow vector maps *u* and *v* that dictate perceived motion in both the x and y co-ordinate plane; performing the following equation at each pixel will produce a magnitude map for each frame.

$$m_{x,y} = \sqrt{u_{x,y}^2 + v_{x,y}^2}$$

Using two consecutive magnitude maps a single binary map is generated that describes the change in motion between two frames. This is accomplished by taking the difference between two magnitude maps at each pixel and assigning a value of 1 if the resultant difference is greater than a set threshold.

$$\begin{cases} 1 \text{ if } \left|m_{x,y,t} - m_{x,y,t-1}\right| \geq \theta \\ \quad 0 \text{ otherwise} \end{cases}$$

The threshold is assigned to the average of absolute magnitude difference $\left|m_{x,y,t} - m_{x,y,t-1}\right|$. Using an adaptive threshold requires no parameter tuning and offers an easy method of identifying significant motion based on local dynamics; this is useful as all scenes have an expected degree of motion variance which a global threshold may not be able to adequately capture.

A mean magnitude map is computed by summing together a series of binary maps representing the change in motion between successive frames and normalizing the result.

$$\bar{b}_{x,y} = \frac{1}{T}\sum_{t} b_{x,y,t}$$

The resulting binary map is split into M x N non-overlapping cells. The contents of each cell are used to populate a fixed sized histogram; all cell histograms are concatenated to create the final ViF descriptor.

TRAJECTORY HISTOGRAM OF ORIENTED FLOWS

The aim of Trajectory-HOF descriptor is to provide long term temporal description of motion trajectory and its surrounding motion. The descriptor is a combination of a trajectory pattern and a Histogram of Oriented Flows describing the re-active motions that exist around the path trajectory. This method is derived from the paper *"Action Recognition by Dense Trajectories"* (Wang et al, 2011)*,* with the main change being the method in which trajectories are selected and composed.

Optical flow estimation is used to compute vectors of perceived motion between adjacent frames; given a series of flow vectors each point is traced in order to create a set of trajectories representing object motion across a sequence of frames. The amount of trajectory data is too vast to apply full description to each path and a lot of short trajectories simply represent optical flow errors or slight camera shudder; with this stated it is apparent that the trajectory set is too large and needed to be reduced.

First I created a value representing the energy of a trajectory with length T; it is computed as the sum of absolute difference between all successive points along a trajectory path:

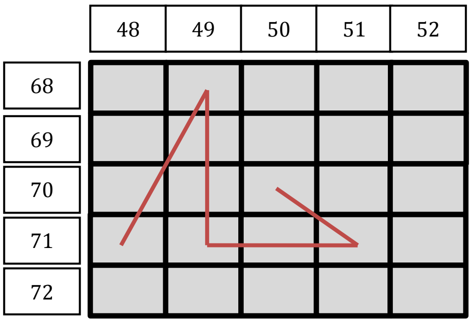
$$Energy = \sum_{t}^{T-1} |X_{t+1} - X_t| \; + \; \sum_{t}^{T-1} |Y_{t+1} - Y_t|$$

All trajectories with an energy value less than a set threshold are discarded. The threshold is dynamically assigned to the half the average energy of all non-zero energy paths.

$$Threshold = \frac{1}{2}\, mean(Energy > 0)$$

It was important not to set the threshold too high as small paths representing small motions can be useful in describing certain scenes; the aim of the threshold was set just to remove erroneous and zero length trajectories. Even with the application of a threshold we typically obtained too many trajectories, the amount of memory and computation time required meant processing them further is not feasible. To reduce the amount of data again I chose a random set of evenly distributed paths; an even distribution ensured that both the long and short paths that met the threshold requirement were used in scene description.

A trajectory is represented as two vectors, a series of *x* co-ordinates and a series of *y* co-ordinates that the path covers. A trajectory shape descriptor is formed by measuring the difference between adjacent points within each vector; the two resulting difference vectors will be concatenated into a single vector which will be *2* * (*T* - 1) units long given a trajectory length *T*.



Path trajectory:
{48,71}{49,68}{49,71}{51,71}{50,70}

Path Shape:
{1,3}{0,-3}{2,0}{-1,1}

Shape Descriptor:
[1, 0, 2, -1 ,3 ,-3 ,0 ,1]

FIGURE 8: TRAJECTORY HOF SHAPE DESCRIPTOR

An *N* by N area around each point along a trajectory path is extracted and split into σ x σ spatial cells and τ temporal cells; the parameters for *N* = 32, σ =2 and τ =3 resulting in 12 cells. For each cell a 9 bin histogram of flows is computed, 8 bins are used to encode vector direction and magnitude. The 9th bin is used for zero vectors.

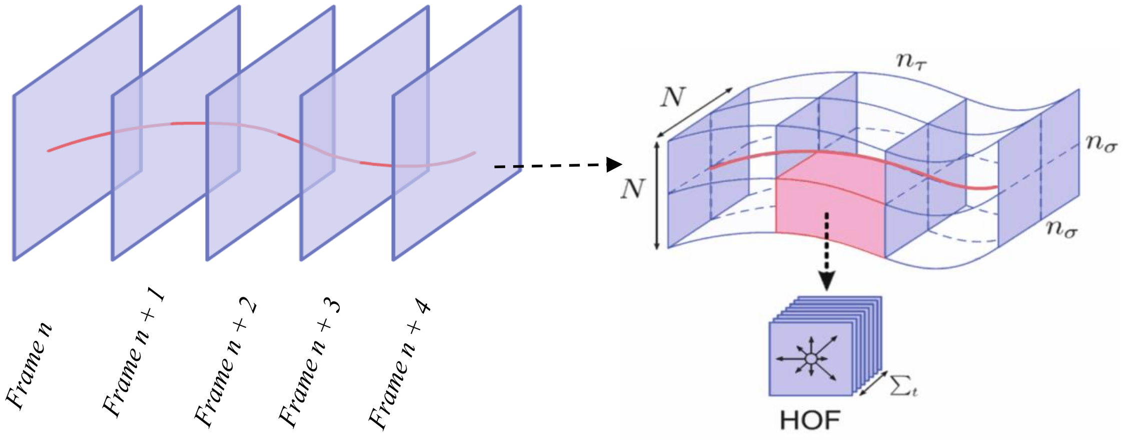


FIGURE 9: TRAJECTORY HISTOGRAM OF FLOW VISUAL REPRESENTATION

The trajectory shape and HOF are concatenated to create a trajectory descriptor. Within a window of length *T* frames, a set number of features are extracted and used with a Bag of Words model for quantization and then classification.

A spatial pyramid is created in order to re-introduce spatial information that is otherwise lost with the application of a Bag of Words model. A spatial pyramid is a multi-layer data structure that at each layer partitions all features into a different number of cells with each layer having increasingly more cells (Lazebnik et al, 2006).

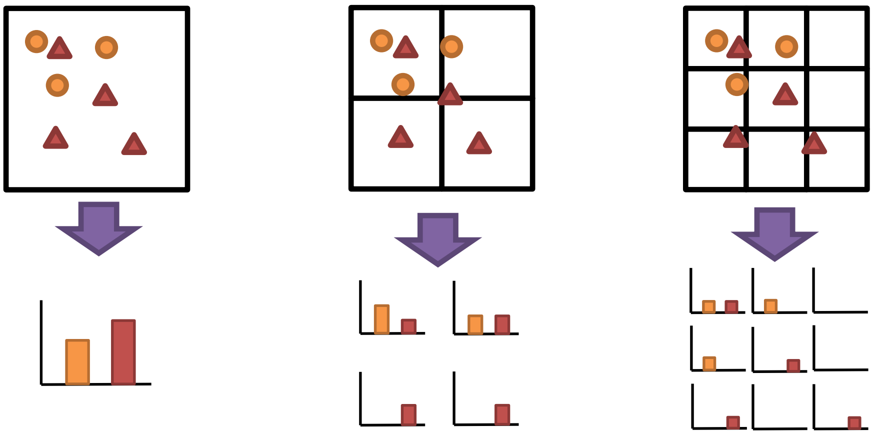


FIGURE 10: SPATIAL PYRAMID (FINAL DESCRIPTOR IS A CONCATENATION OF ALL HISTOGRAMS

The Trajectory HOF features are assigned cell locations based on their average path location:

$$(M_x, M_y) = (\overline{x_t}, \overline{y_t})$$

Choosing a trajectory length is difficult due to the variable nature of motion length, choosing a short trajectory length may not capture full motion and a long trajectory may over describe a motion; to alleviate this problem a trajectory is sampled at different lengths, and concatenated into a single descriptor.

GREY LEVEL CO-OCCURRENCE TEXTURE MEASURES, EDGE CARDINALITY AND PIXEL INTENSITY DIFFERENCE (GEP)

The aim of this action recognition method is to describe a set of actions using global visual descriptor methods. The measures highlighted in this method were originally developed in order to extend the four other motion based descriptors and increase their performance; all measures were chosen to suit the crowded nature of inner city environments. Analysis showed that these measures alone held enough discriminatory power to classify actions and so they were assigned their own method for comparison purposes. Throughout the remainder of the document this method will be known as GEP for ease of writing.

The method is comprised of four measures, texture energy, texture contrast, edge cardinality and pixel difference between adjacent frames. A grey level co-occurrence matrix will describe the brightness intensity relationship between pixel pairs within an image; using the grey level co-occurrence matrix you can derive multiple statistics that describe the textural nature of an image. Contrast and energy are computed by performing two different weighted sums on the same grey level co-occurrence matrix.

An edge cardinality measure is obtained by performing Canny edge detection and simply counting the number of edges perceived. The highest number of edges possible would be for every pixel to depict an edge therefore to normalize the edge count I divide by the frame dimensions. An edge count provides a rough indication as to the number of objects found within a scene, this is useful for distinguishing between scenes of different visual population.

A potential problem with identifying scenes using global visual descriptors is that the measures might be describing background visuals more than the important foreground; to reduce this possibility a fourth measures was introduced. The normalized pixel difference measure describes the amount of visual change between two adjacent frames; it is computed by calculating the sum of absolute difference between pixel pair intensities across two frames. The value is normalized by dividing by the maximum possible rate of change.

The descriptor is obtained by computing the above mentioned measures on a set of frames in sequence. All measures obtained from each frame are averaged and the variance computed resulting in an eight element vector that describes the basic visual representation of a frame set with an indication of the diversity they exhibit over a period of time. To include some spatial description, each frame is partitioned into M by N cells; each cell is described using the above method and cell descriptors are concatenated.

DATA SETS

Scenes of violence can be formed in a variety of different ways between variable numbers of participants; no dataset at the time of writing this document provides suitable number of violence instances that cover a wide spectrum of different fight types; existing datasets only focus on specific scenes of violence. South Glamorgan Police provided many video samples taken from within Cardiff City centre in order evaluate the possibility of automatic violence detection; the dataset contains a small number of fights whose visual variety is lacking, by looking at other scenes of violence found on video sharing websites, it was clear that the dataset was not fully representative of potential violence in city centre locations. In order to evaluate the true potential of each classification method other datasets were acquired, these are the Hockey Violence and Violent Flows dataset.

CARDIFF DATASET

The Cardiff dataset was provided by South Glamorgan Police so to determine whether or not computer vision techniques could be used to automatically detect scenes of violence. The dataset is flawed in a few ways; the biggest issue is the lack of violent video samples. Out of twenty three hours of footage, only seven fight instances spanning approximately 5 minutes total, exist. Three of these fights are heavily occluded by pedestrians, scenery and blurring caused by camera movement; these factors make all three scenes of violence unsuitable for use.

To introduce more information, eight extra fight sequences were obtained from video sharing sites and altered to match the frame rate and dimensions of all other video files in the Cardiff dataset. Sixty-six samples of non-violent data were extracted across the entire dataset; a reduced set was used as it would be too computationally expensive to process the entire dataset. In total the modified dataset contains twelve fight sequences and sixty-six non-fight sequences.

Longest Clip	1:37
Shortest Clip	0:10
Average Non-Violence Clip Size	0:51
Average Violence Clip Length	0:20

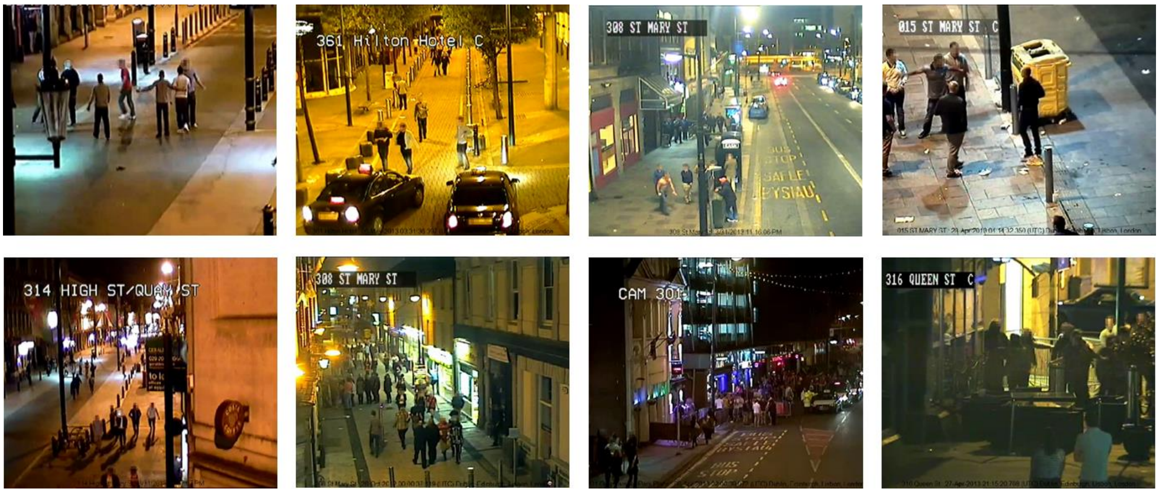


FIGURE 11: CARDIFF DATASET EXAMPLES

Even with a reduced dataset, the number of non-violence samples greatly numbers the quantity of violent samples; an unbalanced dataset may cause a classification bias. To get around this issue, videos are sampled using a sliding window, frames $k + n$ through $k + n + m$ are described using any of the descriptor methods where m is the window size and k is the frame offset in a video sequence.

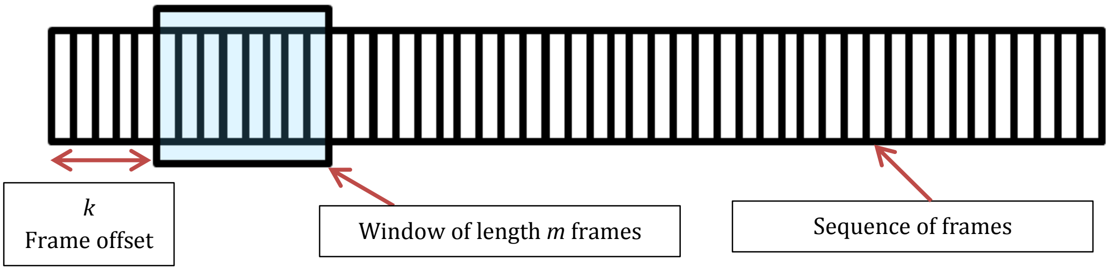


Figure 12: Visualization of the sliding window used to solve sample bias

Scenes that depict violent data will increase the frame offset by '1' between each sample; this will result in a large amount of data with a low amount of diversity. Non-violent data will have a larger frame offset increment of '15', which provides a similar number of samples for classification but with a larger variety of scene types.

The fights shown in the Cardiff dataset can be split into three distinct types:

1) A group of pedestrians (4+) merge together with force and barge each other. They are generally too close for effective punching and kicking.
2) The second fight type consists of 2-3 participants that stay in close proximity to one another and attempt to punch and kick without getting too close to their target.
3) The final fight is composed of 2 or more people that keep a great distance from their target, they occasionally move in for a quick swipe and then retreat a great distance.

The altered dataset contains four Type 1 and eight Types 2 fights. Only a single instance of a type 3 fight is available in the original dataset, unfortunately it is not unusable due to extreme blurring and occlusions.

The playback frame rate of the Cardiff dataset is six frames per second; this extremely low frame rate results in a lesser amount of available motion information. The high rate of change between adjacent frames may be too great for both implemented optical flow methods to approximate perceived motion accurately.

The dataset is split into four, non-overlapping subsets that will be used to perform four fold cross validation testing. Typically a larger number of data subsets are used in order to offer a better generalization of a classifiers performance on unknown, real world data. The limiting factor was the number of violence samples; as stated previously, only four Type 1 fight exist meaning that in order to fairly divide data only four folds could be created.

Violent Flows

The Violent Flows dataset was built to test violence that outbreaks within densely populated areas; the set contains 246 samples of both violence and non-violent scenes. The Violent Flows dataset primarily focuses on crowd behaviour at sporting events due to the extremely high pedestrian count. The Violent Flows dataset was chosen as a method of evaluating violence within city centre locations because scenes depicted have a reasonable chance of occurring in and around any city that has a sports stadium; Cardiff is home to the Millennium stadium.

An apparent flaw with this dataset stems from the fact that all videos are obtained from YouTube; the application of video compression required for internet streaming has degraded the visual quality of a few samples which makes feature extraction difficult.

The dimensions of each video vary between samples but they all share the same playback rate of 25 frames per second.

The Violent Flows dataset is divided into five subsets in order to perform five-fold cross validation; the original dataset creator had previously split the data so that no two folds contain footage from the same camera source. Due to the even number of source videos all folds contain a different number of class instances.

Longest Clip	0:06
Shortest Clip	0:02
Average Non-Violence Clip Size	0:03
Average Violence Clip Length	0:04

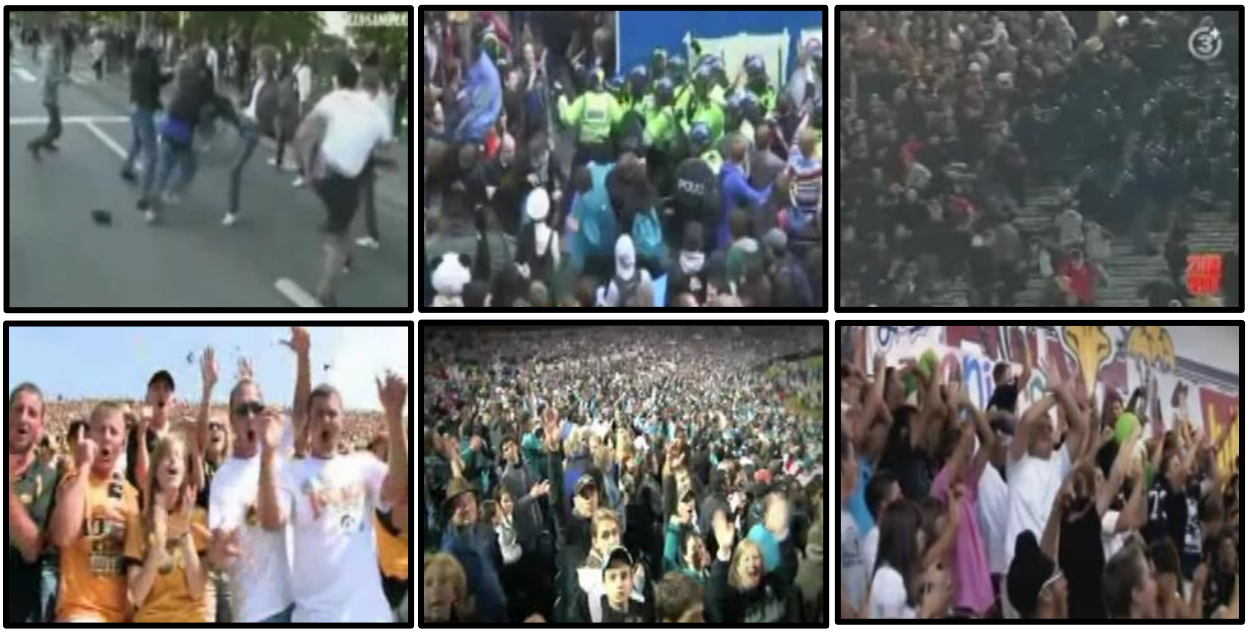


FIGURE 13: VIOLENT FLOWS DATASET EXAMPLES

HOCKEY DATASET

The hockey dataset contains scenes of violence from Ice Hockey matches; the fighting style found in the hockey dataset is unlike any fight found in the other two datasets. Violence typically involves two players punching each other primarily in the head. The range of motion during fights is extremely limited as players are required to keep their balance on ice.

The scenes of violence in the hockey data set consists of grappling alongside very pronounced punches, this style of fight does mimic one-on-one fights found between two people outside the hockey environment. The scenes of non-violence are however not indicative of city street behaviour as players tend to move extremely fast and hold themselves in odd positions; because of this, results from the hockey dataset tests should be considered less important than both the violent flows or Cardiff dataset test results.

The dataset consists of 1000 samples, 500 violent and 500 non-violent scenes. The contents of non-violent scenes are comprised of typical play found in Ice Hockey. The data is shot at 24 frames per second and has a resolution of 720x576.

Longest Clip	0:02
Shortest Clip	0:02
Average Non-Violence Clip Size	0:02
Average Violence Clip Length	0:02

The Hockey dataset is split into five, non-overlapping groups so to perform five-fold cross validation; each set contains 100 scenes of violence and 100 scenes of non-violence.

FIGURE 14: HOCKEY DATASET EXAMPLES

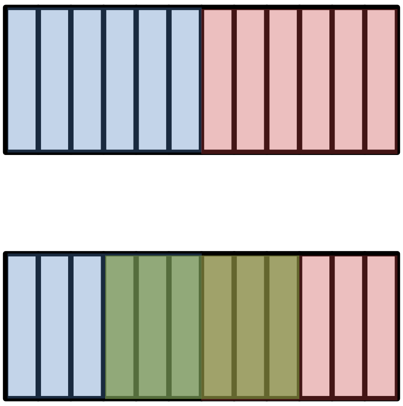
TESTING METHODS

DATA SAMPLING

The Cardiff and Violent Flows dataset is comprised of videos with variable frame lengths; typically the videos within the Cardiff dataset that depict fights are dramatically smaller (~100-250 frames smaller). Due to this difference, a descriptor may start to describe videos by their length. STIP will easily do this as more frames results in a larger amount of detected features which ultimately effects the composition of the final bag of words histogram such that scenes of violence will have marginally lower bin counts than non-violent histograms. Even if a descriptor normalizes its features temporally (Violent Flows) certain feature compositions are less likely to occur over few frames than they are over many frames, a method of comparing a constant number of frames must be incorporated to ensure fair comparisons.

All videos will be sampled in blocks, these blocks will constitute of a number of frames that will be described using each descriptor method; if each descriptor describes a constant number of frames then we the performance between different classification methods can be directly compared. Processing videos in this manner also fits the surveillance context of the project. Given a live video stream, only the past few frames could be used for classification; the number of frames in a real-time environment will be determined by either a set frame buffer or by computational limitations.

Samples are extracted using overlapping windows; the purpose of using them is to capture motion dynamics that two non-overlapping adjacent blocks only partially describe. Each sampling window will overlap their neighbour by half the window size unless otherwise stated.



A series of 12 frames using a 6 frame sampling window will separate data into two distinct samples as shown. The change in dynamics between these two windows isn't captured.

By using an overlapping window the dynamics between samples is captured.

FIGURE 15: BENEFIT OF OVERLAPPING WINDOWS

A shorter window overlap value could be used to provide more data for training and testing but the results wouldn't see too much change as neighbouring samples would share great similarities. Half window overlap provides more information while keeping neighbouring sample similarity at a minimum.

The vast majority of hockey samples have a frame length of 39, with a few having a length of 40. To include all samples, each video is processed using a sampling window length of 39. Leaving out the final frame of some videos is justifiable as actions depicted in each video occurs midway through the scene, the last few frames hold comparably very little descriptive power. The hockey dataset does not use overlapping blocks to extract samples as actions performed in each video are already very short.

The Violent Flows dataset will be processed in 40 frame length windows with a 20 frame offset between samples. The window of 40 was chosen as it is the equal in length to the smallest available sample.

The Cardiff dataset will be sampled using two different window lengths; the reason for this is that short window tests may fail to provide any useful results due to the reduced amount of descriptive power caused low frame rate of each video. The first window length is 30 (5 seconds), this is used to see how well fights are classified given a large amount of data. A short window length of 8 (1.1 seconds) frames is also used for a second, independent test. A shorter window will be used to gauge how well the Cardiff dataset can be classified in a real life scenario. Smaller windows allows for a quicker classification response which is important during surveillance.

Parameter Selection

Violent Flows: Requires three parameters, values *N* and *M* dictate how to split the final descriptor in order to re-introduce spatial information, these are assigned the value of 4 across all datasets. The third variable is the histogram size used to describe the magnitude of motions across a series of frames; this is set to 80 for all datasets.

Trajectory HOF: Requires two parameter vectors, vector *T* indicates the number of frames to trace a trajectory before applying description; a trajectory can be sampled at multiple lengths; in all cases a trajectory is described over the course of the entire sample. The vector *T* for each dataset is as follows:

- Cardiff Large Window: [10 20 30]
- Cardiff Short Window: [2 4 6 8]
- Hockey: [10 20 30 39]
- Violent Flows: [10 20 30 40]

The second vector defines the pyramid structure used to introduce spatial information into the final descriptor; all datasets use a two layer pyramid with the first layer representing a single cell and the second representing a 2 by 2 grid of cells.

Motion Binary Pattern: Threshold *T* is used to identify motion with large magnitudes; *T* is set to 7 for all datasets as it has shown to perform well. MBP uses a step vector that defines the number of frames to skip when obtaining three frames for motion estimation, this is set to [1 2 3 4 8] for all datasets.

STIP: The only parameter available decides whether to extract HOF, HOG or both descriptors around each point; this parameter was set to both.

GEP: A vector that states the spatial relationship used to form the grey level co-occurrence matrix is required. The vector is formed such that each pixel placed 45 degrees apart within a radius of 5 pixels are used to generate the GLCM. Using 45 degree steps introduces a limited amount of spatial invariance to the GLCM. Features within GEP are extracted over the entire frame as opposed to using cells.

As I am dealing with three separate datasets the option to tune parameters on a dataset level in order to maximize performance was available to me. In a real-life scenario an algorithm would not have prior knowledge of the type of data soon to be described; because of this fact, parameters that are tuned to perform best on a certain type of violence cannot be applied; therefore tuning dataset parameters would not provide a reliable indication of the overall performance of a descriptor method.

Determining the best solutions

During testing each descriptor method will return three different sets of results as they are classified using random forests, linear SVM and RBF SVM. Each set of results is represented by six performance values; these are true positive classification rate, true negative classification rate, specificity, sensitivity, overall accuracy and the area under the receiver operator curve (ROC AUC).

When given a list of multivariate vectors it can be difficult to intuitively say that one vector is better than the other. As each result is a series of values, a multi-objective optimization method is adopted to determine which result sets are objectively better than the others. These objectively better results are known as Pareto Optimal solutions and are considered non-dominated vectors; a vector is dominated if each variable value in one vector is smaller than their respective variable in another vector. All Pareto Optimal solutions create a Pareto Frontier; all solutions on the frontier are considered equally good. The application of subjective reasoning is required in order to evaluate their true effectiveness within context of the project.

Typically, measuring best descriptor performance is completed by identifying the method that provides the greatest overall accuracy or AUC value. However, these measures do not give any indication as to the number of correct classifications achieved per class; per class classification rates are important factors when deciding which methods are suitable for use in real world applications.

There are two perspectives to take when deciding which descriptor method is the best; the first will focus on the system's general ability to classify scenes of violence and non-violence by looking at overall accuracy and/or ROC AUC values, this can be seen as the typical way of interpreting results; the greatest accuracy or AUC values indicates the best method.

The second perspective requires some subjective reasoning to show that overall accuracy isn't as important as true negative classification rate; in the context of this project, the true negative rate is the probability that a scene of non-violence will be classified correctly. One of the project aims is to develop action recognition methods that aid a human observer at identifying scenes of violence. When put into perspective, an extremely high classification rate isn't necessarily required. Based on the samples in the Cardiff dataset, a fight lasts an average of 20 seconds, now suppose our classification method achieves a violence detection rate of 50 %; this will result in 10 seconds of alerts for an observer. Providing adequate attention grabbing power, 10 seconds of alerts, whether constant or rapidly intermittent, won't easily go unnoticed within a small timespan.

Applying the same logic to scenes of non-violence shows that even a high true negative rate will result in an absurdly high amount of false alerts due to the greater length of non-violent scenes. The average length of a non-violent scene in the Cardiff database is 97 seconds, statistically a true negative rate of 75% results in 22 seconds of alerts; depending on how intrusive the alert is this may become far too distracting. A highly distracting alert system will skew the observers' attention and make them less efficient at their job, and this would defeat the purpose of project.

A method that achieves a high overall classification performance may not necessarily be best suited for real world application as they may provide too many false alerts; for this reason when evaluating results I will identify both the best overall method for classification and the best method suited for real world application(if one exists).

To determine an acceptable threshold of false negatives a study must be completed; unfortunately I did not have the enough time to start this when evaluating results I shall state that any methods that achieves a true positive rate greater than 50% and a true negative rate of over 90% shall be deemed adequate for use in real-life applications as surveillance observation aid.

Initial Testing

The initial testing phase sees the five base descriptor methods tested against each of the four previously outlined datasets. The main purpose of the first step in testing is to evaluate results and explain why certain descriptors behave as they do.

K-fold cross validation is used in testing; what this means is that a dataset is split into k non-overlapping subsets. K number of tests will then be performed using k-1 subsets for system training with the remaining set is used for testing; the combination of data subsets that make up the training set changes on each test and so does the testing set. The reason for performing cross validation is to generalise system performance so that all training/testing set bias is removed, therefore the output of k-fold will be more representative of results expected in real-life circumstances.

Descriptor Method Extension Testing and Comparison

As mentioned in my approach, I aim to create some new descriptive measure that when added to certain descriptor methods will increase classification performance. This can only be completed after the first stage of testing has been performed as the results will give some indication as to the areas which can be approved upon.

Once an extension measure has been created the same testing process as before will be undertaken but all results shall be presented as a comparison between non-extended and extended descriptors.

Combinatorial Testing

One major issue with action recognition techniques is that a single descriptor method cannot describe all possible actions, they are generally suited to one kind of data or another; testing will reveal which of the five descriptors can describe which dataset adequately. As the expectation is that no one method will shine through on all tests a combinatorial testing method has been proposed.

Combinatorial testing involves merging two or more descriptors vectors together through concatenation. There are two reasons for doing this, the first is that two different descriptor methods may extract complimentary features that when combined offer a boost in performance. The second reason for combinatorial testing is that a merged descriptor may achieve greater performance across all datasets than any single method. Cross-dataset performance in important as all datasets tests are indicative of violence seen within city street environments and so to provide an adequate solution each form of data must be classifiable.

Overall best solution

The three datasets used within this project were all chosen because the scenes of violence depicted within them are indicative of violence found within city centre locations. To determine how well a single method or method combination performs overall I will simply be taking the average classification rates across all datasets. Once this move has been performed a Pareto frontier can be generated and best methods will be identified.

Results and Evaluation

Lucas- Kanade Optical Flow or SIFT Flow

Before full scale testing took place I needed to determine whether to use SIFT flow or the Lucas-Kanade optical flow method for motion estimation. Trajectory HOF, Violent Flows and STIP all use optical flow vectors for description. I have no control over the method used by the STIP algorithm as its closed source software; because of this determining which method was better was derived from results output from both Violent Flows and Trajectory HOF algorithms.

Due to time constraints alongside the insanely large amount of time required to tests all descriptor combinations I must identify the performance difference between using SIFT flow or Lucas-Kanade as performing each test with both methods will not be time feasible. To determine which method to use I will compare the overall classification accuracy for each dataset using both 'Trajectory Histogram of Flows' and the 'Violent Flows' methods.

Table 1: Comparison of optical flow methods using violent flows descriptor

Violent Flows Method	Sift Flow Random Forest	Lucas-Kanade Random Forest	SIFT Flow Linear SVM	Lucas-Kanade Linear SVM	Sift Flow RBF SVM	Lucas-Kanade RBF SVM
Hockey	68.98%	62.07%	65.16%	63.77%	49.70%	47.69%
Cardiff Short	72.34%	65.43%	71.08%	69.69%	62.81%	60.80%
Cardiff Long	71.94%	69.03%	69.74%	68.55%	58.32%	56.31%
Violent Flows	70.70%	68.69%	66.76%	65.43%	59.91%	57.90%

Using the Violent Flows description method across all datasets showed that SIFT Flow outperformed Lucas-Kanade optical flow estimation by 4.69%, 1.32% and 2.01% using Random Forest, Linear SVM and RBF SVM classification methods respectively.

Table 2: Comparison of optical flow methods using Trajectory HOF

Trajectory HOF Method	Sift Flow Random Forest	Lucas-Kanade Random Forest	SIFT Flow Linear SVM	Lucas-Kanade Linear SVM	Sift Flow RBF SVM	Lucas-Kanade RBF SVM
Hockey	74.10%	72.37%	65.66%	65.26%	49.20%	47.64%
Cardiff Short	78.00%	77.01%	70.45%	69.57%	62.22%	61.36%
Cardiff Long	77.24%	75.41%	61.52%	59.69%	58.27%	56.91%
Violent Flows	72.16%	70.33%	62.24%	62.08%	59.91%	59.67%

Using the Trajectory Histogram of Flows description method to classify each dataset has shown SIFT Flow to outperform Lucas-Kanade by 1.6%, 0.82% and 1.01% using Random Forest, Linear SVM and RBF SVM classification methods respectively.

SIFT Flow exhibits marginally better results over Lucas-Kanade optical flow and thus will be used for all other tests involving optical flow fields.

Individual Descriptor Method Results and Evaluation

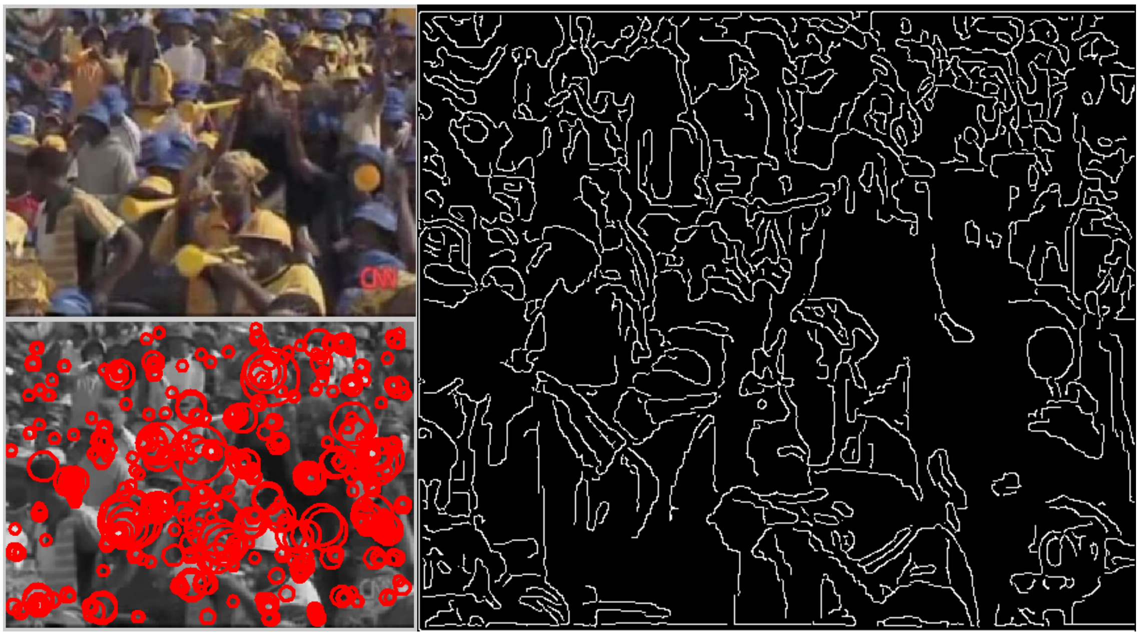
The following section will present the classification results for each dataset using STIP, GEP, VIF and MBP descriptors. At this point in the project I was running low on time and so RBF-SVM classification had to be omitted as the parameter tuning required takes a considerable amount of time.

TABLE 3: VIOLENT FLOWS DATASET RESULTS

Method	Classifier	True Positive	True Negative	Accuracy	AUC
GEP	RF	78.59%	68.73%	74.64%	76.58
	Linear	70.80%	**70.18%**	70.55%	74.98
STIP	RF	**91.97%**	50.55%	**75.36%**	78.75
	Linear	80.05%	63.27%	73.32%	76.99
MBP	RF	78.10%	38.91%	62.39%	63.38
	Linear	66.18%	52.73%	60.79%	63.86
Violent Flows	RF	84.67%	49.82%	70.70%	75.73
	Linear	75.91%	53.09%	66.76%	71.24
Trajectory HOF	RF	80.29%	60.00%	72.16%	**79.03**
	Linear	74.21%	44.36%	62.24%	60.73

The above results show that in all cases, violence detection achieves a higher classification rate than non-violence detection by a difference of 22.95%; this is a substantial inter-class difference and implies that features within non-violent data are either too weak or too varied between class samples. Weak motion features are the most likely candidate for this poor non-violent scene classification performance due to a combination of camera movement and tight crowd behaviour that heavily restricts the motion of pedestrians. Conversely, scenes of violence have little to no camera motion because they are focused on the fight and crowds tend to disperse from violence giving more room for attackers to make greater, more easily described motions.

GEP performs the best in terms of non-violence detection because it is describes visual information more than motion; the global texture measure describes the appearance of the entire crowd, this is marginally more descriptive that the local HOG features used in STIP; local HOG features extracted from a dense crowd will appear mostly as noise because edges that describe a person's shape will blend into their neighbours resulting in a descriptor that isn't indicative of any shape.

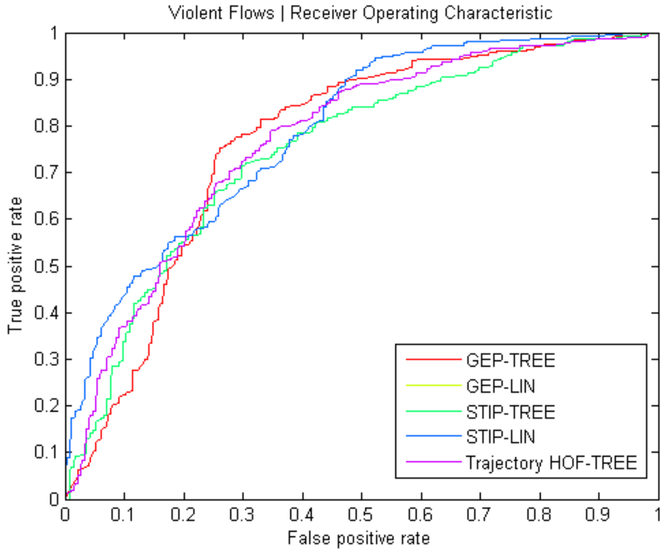


As crowd density increases up the image so does the number of edges. Shapes begin to merge and line composition can be described as noise.

TABLE 4: VIOLENT FLOWS DATASET FRONTIER

Method	Classifier	True Positive	True Negative	Accuracy	AUC
GEP	RF	78.59%	68.73%	74.64%	76.58
	Linear	70.80%	**70.18%**	70.55%	74.98
STIP	RF	**91.97%**	50.55%	**75.36%**	78.75
	Linear	80.05%	63.27%	73.32%	76.99
Trajectory HOF	RF	80.29%	60.00%	72.16%	**79.03**

The STIP descriptor provides the best violence classification rate and overall accuracy but achieves an abysmally low non-violence detection rate.



Although GEP using a linear classifier offers the best non-violence detection rate it is still far too low for use in real life surveillance observation due to the amount of false alerts that would signalled; as stated in the testing methods section, large numbers of false alerts can be distracting and actually distract users from identifying undesirable behaviour.

TABLE 5: HOCKEY DATASET RESULTS

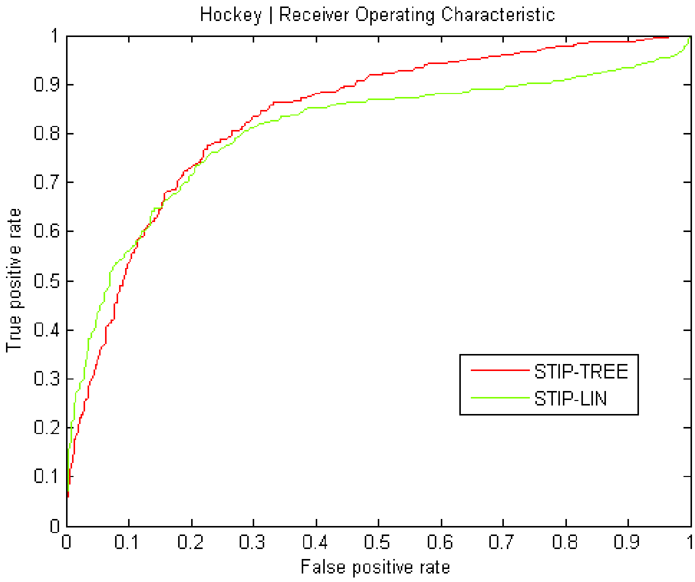
Method	Classifier	True Positive	True Negative	Accuracy	AUC
GEP	RF	76.86%	77.56%	77.21%	83.45
	Linear	69.42%	81.36%	75.40%	80.42
STIP	RF	**92.35%**	82.36%	**87.35%**	**95.04**
	Linear	84.10%	**87.37%**	85.74%	92.24
MBP	RF	78.47%	74.75%	76.61%	83.13
	Linear	72.84%	75.75%	74.30%	81.22
Violent Flows	RF	75.86%	62.12%	68.98%	75.72
	Linear	67.00%	63.33%	65.16%	71.68
Trajectory HOF	RF	77.46%	70.74%	74.10%	82.03
	Linear	66.80%	64.53%	65.66%	70.57

The expectation was that all texture descriptors would see low performance due to their global descriptive nature being used to describe sparse actions; contrary to this, both GEP and MBP hold good classification results. The fact that both a motion and visual descriptor method obtained reasonable results implies that hockey is not only highly classifiable based of motion but it can also be classified based on visual appearance separately.

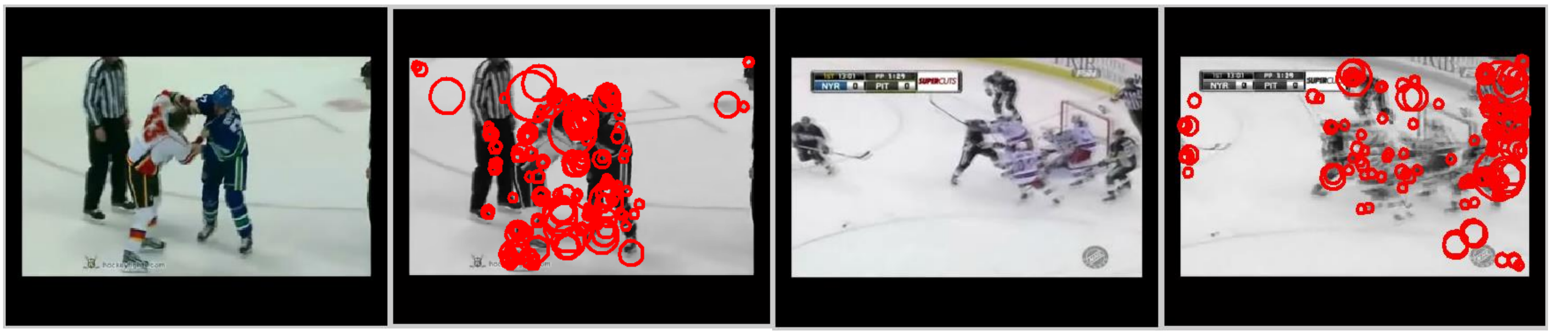
Motion blur is an artefact created by video capturing devices that do not have the capacity to record high speed motions. The footage in the Hockey dataset is clearly recorded using inadequate capture devices as motion blur can be seen in both scenes of violence and non-violence. The motion blur has an obvious effect on the chosen method of optical flow generation, SIFT Flow. SIFT Flow determines motion by matching SIFT features between frames; these features rely on distinct edges that aren't present during motion blur. The mediocre results seen by both Violent Flows and Trajectory HOF can be partially attributed to this fact.

TABLE 6: HOCKEY DATASET FRONTIER

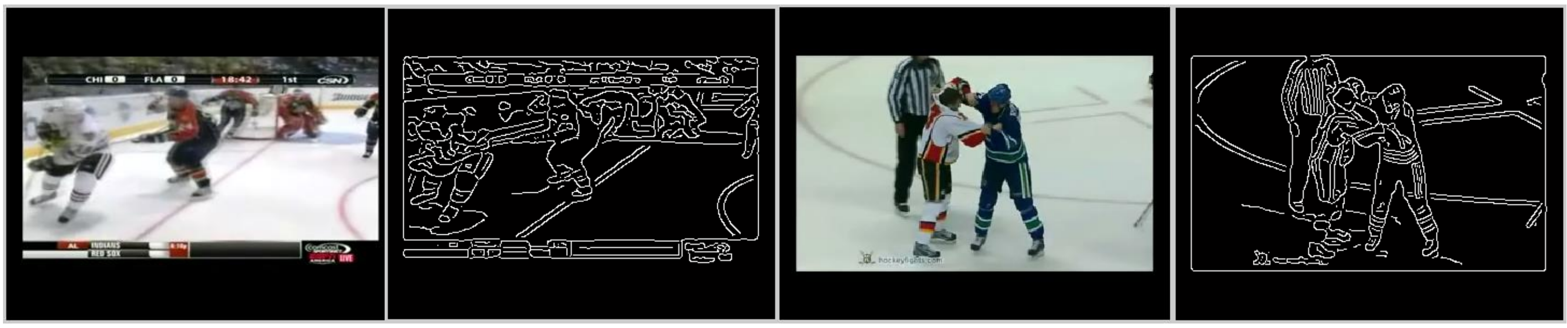
Method	Classifier	True Positive	True Negative	Accuracy	AUC
STIP	RF	**92.35%**	82.36%	**87.35%**	**95.04**
	Linear	84.10%	**87.37%**	85.74%	92.24



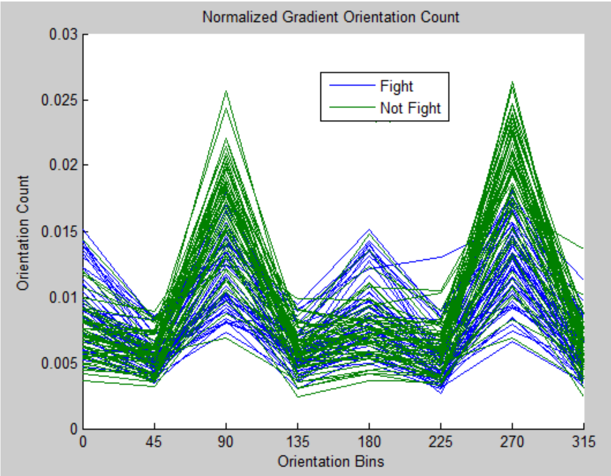
STIP is objectively better than all others methods, there are a few potential reasons for this. The first is due to the mostly plain background that is the ice arena; the surface enables the interest point detector to work effectively.

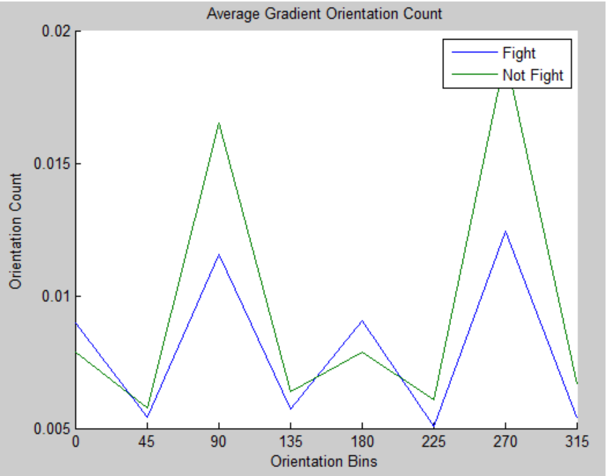


The plain background provides a huge benefit by giving players distinct edges. As shown by the application of Canny edge detection below, little information exists on the ice allowing for strong edges to be identified around the player.



The stance of players during standard play and fights is dramatically different, players tend to stand tall during fights and are highly bent over during play; this has an impact on the composition of each HOG descriptor as the posture of players will be dictated by the orientation of edges. To analyse the statistical difference in edge orientation between classes, all frames in each video sample were reduced to a histogram of edge orientation; the edge direction were rounded and put into eight bins representing eight directions 45 degrees apart.





Visually you can see that scenes of violence have a lesser amount of up/down oriented edges and slightly more left/right edges on average. To show that the edge orientation difference between classes is significant a similarity test is performed. The Mantel test is a measure of correlation between two populations; it is applicable on populations whose members are multivariate.

Mantel Similarity Test

R = -0.0204

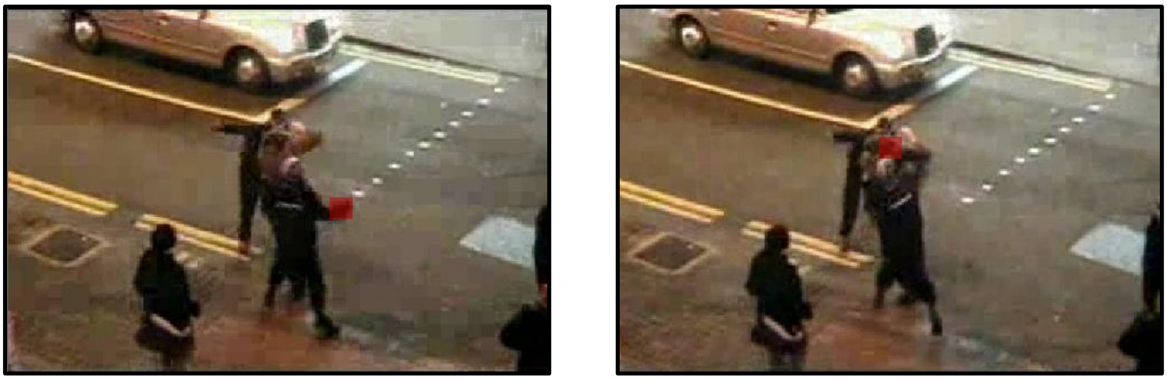
The Pearson correlation co-efficient R is a measure of correlation between -1 and +1. Negative 1 indicates a highly negative correlation while positive 1 indicates a highly positive correlation; a value close to zero implies little correlation and so the Mantel test implies almost no correlation between the two classes exist suggesting that they are independent.

Table 7: Cardiff long window dataset results

Method	Classifier	True Positive	True Negative	Accuracy	AUC
GEP	RF	72.04%	96.55%	86.33%	**96.65**
	Linear	**98.25%**	87.30%	**91.86%**	96.3
STIP	RF	18.75%	99.53%	65.86%	93.71
	Linear	21.27%	**99.84%**	67.09%	76.64
MBP	RF	12.94%	99.06%	63.16%	77.72
	Linear	22.26%	99.69%	67.41%	84.01
Violent Flows	RF	42.21%	93.18%	71.94%	78.73
	Linear	35.20%	94.44%	69.74%	72.79
Trajectory HOF	RF	52.85%	94.67%	77.24%	74.46
	Linear	31.47%	82.99%	61.52%	50.87

GEP is by a large margin, the best overall choice for violence detection on the Cardiff dataset; GEP using a linear classifier performs 14.62% better than the best performing motion based descriptor. MBP and STIP methods have an almost perfect True Negative classification rate but achieve an unsatisfactory rate of violence detection.

The poor violence detection within motion based methods can be attributed to the low playback rate of videos in the Cardiff dataset which is six frames per second. The following image pair shows two adjacent frames from a video within the Cardiff dataset; one member performs a clear punch to another person. The clenched fist of the attacker has been manually highlighted using a red box in both frames.



The distance travelled by the highlighted area between the two frames is just too large for either optical flow generation method to estimate; the lack of motion estimation means that this attack cannot be described. This problem arises throughout most scenes of violence throughout the entire dataset. The classification of non-violent data does not suffer from this problem as perceived motion in non-violent scenes has a lower rate of change than violent attacks; non-violent scenes also contain many examples of non-violent actions due to the magnitude of pedestrians in view, violent scenes however only contain a small number of actions indicative of a fight due to the generally low number of fight participants.

Edge composition during fight sequences between groups of people can also result in poor classification performance. As fight participants group together and merge closely, the edge composition will change. If the clothing of each member is similar then fights may show a lack of perceivable edges and therefore be almost impossible to classify using HOG; this also poses a problem for the SIFT based optical flow estimation which also relies on pronounced edges.

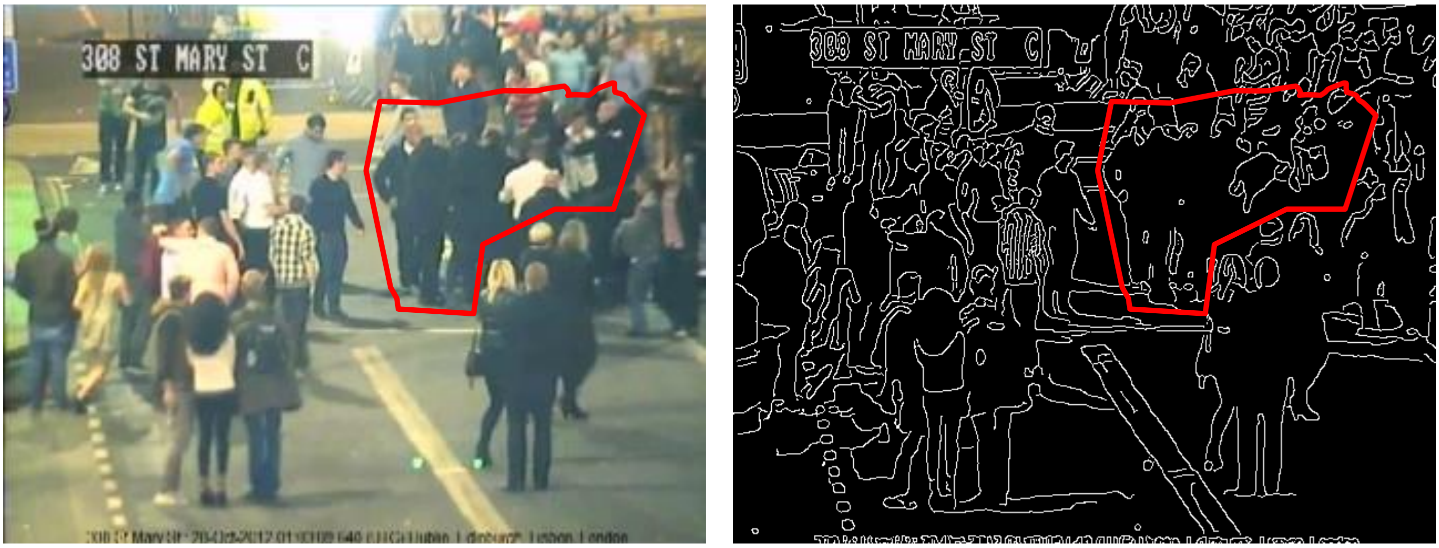


FIGURE 16: HIGHLIGHTED AREA OF VIOLENCE SHOWS A LOW NUMBER OF DETECTABLE EDGES

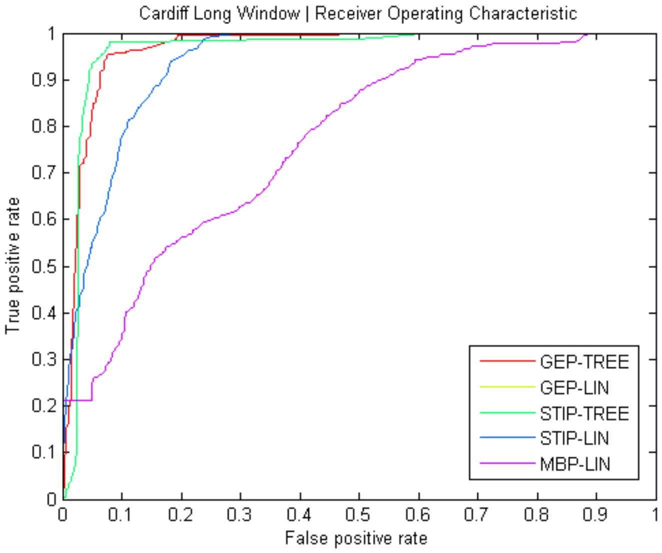
When people cluster together closely, even with each person wearing noticeably different clothing, the motion estimation will fall apart as each person occludes the movements of other people resulting in poor motion description, combine this fact with the effects of low video frame rate and it should be clear as to why motion estimation methods fail at identifying violence in the Cardiff dataset.

TABLE 8: CARDIFF LONG WINDOW DATASET FRONTIER

Method	Classifier	True Positive	True Negative	Accuracy	AUC
GEP	RF	72.04%	96.55%	86.33%	**96.65**
	Linear	**98.25%**	87.30%	**91.86%**	96.3
STIP	RF	18.75%	99.53%	65.86%	93.71
	Linear	21.27%	**99.84%**	67.09%	76.64
MBP	Linear	22.26%	99.69%	67.41%	84.01

The Pareto frontier consists of both visual and motion descriptors; each of the motion descriptors obtain an unacceptably low violence detection rate but achieve almost perfect non-violence recognition. GEP using a linear classifier is the best method for violence recognition in general.

Even though motion based methods score almost perfect results on non-violent data I cannot recommended them for use in real life applications due to the low violence recognition



rates. GEP alongside a random forest classifier is suitable for providing observation aid in real life applications as it achieves high classification for both classes but has a stronger emphasis on non-violent scene classification.

TABLE 9: CARDIFF SHORT WINDOW DATASET RESULTS

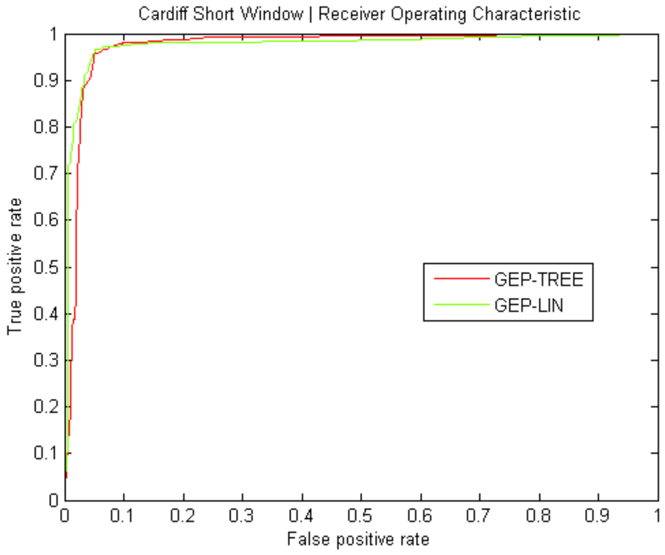
Method	Classifier	True Positive	True Negative	Accuracy	AUC
GEP	RF	85.97%	**97.01%**	91.91%	97.31
	Linear	**97.11%**	93.35%	**95.09%**	**97.58**
STIP	RF	46.51%	96.42%	73.36%	92.42
	Linear	30.10%	96.20%	65.66%	68.52
MBP	RF	28.66%	94.67%	64.17%	75.73
	Linear	22.87%	96.86%	62.67%	75.38
Violent Flows	RF	56.29%	86.12%	72.34%	74.43
	Linear	52.98%	86.63%	71.08%	72.63
Trajectory HOF	RF	71.43%	84.09%	77.36%	87.99
	Linear	61.39%	79.17%	69.72%	78.19

The reduced window size test shows an overall improvement of 2.26% over its larger window counterpart. Violence detection sees a 12.38% increase at the cost of a 1.90% decrease in true negative classification. When using long action samples, the probability that the descriptor describes two separate actions as one single action increases. The joint description of two separate actions may not be representative of the scene being described and thus just appear as noise; by using a smaller window we reduce this possibility which leads to more concise descriptions of individual actions.

The short window test will still suffer from all issues outlined in the large window Cardiff test as the properties of the dataset are unchanged.

TABLE 10: CARDIFF SHORT WINDOW DATASET FRONTIER

Method	Classifier	True Positive	True Negative	Accuracy	AUC
GEP	RF	85.97%	**97.01%**	91.91%	97.31
	Linear	**97.11%**	93.35%	**95.09%**	**97.58**



GEP description is the only method found within the Pareto frontier for short window classification on the Cardiff dataset; this implies that visual description of short window actions offer greater discrimination over motion based description methods.

The Linear SVM classifier produces the best overall performance while the Random Forest variant manages to obtain a higher true negative rate at the cost of true positive classification, the increase in non-violence detection outweighs the decrease in violence recognition making the random forest method better suited to real life use.

In conclusion, the first stage of testing has shown that each dataset can be reasonably described using one of the implemented methods of action recognition. Unfortunately the Violent Flows dataset could not be assigned a method that would prove adequate for use in real life applications due to the low, but still respectable true negative classification performance, hopefully either descriptor extension or combinatorial testing can resolve this.

Overall Performance

The overall average frontier is created using the average performance measures from the Hockey, Violent Flows and short window Cardiff dataset tests. Each of the three datasets are indicative of violence found within city locations, by looking at the average performance across all datasets we can gauge which method offers the best overall description of the various types of fight. When using visual computing techniques to automatically classify a live video feed you would require short observation windows from which to extract features; shorter windows equate to faster potential response times as less frames are needed for description. For this reasons when determining overall method suitability the large window Cardiff test is omitted as it is not representative of data used in real-life application.

TABLE 11: CROSS DATASET RESULTS

Method	Classifier	True Positive	True Negative	Accuracy	AUC
GEP	RF	78.36%	84.96%	82.52%	88.50
	Linear	**83.89%**	83.05%	**83.23%**	87.32
STIP	RF	62.40%	82.22%	75.48%	**89.98**
	Linear	53.88%	**86.67%**	72.95%	78.60
MBP	RF	49.54%	76.85%	66.58%	74.99
	Linear	46.04%	81.26%	66.29%	76.12
Violent Flows	RF	64.76%	72.81%	70.99%	76.15
	Linear	57.77%	74.37%	68.19%	72.08
Trajectory HOF	RF	68.72%	78.80%	75.37%	78.44
	Linear	54.68%	70.78%	64.97%	62.80

Both random forest and linear variants of STIP and GEP populate the Pareto frontier, the three other descriptor methods are objectively worse. No single methods offers a high enough true negative rate for it to be used as a real-time observation aid; the stated definition of suitability requires a greater than 90% true negative rate.

In conclusion, the results have shown that the Violent Flows dataset is the most difficult to describe as most methods don't see overall performance surpass 70% whereas the other datasets see high accuracy by at least one method. All four tests see either GEP or STIP as the best overall methods for classifying violence with GEP coming out on top overall.

It should be noted that global motion descriptors MBP and VIF perform less effectively than the the two local feature descriptor methods; this is an unexpected results considering that global motion description in theory should have performed better on crowded datasets.

Performance Increasing Measures

As mentioned previously in this document, the GEP descriptor wasn't supposed to be its own descriptor method; the intent behind the features contained within GEP was originally used to inject general visual information into the other four methods so to improve classification performance.

Ignoring the results output from the GEP descriptor in the previous section, the Cardiff dataset has the worst true positive (violence) classification rate; this can be attributed to the low visual variety of fight samples alongside the extremely low frame rate. The results show that motion description alone isn't powerful enough to discriminate between classes; from this it is clear that in order to create a more effective classification method visual attributes must incorporated. STIP does provide a method of visual appearance description but it still obtains some of the worst classification results. This would suggest that the local nature of STIP based HOG description isn't suited to the crowded nature of city street violence. As local features were not useful I found myself researching global measures of visual image composition that would be able to describe an entire scene as opposed to describing small sub-sections.

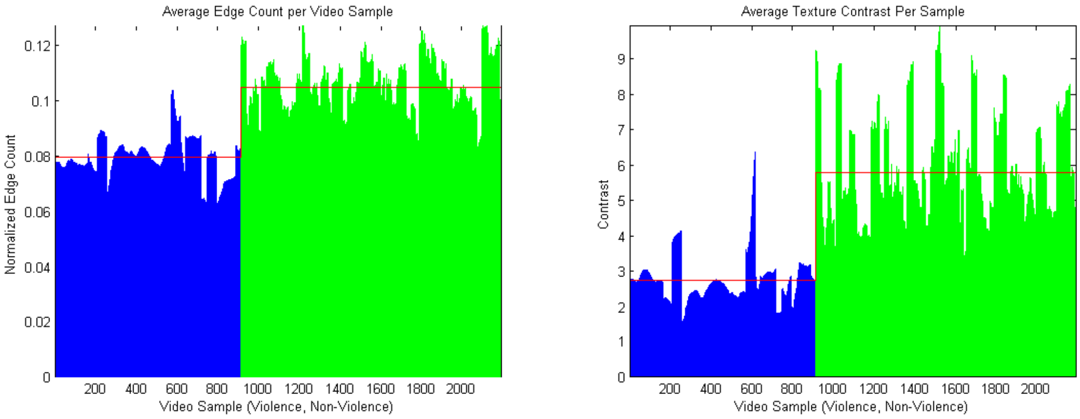
Measures of texture seemed like one of the more obvious choices of visual description as they could easily describe the crowd density of an image; crowded scenes would have lots more pedestrians which would create an un-even surface and would represent a rough texture whereas sparse city street conditions would be smoother as less noise exists within frame. GLCM texture energy is the measure used to describe the uniformity of a scene and GLCM texture contrast is used to describe the co-occurring pixel intensity variation.

Edge cardinality was the third measure investigated; it is the normalized edge count when a Canny edge detector is applied to an image. Edge cardinality gives a rough indication of the number of on-screen objects; scenes with many edges are generally more populated visually. The problem with edge cardinality and both texture measures is that they can potentially describe the background which may lead to classification by street rather than the pedestrians' configuration within the streets. To provide some non-background describing information I also added a measure that is created by taking the normalized absolute difference of pixel intensities between adjacent frames.

Each dataset is comprised of video samples that are made of multiple frames, in order to describe each sample, each of the four visual measures is computed for every frame, the average and variance is then calculated across all frames in a sample resulting in an eight measure vector that describes the global visual composition and diversity shown over a number of frames.

To evaluate the usefulness of the four aforementioned measures I needed to determine whether or not the measures extracted from violent data were significantly different from the same measures taken from non-violent data; one of two statistical methods was used to do this, either Students t-test or the Wilcoxon rank-sum test. Student's t-test typically provides more accurate results than the Wilcoxon rank-sum test but requires that the data population to be normally distributed; to test for population normality a Kolmogorov-Smirnov test was performed. If the normality test failed then the non-parametric rank-sum test is performed.

Looking at the edge cardinality and texture contrast measures across the Cardiff dataset we can see that either one of these will provide reasonably high discrimination between scenes of violence and non-violence on their own.



Test: Wilcoxon rank-sum
Difference Type: Significant
P-Value: 1.7774e-080

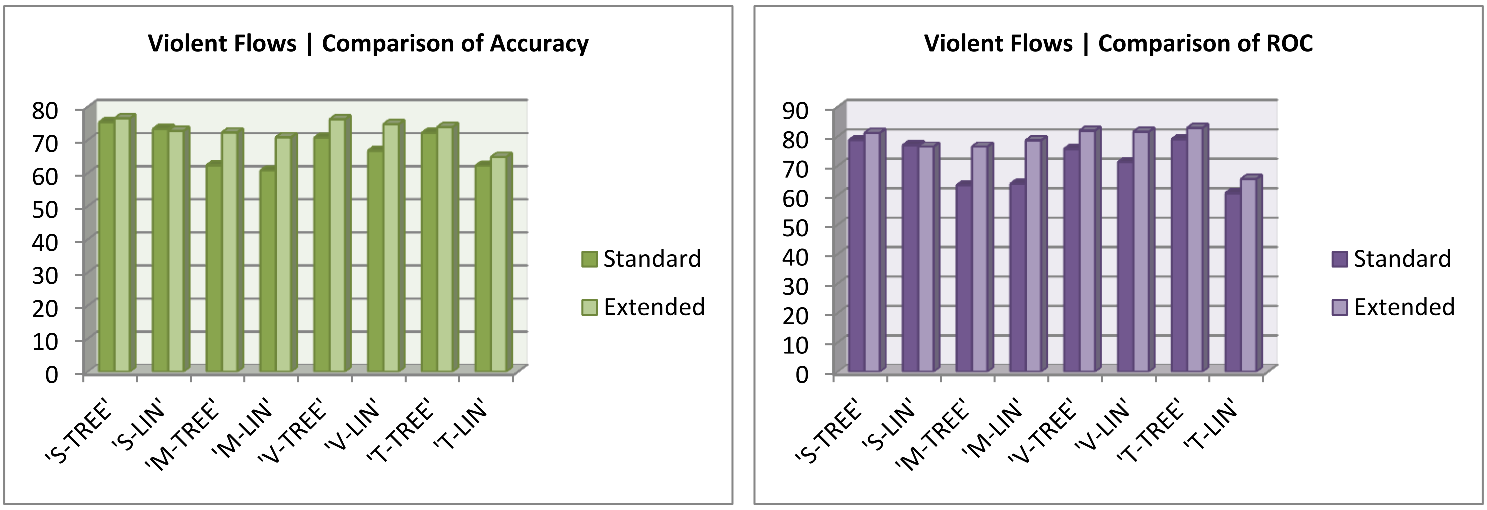
Test: Wilcoxon rank-sum
Difference Type: Significant
P-Value: 2.5987e-018

FIGURE 17: CARDIFF EDGE CARDINALITY AND TEXTURE CONTRAST

The p-value represents the idea that the difference between populations occurs by chance; if a small p-value is returned you can reject the idea that the difference between populations has occurred randomly.

Almost all measures across each dataset showed a significant difference between classes (Appendix A, B, C) with the exception of texture energy and contrast variance on the hockey dataset (Appendix C5 C8).

Results of Extended Description

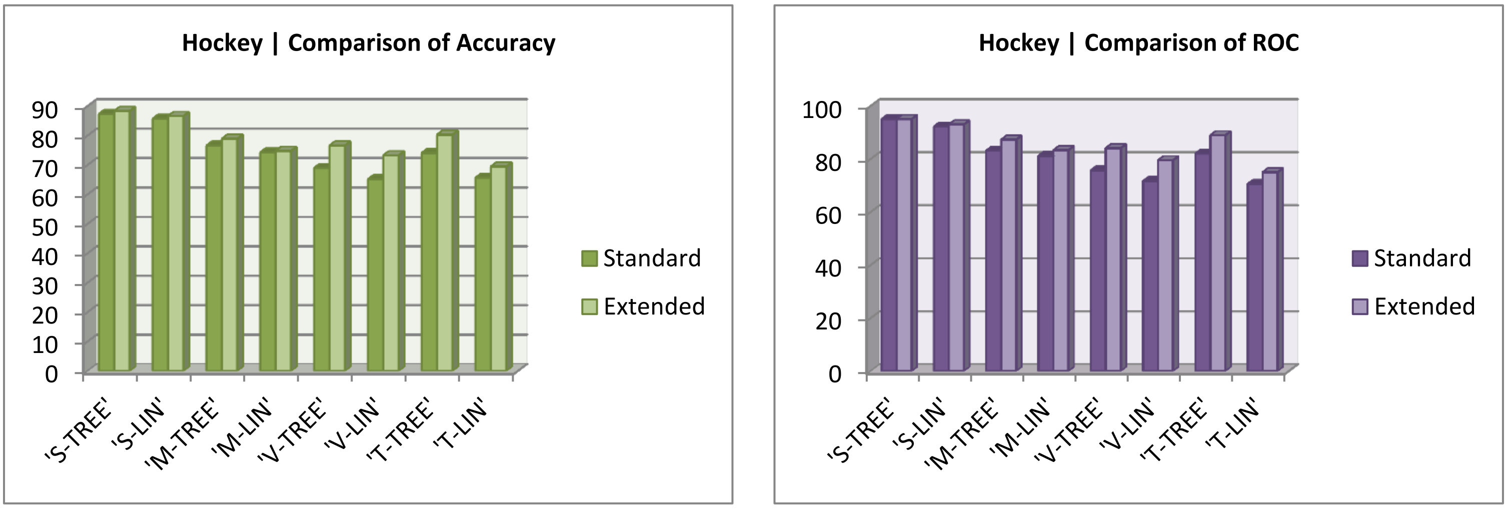


4.81% Increase on Average

6.86% Increase on Average

Figure 19: Violent Flows dataset results with and without method extension

All methods show an improvement in classification performance except STIP with a Linear SVM classifier which sees a 0.58% and 0.47% decrease in AUC and accuracy respectively. MBP sees substantial gains with 13.95% AUC improvement and 9.99% increase in general accuracy. STIP is the best method of classification for the Violent Flows dataset before and after the descriptor extension.

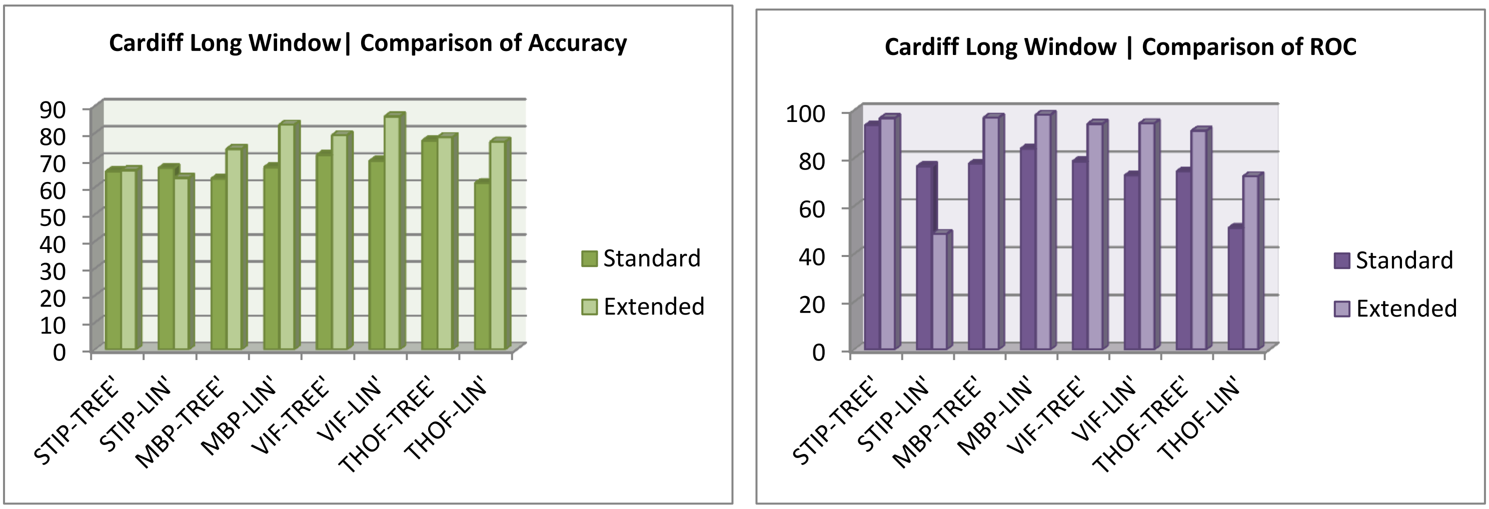


3.81% Increase on Average

4.34% Increase on Average

Figure 18: Hockey dataset results with and without method extension

The change in performance in entirely positive when applied to the hockey dataset. The lowest increase is seen when applying GEP extension to the STIP algorithm, an increase of >0.5% is shown, this is too small to justify the increased computation time required to generate both STIP and GEP. The lack of a justifiable increase in STIP performance probably stems from the fact that HOG provides powerful local description of visual elements seen in hockey data, and adding less descriptive measures wouldn't affect performance. Trajectory HOF sees the largest improvement with 8.13% and 7.88% increase in both AUC and accuracy. Although STIP sees little performance increase it remains the best method for hockey violence classification.

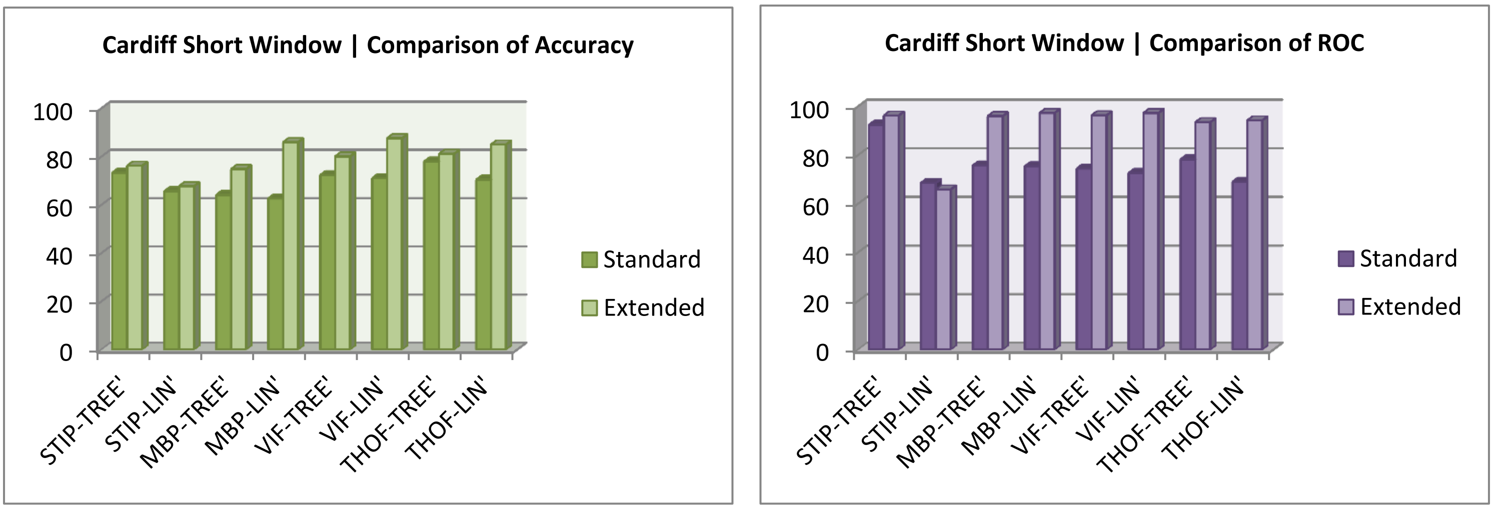


7.93% Increase on Average

10.53% Increase on Average

FIGURE 20: CARDIFF LONG WINDOW DATASET RESULTS WITH AND WITHOUT METHOD EXTENSION

STIP using a linear classifier shows a minor decrease in performance but a major fall in AUC value, this decrease in AUC indicates that the ability of the linear SVM to discriminate between the two classes has fallen so far that the accuracy achieved by the extended STIP with linear classifier was essentially by chance. All other methods show an impressive increase in performance. The feature extension allows Violent Flows overtake Trajectory HOF as the best overall action descriptor for the Cardiff dataset.



10.23% Increase on Average

16.31% Increase on Average

FIGURE 21: CARDIFF SHORT WINDOW DATASET RESULTS WITH AND WITHOUT METHOD EXTENSION

Short window Cardiff tests show a greater increase than its larger window test counterpart but once again STIP using a linear classifier is the only source of performance regression where the AUC has dropped 2.66%. MBP sees the largest accuracy gain with a 23.31% increase to its linear classification variant which elevates it to the best method of motion description for this particular test type.

Across all four datasets a noticeable increase in performance is seen when combining GEP features with each motion based descriptor method. The one exception to this is the STIP algorithm whose linear classifier variant sees a consistent decrease in performance. The most likely cause of this regression is that when the GEP features are combined with the STIP features a non-linear class divide is created which the linear SVM cannot be adequately define.

The correlation between changes in true positive and true negative classification rates when extension is applied are as follows:

	Hockey	VIF	Cardiff Long Window	Cardiff Short Window
Pearson's R	0.831407	0.565433	0.064552	0.004652

The results from the Hockey dataset show a high positive correlation meaning that the extended features increased classification rates for both classes by an almost equal amount.

The results from the Violent Flows dataset show a weaker positive correlation which means that at each test one class shows a greater rate of improvement over the other; looking at the data we can see that classification rates for non-violent scenes increases a greater amount than violent scenes (See Appendix G). Although non-violence detection sees an increase it is not enough to bring any method above the currently best true negative rate of 70.18% held by the GEP descriptor; this means that the status of identifying a method suitable for describing the Violent Flows data within a real life scenario remains unchanged.

Both Cardiff dataset results show extremely weak positive correlations indicating that both classes see classification improvements that are non-linear; based on the fact that true negatives had little room for improvement intuitively tells us that most performance increase happens to the violence class; this can be backed up by the data (See Appendix D E).

By taking the average rate of change in classification performance over all four datasets for each method we can see how extension effected results overall.

TABLE 12: AVERAGE CHANGE IN PERFORMANCE OVER ALL DATASETS USING EXTENDED DESCRIPTORS

	STIP		VIF		MBP		Trajectory HOF	
Classifier	Random	Linear	Random	Linear	Random	Linear	Random	Linear
Accuracy	1.35%	-0.31%	8.47%	**12.38%**	7.04%	12.27%	3.09%	9.07%
ROC AUC	2.32%	-7.65%	**14.21%**	13.25%	13.03%	16.17%	10.76%	14.01%

Based on the noticeable increase in classification rates across VIF, MBP and Trajectory HOF methods I can declare that I have fulfilled one of my set project goals by successfully proposing a way of extending pre-existing descriptor methods with the intent of producing improvements to classification performance.

Combinatorial Testing

The many different ways violence can occur cannot be easily described by one single method as indicated by the results of the four dataset tests. It may be the case that when two different descriptor methods are combined a more powerful descriptor is created whose performance not only increases, but also extends over to different types of data thus producing a rounded solution that fits multiple violence subtypes.

To evaluate whether or not merged descriptor methods offer greater classification accuracy over their standalone sub-parts, every possible combination using the five different descriptor methods and their subsets will be computed across all four datasets with both linear SVM and random forest variants for a total of 62 different tests per dataset. The results from both merged methods and the single method tests will be used to generate a Pareto frontier which dictates the objectively best results.

In all further tables S, V, G, and T represent STIP, VIF, GEP and Trajectory HOF respectively; for example, a method labelled "SVG" indicates that it is comprised of STIP, VIF and GEP descriptor methods concatenated together.

Table 13: Violent Flows dataset frontier including method combinations

Method	Classifier	True Positive	True Negative	Accuracy	AUC
G	Linear	70.80%	**70.18%**	70.55%	74.98
SVTG	RF	91.24%	51.27%	75.22%	82.72
SVGM	RF	**93.67%**	52.00%	76.97%	81.89
GVTM	RF	87.35%	59.64%	76.24%	**84.97**
GVT	RF	87.59%	60.00%	76.53%	84.23
SGM	RF	90.27%	51.64%	74.78%	82.09
GVM	RF	88.08%	60.36%	76.97%	81.83
GVM	Linear	82.97%	69.45%	**77.55%**	82.90
SG	RF	91.97%	53.45%	76.53%	81.16

The main problem with the Violent Flows dataset was that no single method was suited for real-life application due to the low true negative rate, unfortunately introducing combinatorial descriptors doesn't help at all; no descriptor combination is able to outperform the previously best true negative rate of 70.18% held by GEP. The combination descriptor GVM does hold the greatest overall performance of any method; it outperforms the highest performing single method that is STIP, by 2.19%.

TABLE 14: HOCKEY DATASET FRONTIER INCLUDING METHOD COMBINATIONS

Method	Classifier	True Positive	True Negative	Accuracy	AUC
SVG	RF	92.15%	84.37%	88.25%	94.88
SGT	RF	91.95%	84.37%	88.15%	95.03
SV	RF	**92.76%**	84.17%	**88.45%**	94.75
SV	Linear	85.92%	**89.58%**	87.75%	93.90
SG	RF	92.56%	84.17%	88.35%	**95.06**

STIP appears to exist within each of the five most dominant method combinations; this suggests that STIP features contain extremely high descriptive power when applied to the hockey dataset. STIP had previously shown to be the dominant method with 87.35% accuracy; combinatorial results do not show much improvement with the next best method offering an overall performance increase of 1.1%. The results from the non-combinatorial tests on the hockey data shows that no method is suitable for use as an observation aid as they do not meet the suitability criteria; the combination of STIP + VIF does however meet them.

TABLE 15: CARDIFF LONG WINDOW DATASET FRONTIER INCLUDING METHOD COMBINATIONS

Method	Classifier	True Positive	True Negative	Accuracy	AUC
G	RF	72.04%	96.55%	86.33%	**96.65**
G	Linear	**98.25%**	87.30%	**91.86%**	96.30
SVM	RF	21.05%	99.69%	66.91%	92.02
SVM	Linear	30.04%	99.76%	70.70%	89.26
SVG-	RF	26.86%	99.61%	69.29%	94.45
GVM	RF	39.36%	99.53%	74.45%	93.51
GVM	Linear	62.17%	97.96%	83.04%	95.28
SV	RF	22.26%	99.69%	67.41%	89.48
SV	Linear	31.36%	99.61%	71.16%	89.25
SM	Linear	21.93%	**100.00%**	67.46%	81.63
SG	RF	19.41%	99.69%	66.22%	86.74
MG	Linear	60.96%	98.90%	83.09%	88.15

The STIP+MBP combination achieves a perfect classification rate when identifying non-violent data unfortunately the violence detection is lacking greatly. GEP remains the best method for detecting violence in the large window Cardiff dataset as it holds the greatest overall accuracy against all method combinations.

One of the requirements needed for a method to be considered for real-life application use was that their true positive classification rate was greater than the define threshold of 50%; combinatorial testing has provided two of these, MG and GVM. These two methods also hold high true negative rates which make them adequate candidates for use as surveillance observation aid.

TABLE 16: CARDIFF SHORT WINDOW DATASET FRONTIER INCLUDING METHOD COMBINATIONS

Method	Classifier	True Positive	True Negative	Accuracy	AUC
G	RF	85.97%	97.01%	91.91%	97.31
G	Linear	**97.11%**	93.35%	**95.09%**	**97.58**
SVGM	Linear	45.83%	**98.54%**	74.18%	91.39
SGM	Linear	42.18%	**98.54%**	72.50%	92.58
GVM	Linear	64.12%	98.17%	82.44%	97.45
MG	Linear	71.60%	98.32%	85.97%	97.32

As with the previous Cardiff dataset test, GEP comes out on top with a near perfect overall classification rate. By eliminating all methods on the frontier that obtain a true positive rate that falls below the acceptable threshold we are left with only GEP and a two combination descriptors, MG and GVM. Each of these three methods obtains a high true negative rate that makes them suitable for application as an observation aid. It is not clear which one would perform the best without a study on acceptable false negatives.

TABLE 17: CROSS DATASET PERFORMANCE FRONTIER INCLUDING METHOD COMBINATIONS

Method	Classifier	True Positive	True Negative	Accuracy	AUC
SVTMG	RF	63.92%	82.99%	76.42%	90.66
G	RF	78.36%	84.96%	82.52%	88.50
G	Linear	**83.89%**	83.05%	**83.23%**	87.32
SVGM	Linear	61.12%	87.35%	76.54%	87.64
GVTM	RF	70.68%	81.30%	78.05%	89.45
SVM	Linear	58.72%	87.07%	75.37%	85.91
SVG	RF	63.98%	82.95%	76.53%	91.21
SVG	Linear	62.62%	87.02%	77.06%	86.59
GVT	RF	71.33%	80.70%	77.99%	88.51
GVM	Linear	71.63%	86.04%	80.24%	89.98
TGM	RF	71.66%	81.51%	78.28%	89.77
SV	Linear	60.44%	**87.51%**	76.35%	85.15
TG	RF	72.20%	81.57%	78.46%	89.20
SG	RF	63.84%	83.71%	76.83%	**92.30**

Before performing combinatorial testing the expectation was that the final average frontier would be comprised of only merged methods; as shown above this is not the case. GEP remains the best overall classification method across all datasets. The only dataset that shows GEP to not fall within the Pareto frontier was the hockey dataset; this implies that global visual descriptive nature of the measures held within GEP were not suited to sparse action scenes. STIP + Violent Flows holds the top spot for most suitable candidate for use in real life applications but doesn't quite reach the 90% true negative rate requirement.

CONCLUSIONS

The main goal of the project was to create a method of action recognition that is capable at detecting crowd characteristics that are indicative of city centre locations; the intended method of identifying classification methods fit for purpose was to take multiple modern action recognition and evaluate their performance on suitable datasets. The results from all testing has shown that on average, global visual descriptors were better suited for violence detection across a wide variety of different violence types than other motion based methods.

The average cross-set performance showed local motion features descriptors to outperform the global motion alternatives; the implication of this is that local descriptors are in fact better suited at describing densely populated scenes; this was the opposite of what was expected. The greatest cross-dataset classification performance achieved was 83.22%; this value is high enough for me to say that I have successfully developed a method for differentiating between different scene types depicting instances of violence and non-violence using computer vision techniques; this rate of classification was achieved by my own original descriptor design.

One of my goals was to evaluate the results from the first wave of testing and derive a suitable way of improving classification scores by building on the flaws of each method. By analysing both the worst obtained results and the composition of each dataset I managed to devise four different measures that when combined with the already tested motion descriptor provided a sizeable increase in overall performance; these measure went on to form an independent method referred to as GEP (GLCM + Edge cardinality + Pixel frame difference) as they held significant descriptive power as shown by the overall results. Before these measures became their own entity they were first applied to each of the other four methods, three of which showed to be very receptive to the new data; VIF, MBP and Trajectory HOF showed increases in performance of up to 16%.

The final phase of testing and evaluation was to determine whether or not merging different methods would result in a combination of descriptors that show greater performance than any method previously seen up to that point. The results from this were ultimately negative; no combination was able to surpass the standalone performance of GEP.

Within the document I established a scenario where obtaining the highest overall classification rate doesn't translate to being the best method for use as an assistive piece of software to aid human observers identify violence. At each stage of testing and for each dataset I have highlighted descriptor methods that hold adequate performance for use in the real world. I identified methods suitable for both Hockey and Cardiff datasets. To identify suitable methods I state that the non-violence detection rate is more important than any other measure. Although results from the Violent Flows dataset show a high violence classification rate, it achieves a true negative rate that is too low for real world use; the same can be said about overall performance across all three datasets.

FUTURE IMPROVEMENTS

All methods created for the purpose of action recognition hold a high computational cost; none of the methods proposed come close to running in real-time due to a combination of poorly optimized code and large amounts of numerically complex operations. In order to make any of these methods useful for real world applications a speed increase needs to be achieved.

I use a loose definition of suitability when deciding whether or not a classification method provides adequate results for use in assisting surveillance observation. Suitable methods require a violence and non-violence detection rate of >50% and >90% respectively. These values were picked by applying perspective to the quantity of false alerts that would be presented to an observer. The definition may not be fully representative of what is acceptable in real workplace environments; therefore this definition can become far more precise by undertaking a study of human attention when a person is subject to varying types of visual or audio cues.

LEARNING REFLECTION

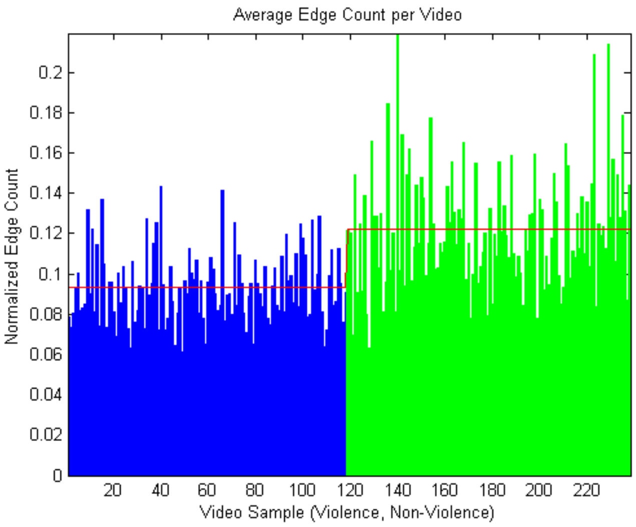
During development I constantly found myself implementing various algorithms required for scene description and method testing; typically each algorithm was quite long and would cover a wide range of operations. Towards the end of the project I found myself revisiting what I had written only to find chunks of completely incorrect logic; because of this I spent a lot of time fixing what should never have been broken in the first place. This problem arose because it can be quite difficult to notice mistakes when writing somewhat complex code without a set work structure; using document outlines and proper development plans I could have avoided this issue entirely, instead I found myself fixing errors and re-testing all my data; this is why RBF-SVM classifications are absent throughout most of the document, there just wasn't enough time to re-calculate them.

The difficulty in writing this document was vastly underestimated; at the start of the project I dedicated the last week and a half to completing this document. The problem with this was that methods I had completed earlier in the projects life-cycle were distant memories and required that I return to them to fully understand how they work so that I could provide adequate documentation; this took a lot more time than I expected. The amount of time I dedicated to writing this report was not enough and upon reading it back I find that the standard of work produced wasn't as high as I would have liked. In hindsight it is obvious that I should have been developing this document as the project progressed.

Overall I enjoyed this project immensely and the experience has made me realise that I would like to continue studying within the visual computer science topic area. After this project is submitted I aim to understand the limitations of the proposed GEP method outside of violence detection context.

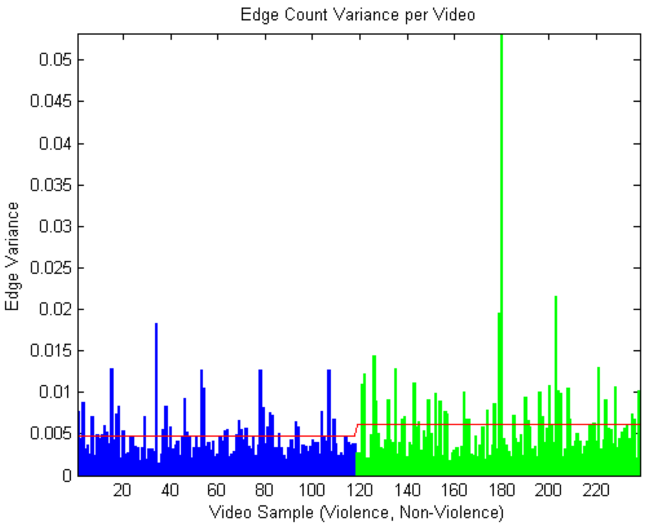
APPENDIX A

A.1



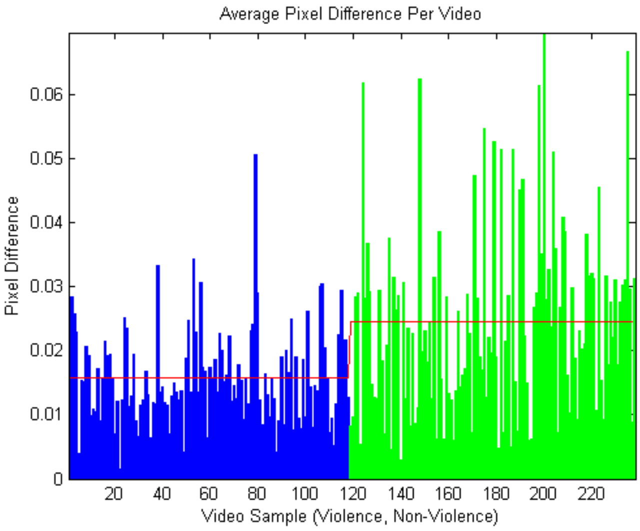
Test: Wilcoxon rank-sum
Difference Type: Significant
P-Value: 1.0621e-016

A.2



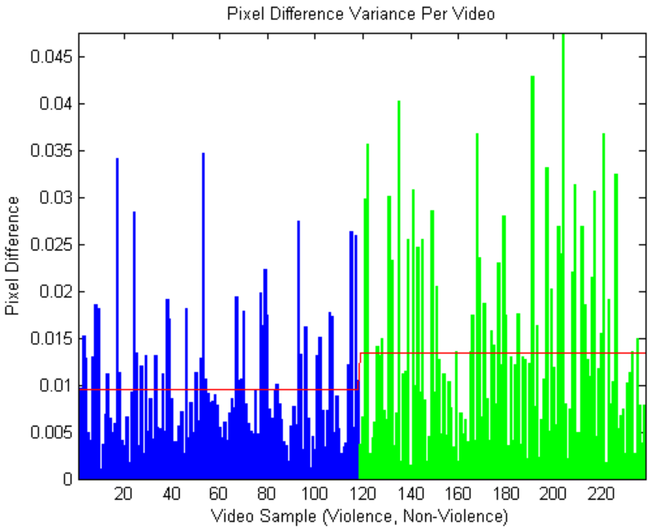
Test: Wilcoxon rank-sum
Difference Type: Significant
P-Value: 0.0209

A.3



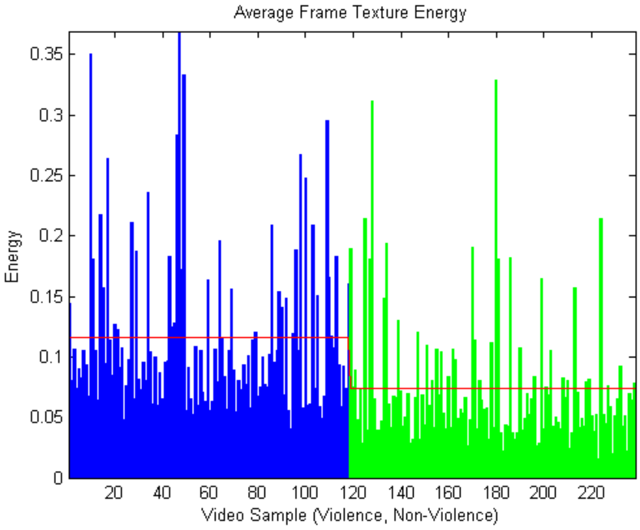
Test: Wilcoxon rank-sum
Difference Type: Significant
P-Value: 2.0726e-006

A.4



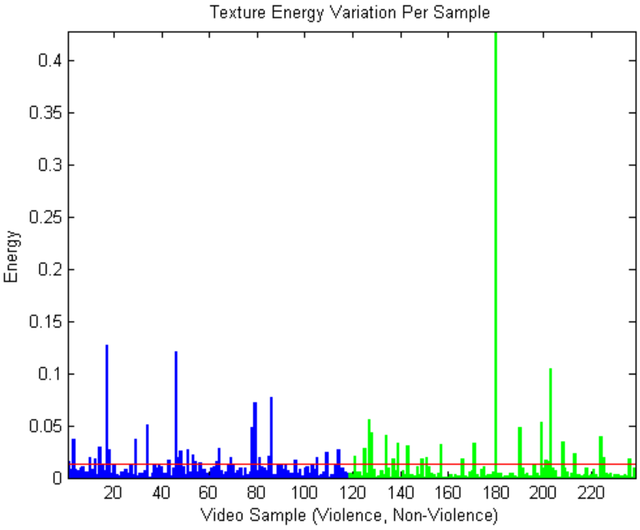
Test: Wilcoxon rank-sum
Difference Type: Significant
P-Value: 0.0087

A.5



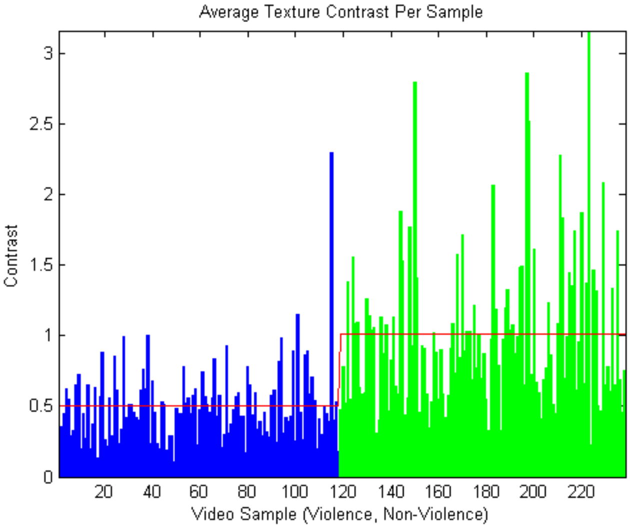
Test: Wilcoxon rank-sum
Difference Type: Significant
P-Value: 3.2535e-012

A.6



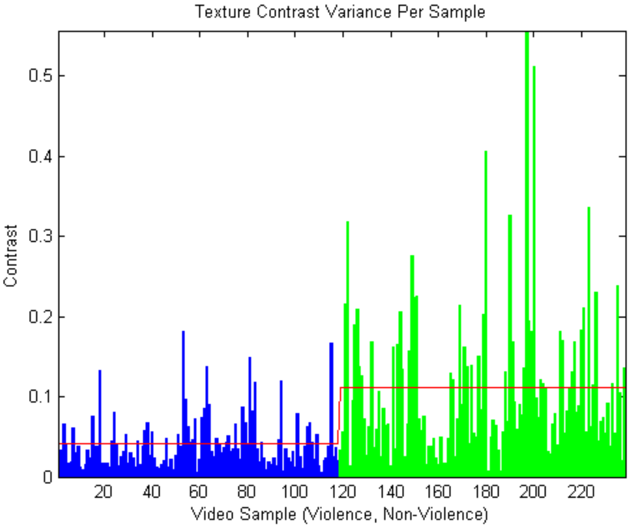
Test: Wilcoxon rank-sum
Difference Type: Not Significant
P-Value: 1.0560e-005

A.7



Test: Wilcoxon rank-sum
Difference Type: Significant
P-Value: 1.4684e-019

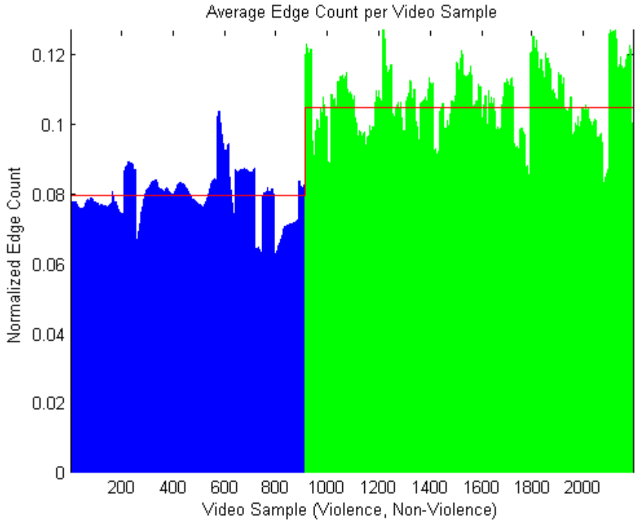
A.8



Test: Wilcoxon rank-sum
Difference Type: Significant
P-Value: 1.7389e-015

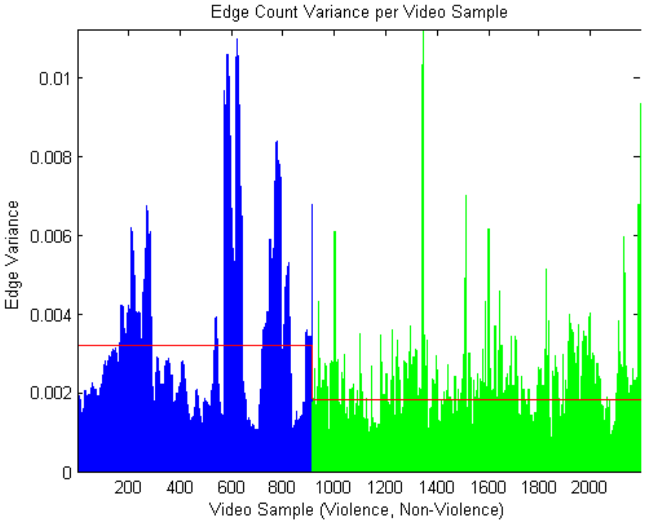
APPENDIX B

B.1



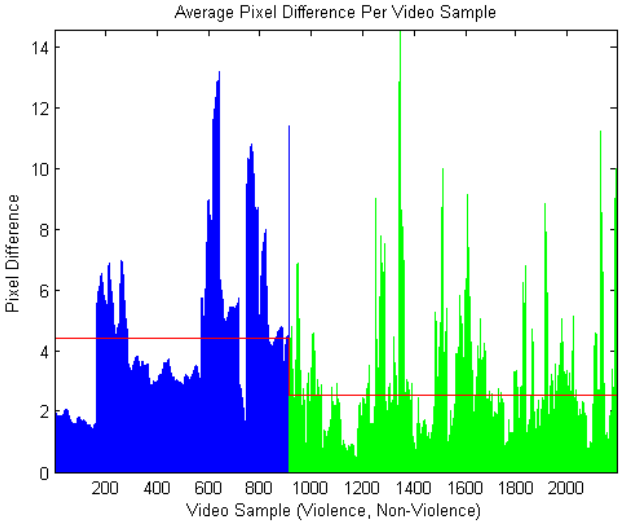
Test: Wilcoxon rank-sum
Difference Type: Significant
P-Value: 1.7774e-080

B.2



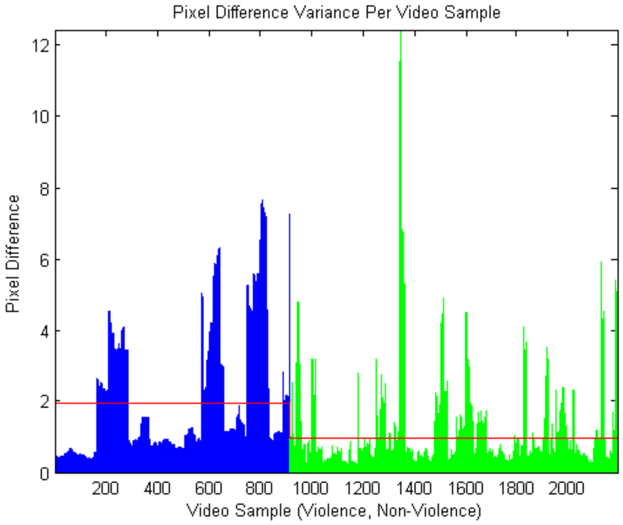
Test: Wilcoxon rank-sum
Difference Type: Significant
P-Value: 0

B.3



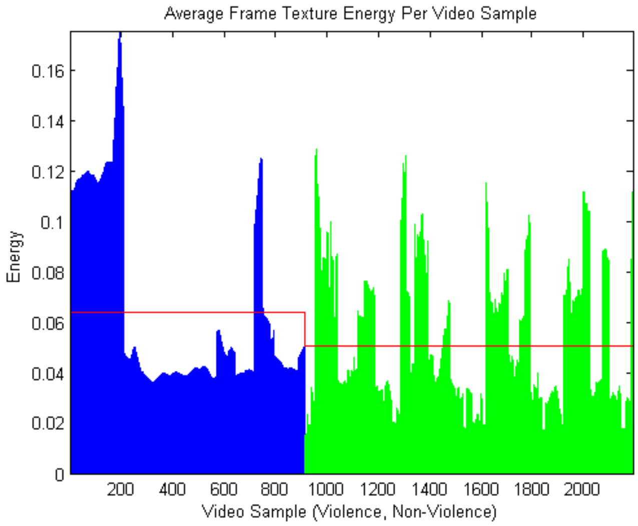
Test: Wilcoxon rank-sum
Difference Type: Significant
P-Value: 1.9132e-104

B.4



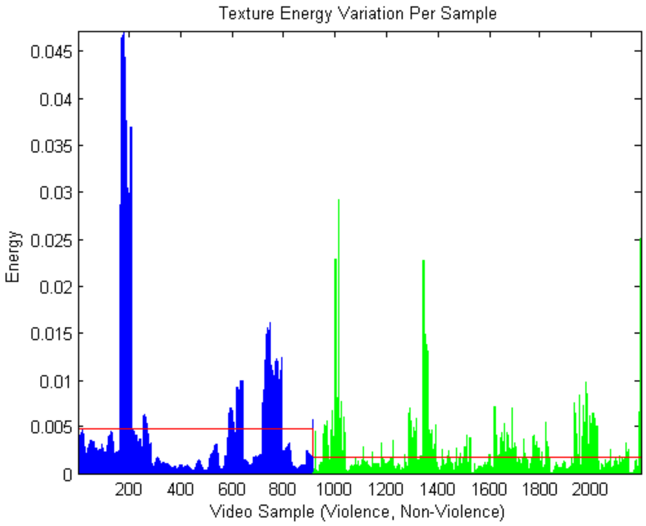
Test: Wilcoxon rank-sum
Difference Type: Significant
P-Value: 2.3587e-089

B.5



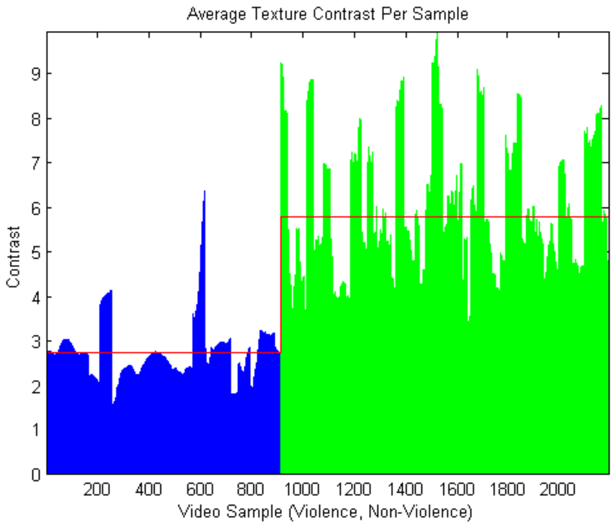
Test: Wilcoxon rank-sum
Difference Type: Significant
P-Value: 6.4266e-035

B.6



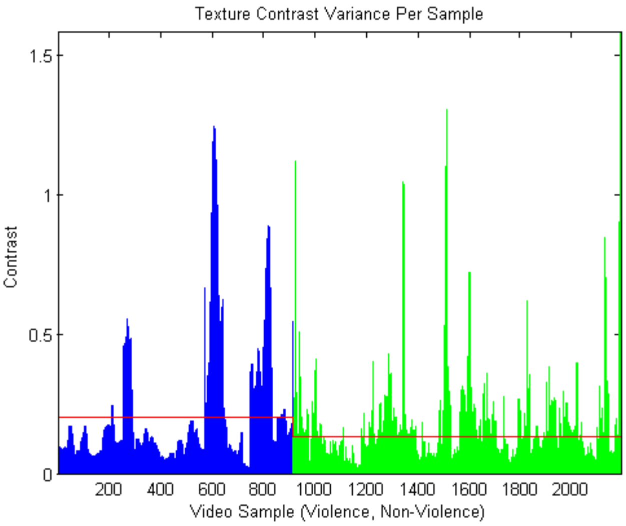
Test: Wilcoxon rank-sum
Difference Type: Significant
P-Value: 2.2338e-068

B.7



Test: Wilcoxon rank-sum
Difference Type: Significant
P-Value: 2.0751e-321

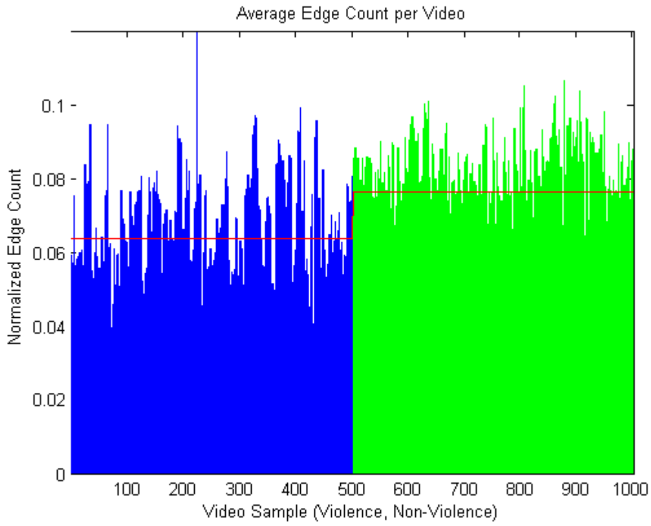
B.8



Test: Wilcoxon rank-sum
Difference Type: Significant
P-Value: 2.5987e-018

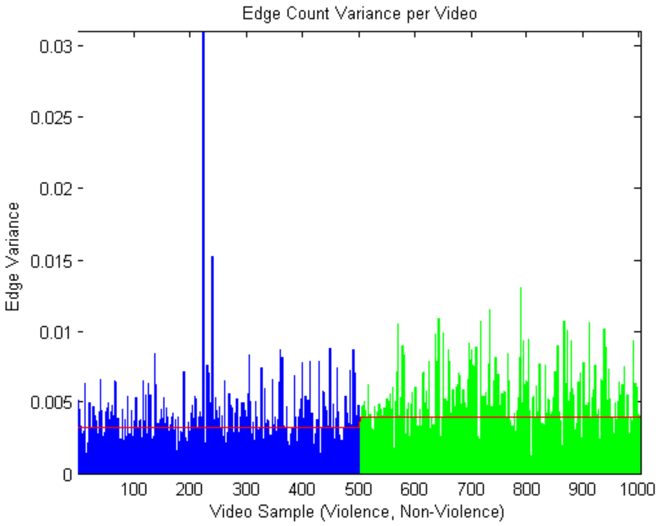
APPENDIX C

C.1



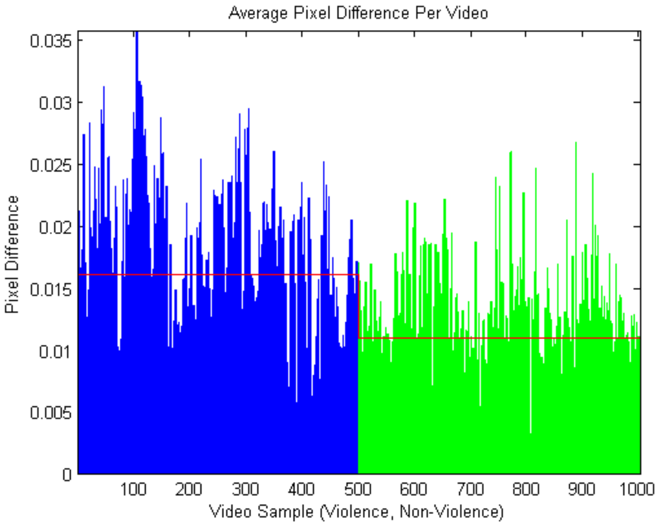
Test: Wilcoxon rank-sum
Difference Type: Significant
P-Value: : 8.6291e-048

C.2



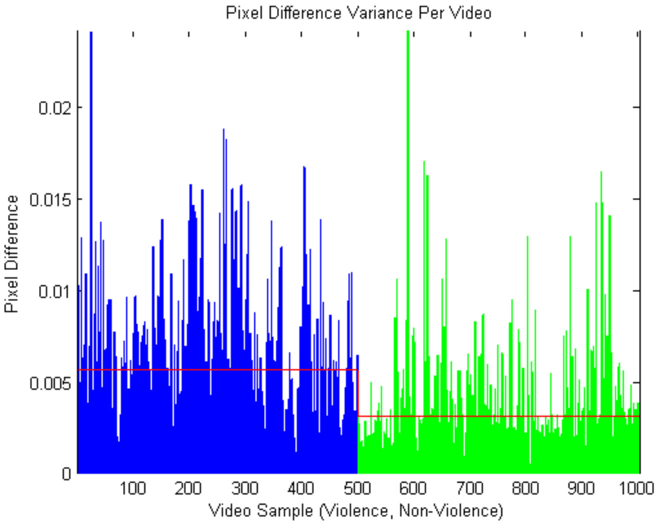
Test: Wilcoxon rank-sum
Difference Type: Significant
P-Value: 5.6799e-012

C.3



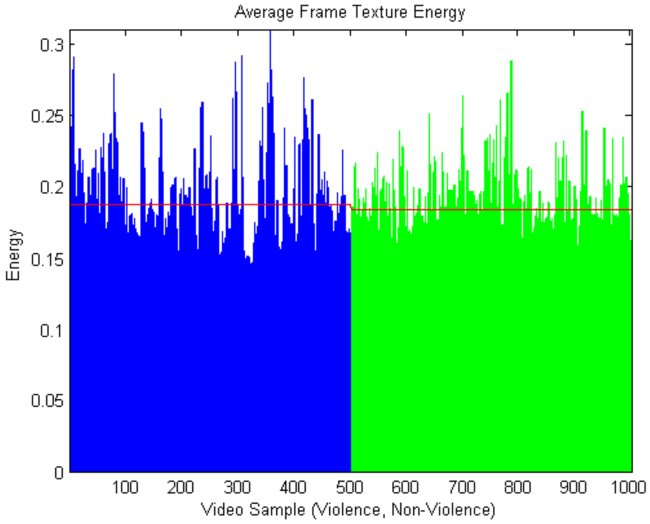
Test: Wilcoxon rank-sum
Difference Type: Significant
P-Value: 5.9792e-045

C.4



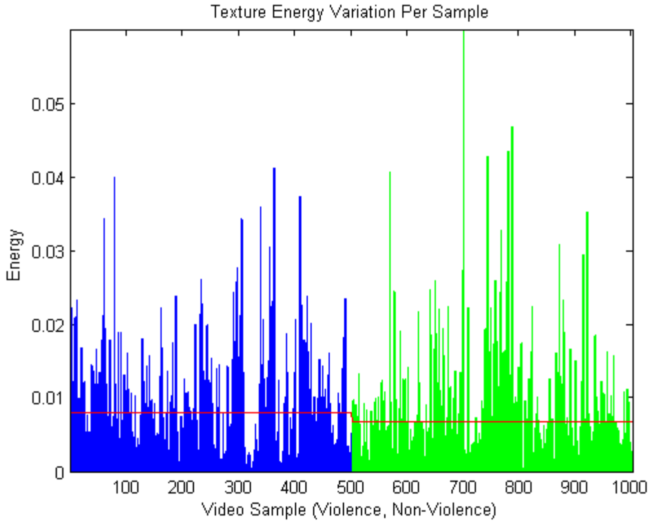
Test: Wilcoxon rank-sum
Difference Type: Significant
P-Value: 3.2461e-045

C.5



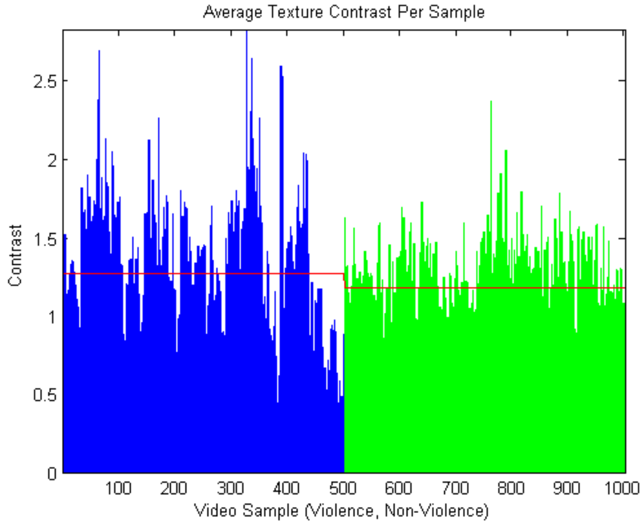
Test: Wilcoxon rank-sum
Difference Type: Not Significant
P-Value: 0.9393

C.6



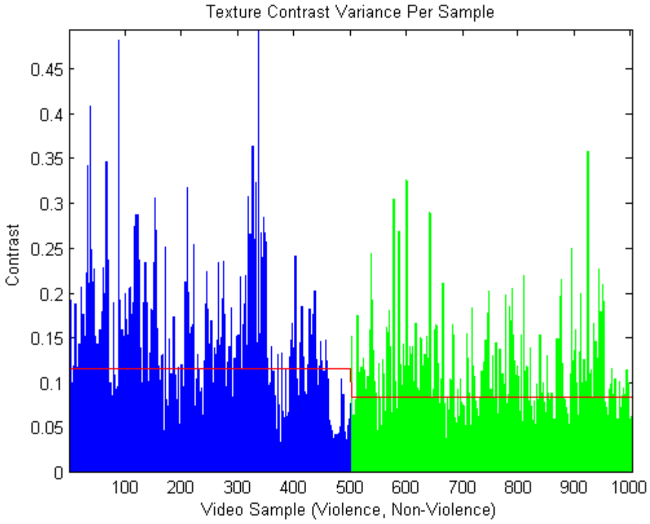
Test: Wilcoxon rank-sum
Difference Type: Significant
P-Value: 5.0064e-006

C.7



Test: Wilcoxon rank-sum
Difference Type: Significant
P-Value: 2.7117e-005

C.8



Test: Wilcoxon rank-sum
Difference Type: Not Significant
P-Value: 5.6692e-016

APPENDIX D

CARDIFF SHORT WINDOW DATASET RESULTS

Method	True Positive	True Negative	Specificity	Sensitivity	Accuracy	AUC
'SVTMG-TREE'	47.36%	97.30%	93.77%	68.27%	74.22%	93.49
'SVTMG-LIN'	52.98%	96.93%	93.68%	70.59%	76.62%	87.01
'G-TREE'	85.97%	97.01%	96.10%	88.95%	91.91%	97.31
'G-LIN'	97.11%	93.35%	92.62%	97.41%	95.09%	97.58
'S-TREE'	46.51%	96.42%	91.78%	67.73%	73.36%	92.42
'S-LIN'	30.10%	96.20%	87.19%	61.57%	65.66%	68.52
'M-TREE'	28.66%	94.67%	82.20%	60.70%	64.17%	75.73
'M-LIN'	22.87%	96.86%	86.22%	59.38%	62.67%	75.38
'V-TREE'	56.29%	86.12%	77.70%	69.64%	72.34%	74.43
'V-LIN'	52.98%	86.63%	77.30%	68.20%	71.08%	72.63
'T-TREE'	71.43%	84.09%	83.58%	72.19%	77.36%	87.99
'T-LIN'	61.39%	79.17%	76.97%	64.39%	69.72%	78.19
'SVTM-TREE'	43.79%	96.13%	90.67%	66.57%	71.94%	88.44
'SVTM-LIN'	47.36%	95.91%	90.86%	67.96%	73.48%	77.59
'SVTG-TREE'	46.34%	96.71%	92.37%	67.72%	73.44%	93.52
'SVTG-LIN'	52.89%	96.86%	93.53%	70.53%	76.54%	87.03
'SVGM-TREE'	42.69%	96.93%	92.28%	66.32%	71.87%	94.49
'SVGM-LIN'	45.83%	98.54%	96.42%	67.93%	74.18%	91.39
'SGTM-TREE'	47.70%	96.42%	91.97%	68.22%	73.91%	93.58
'SGTM-LIN'	54.08%	96.64%	93.26%	71.01%	76.97%	89.04
'GVTM-TREE'	61.48%	93.13%	88.49%	73.78%	78.51%	91.64
'GVTM-LIN'	60.29%	95.47%	91.96%	73.68%	79.21%	90.08
'SVT-TREE'	45.41%	95.76%	90.20%	67.13%	72.50%	88.55
'SVT-LIN'	47.45%	95.40%	89.86%	67.88%	73.24%	75.46
'SVM-TREE'	38.10%	96.64%	90.69%	64.51%	69.59%	89.19
'SVM-LIN'	39.12%	98.54%	95.83%	65.33%	71.08%	83.32
'STM-TREE'	45.07%	95.11%	88.78%	66.84%	71.98%	90.41
'STM-LIN'	49.32%	95.76%	90.91%	68.75%	74.30%	79.33
'VTM-TREE'	57.91%	91.09%	84.81%	71.58%	75.76%	81.80
'VTM-LIN'	44.98%	89.55%	78.72%	65.46%	68.96%	65.58
'SVG-TREE'	44.22%	97.30%	93.36%	67.00%	72.77%	94.57
'SVG-LIN'	51.62%	97.81%	95.29%	70.18%	76.46%	87.75
'SGT-TREE'	48.72%	96.27%	91.83%	68.61%	74.30%	94.43
'SGT-LIN'	54.68%	96.64%	93.32%	71.28%	77.25%	89.68
'GVT-TREE'	63.86%	92.55%	88.04%	74.88%	79.29%	92.84
'GVT-LIN'	59.01%	95.40%	91.68%	73.04%	78.59%	90.33
'SGM-TREE'	48.47%	97.59%	94.53%	68.80%	74.89%	95.56
'SGM-LIN'	42.18%	98.54%	96.12%	66.49%	72.50%	92.58
'GVM-TREE'	53.49%	97.30%	94.44%	70.89%	77.05%	93.97
'GVM-LIN'	64.12%	98.17%	96.79%	76.10%	82.44%	97.45
'TGM-TREE'	67.77%	91.89%	87.78%	76.85%	80.75%	93.65

'TGM-LIN'	63.52%	95.98%	93.14%	75.39%	80.98%	92.47
'SV-TREE'	42.77%	97.30%	93.15%	66.43%	72.10%	89.27
'SV-LIN'	43.71%	97.95%	94.83%	66.95%	72.89%	79.74
'ST-TREE'	46.09%	95.40%	89.59%	67.32%	72.61%	90.30
'ST-LIN'	48.98%	95.84%	91.00%	68.62%	74.18%	80.08
'SM-TREE'	39.80%	97.01%	91.94%	65.23%	70.57%	91.30
'SM-LIN'	34.35%	98.32%	94.61%	63.55%	68.76%	83.62
'VT-TREE'	61.82%	89.04%	82.90%	73.08%	76.46%	80.58
'VT-LIN'	45.15%	86.19%	73.75%	64.66%	67.23%	64.68
'VM-TREE'	45.07%	93.21%	85.07%	66.39%	70.96%	75.36
'VM-LIN'	37.07%	98.54%	95.61%	64.58%	70.14%	73.35
'TM-TREE'	60.20%	90.50%	84.49%	72.58%	76.50%	82.30
'TM-LIN'	46.77%	81.88%	68.92%	64.17%	65.66%	64.86
'TG-TREE'	69.05%	91.53%	87.50%	77.49%	81.14%	93.51
'TG-LIN'	72.36%	95.91%	93.83%	80.16%	85.03%	94.26
'VG-TREE'	60.71%	96.71%	94.07%	74.13%	80.08%	96.32
'VG-LIN'	77.30%	96.49%	94.98%	83.19%	87.62%	97.30
'SG-TREE'	51.45%	97.52%	94.68%	70.04%	76.23%	96.26
'SG-LIN'	36.14%	94.96%	86.03%	63.38%	67.78%	65.85
'MG-TREE'	48.64%	97.37%	94.08%	68.82%	74.85%	95.99
'MG-LIN'	71.60%	98.32%	97.34%	80.12%	85.97%	97.32

APPENDIX E

CARDIFF LARGE WINDOW DATASET RESULTS

Method	True Positive	True Negative	Specificity	Sensitivity	Accuracy	AUC
'SVTMG-TREE'	26.21%	99.45%	97.15%	65.35%	68.92%	93.56
'SVTMG-LIN'	36.84%	99.06%	96.55%	68.70%	73.13%	68.62
'G-TREE'	72.04%	96.55%	93.72%	82.85%	86.33%	96.65
'G-LIN'	98.25%	87.30%	84.69%	98.58%	91.86%	96.30
'S-TREE'	18.75%	99.53%	96.61%	63.15%	65.86%	93.71
'S-LIN'	21.27%	99.84%	98.98%	63.96%	67.09%	76.64
'M-TREE'	12.94%	99.06%	90.77%	61.42%	63.16%	77.72
'M-LIN'	22.26%	99.69%	98.07%	64.21%	67.41%	84.01
'V-TREE'	42.21%	93.18%	81.57%	69.29%	71.94%	78.73
'V-LIN'	35.20%	94.44%	81.89%	67.09%	69.74%	72.79
'T-TREE'	52.85%	94.67%	87.64%	73.75%	77.24%	74.46
'T-LIN'	31.47%	82.99%	56.94%	62.89%	61.52%	50.87
'SVTM-TREE'	23.68%	99.53%	97.30%	64.60%	67.92%	91.98
'SVTM-LIN'	36.18%	99.22%	97.06%	68.51%	72.94%	68.39
'SVTG-TREE'	23.46%	99.06%	94.69%	64.42%	67.55%	93.79
'SVTG-LIN'	36.84%	99.06%	96.55%	68.70%	73.13%	68.84
'SVGM-TREE'	23.57%	99.53%	97.29%	64.57%	67.87%	94.15
'SVGM-LIN'	32.79%	99.22%	96.76%	67.38%	71.53%	87.56
'SGTM-TREE'	18.53%	99.53%	96.57%	63.09%	65.77%	94.22
'SGTM-TREE'	18.53%	99.53%	96.57%	63.09%	65.77%	94.22
'SGTM-LIN'	38.27%	99.14%	96.94%	69.20%	73.77%	72.96
'GVTM-TREE'	46.16%	97.88%	93.97%	71.78%	76.33%	91.97
'GVTM-LIN'	38.27%	97.49%	91.60%	68.84%	72.81%	63.28
'SVT-TREE'	21.82%	99.45%	96.60%	64.03%	67.09%	91.12
'SVT-LIN'	36.18%	99.22%	97.06%	68.51%	72.94%	68.42
'SVM-TREE'	21.05%	99.69%	97.96%	63.86%	66.91%	92.02
'STM-TREE'	23.79%	99.37%	96.44%	64.60%	67.87%	91.71
'STM-LIN'	38.05%	99.06%	96.66%	69.11%	73.63%	72.40
'VTM-TREE'	41.78%	96.08%	88.40%	69.78%	73.45%	83.49
'VTM-LIN'	30.59%	94.83%	80.87%	65.65%	68.05%	57.64
'SVG-TREE'	26.86%	99.61%	98.00%	65.58%	69.29%	94.45
'SVG-LIN'	33.44%	99.14%	96.52%	67.57%	71.76%	87.36
'SGT-TREE'	17.76%	99.69%	97.59%	62.91%	65.54%	93.02
'SGT-LIN'	38.49%	99.14%	96.96%	69.28%	73.86%	73.12
'GVT-TREE'	48.14%	97.49%	93.21%	72.45%	76.92%	88.82
'GVT-LIN'	38.82%	97.41%	91.47%	69.02%	72.99%	63.81
'SGM-TREE'	16.67%	99.69%	97.44%	62.60%	65.08%	94.38
'SGM-LIN'	19.52%	97.81%	86.41%	62.97%	65.17%	61.52
'GVM-TREE'	39.36%	99.53%	98.36%	69.67%	74.45%	93.51

'GVM-LIN'	62.17%	97.96%	95.62%	78.37%	83.04%	95.28
'TGM-TREE'	51.32%	97.18%	92.86%	73.63%	78.06%	93.27
'TGM-LIN'	45.07%	97.18%	91.95%	71.22%	75.46%	69.94
'SV-TREE'	22.26%	99.69%	98.07%	64.21%	67.41%	89.48
'SV-LIN'	31.36%	99.61%	98.28%	67.00%	71.16%	89.25
'ST-TREE'	18.86%	99.45%	96.09%	63.17%	65.86%	91.58
'ST-LIN'	38.05%	99.06%	96.66%	69.11%	73.63%	72.58
'SM-TREE'	16.78%	99.69%	97.45%	62.63%	65.13%	92.05
'SM-LIN'	21.93%	100.00%	100.00%	64.19%	67.46%	81.63
'VT-TREE'	45.61%	94.83%	86.31%	70.93%	74.31%	79.30
'VT-LIN'	30.92%	93.65%	77.69%	65.48%	67.50%	57.39
'VM-TREE'	32.35%	98.43%	93.65%	67.06%	70.89%	81.38
'VM-LIN'	30.26%	99.61%	98.22%	66.65%	70.70%	74.00
'TM-TREE'	46.60%	95.22%	87.45%	71.39%	74.95%	85.11
'TM-LIN'	29.17%	95.06%	80.85%	65.25%	67.60%	56.01

APPENDIX F

HOCKEY DATA DATASET RESULTS

Method	True Positive	True Negative	Specificity	Sensitivity	Accuracy	AUC
'SVTMG-TREE'	91.35%	83.57%	84.70%	90.65%	87.45%	94.84
'SVTMG-LIN'	84.51%	84.57%	84.51%	84.57%	84.54%	90.84
'G-TREE'	76.86%	77.56%	77.33%	77.09%	77.21%	83.45
'G-LIN'	69.42%	81.36%	78.77%	72.76%	75.40%	80.42
'S-TREE'	92.35%	82.36%	83.91%	91.54%	87.35%	95.04
'S-LIN'	84.10%	87.37%	86.90%	84.66%	85.74%	92.24
'M-TREE'	78.47%	74.75%	75.58%	77.71%	76.61%	83.13
'M-LIN'	72.84%	75.75%	74.95%	73.68%	74.30%	81.22
'V-TREE'	75.86%	62.12%	66.61%	72.09%	68.98%	75.72
'V-LIN'	67.00%	63.33%	64.53%	65.83%	65.16%	71.68
'T-TREE'	77.46%	70.74%	72.50%	75.91%	74.10%	82.03
'T-LIN'	66.80%	64.53%	65.23%	66.12%	65.66%	70.57
'SVTM-TREE'	92.15%	83.77%	84.97%	91.47%	87.95%	94.86
'SVTM-LIN'	82.90%	82.77%	82.73%	82.93%	82.83%	89.51
'SVTG-TREE'	92.15%	82.97%	84.35%	91.39%	87.55%	94.89
'SVTG-LIN'	84.51%	84.57%	84.51%	84.57%	84.54%	90.94
'SVGM-TREE'	91.55%	83.17%	84.42%	90.81%	87.35%	94.98
'SVGM-LIN'	85.31%	88.38%	87.97%	85.80%	86.85%	93.57
'SGTM-TREE'	91.75%	83.57%	84.76%	91.05%	87.65%	94.95
'SGTM-LIN'	84.71%	83.37%	83.53%	84.55%	84.04%	90.61
'GVTM-TREE'	87.73%	74.55%	77.44%	85.91%	81.12%	89.23
'GVTM-LIN'	75.45%	66.73%	69.32%	73.19%	71.08%	76.78
'SVT-TREE'	92.15%	83.37%	84.66%	91.43%	87.75%	94.53
'SVT-LIN'	84.71%	83.37%	83.53%	84.55%	84.04%	90.19
'SVM-TREE'	91.75%	82.57%	83.98%	90.95%	87.15%	94.41
'SVM-LIN'	84.71%	87.78%	87.34%	85.21%	86.24%	93.12
'STM-TREE'	91.35%	82.16%	83.61%	90.51%	86.75%	94.65
'STM-LIN'	83.70%	81.76%	82.05%	83.44%	82.73%	89.74
'VTM-TREE'	83.70%	71.94%	74.82%	81.59%	77.81%	84.18
'VTM-LIN'	68.41%	68.14%	68.14%	68.41%	68.27%	72.35
'SVG-TREE'	92.15%	84.37%	85.45%	91.52%	88.25%	94.88
'SVG-LIN'	85.11%	88.58%	88.13%	85.66%	86.85%	93.55
'SGT-TREE'	91.95%	84.37%	85.42%	91.32%	88.15%	95.03
'SGT-LIN'	84.10%	83.57%	83.60%	84.07%	83.84%	90.61
'GVT-TREE'	85.71%	72.75%	75.80%	83.64%	79.22%	88.14
'GVT-LIN'	71.63%	69.94%	70.36%	71.22%	70.78%	76.05
'SGM-TREE'	91.95%	83.77%	84.94%	91.27%	87.85%	94.83
'SGM-LIN'	84.91%	88.58%	88.10%	85.49%	86.75%	93.26
'GVM-TREE'	83.70%	73.15%	75.64%	81.84%	78.41%	85.57

'GVM-LIN'	77.26%	78.56%	78.21%	77.62%	77.91%	84.28
'TGM-TREE'	86.52%	76.95%	78.90%	85.14%	81.73%	89.70
'TGM-LIN'	71.83%	69.34%	70.00%	71.19%	70.58%	75.82
'SV-TREE'	92.76%	84.17%	85.37%	92.11%	88.45%	94.75
'SV-LIN'	85.92%	89.58%	89.14%	86.46%	87.75%	93.90
'ST-TREE'	92.15%	83.37%	84.66%	91.43%	87.75%	94.73
'ST-LIN'	83.30%	82.36%	82.47%	83.20%	82.83%	89.72
'SM-TREE'	92.15%	82.36%	83.88%	91.33%	87.25%	94.59
'SM-LIN'	83.70%	87.37%	86.85%	84.33%	85.54%	92.16
'VT-TREE'	82.09%	68.54%	72.21%	79.35%	75.30%	82.37
'VT-LIN'	69.82%	67.74%	68.31%	69.26%	68.78%	72.96
'VM-TREE'	75.86%	65.73%	68.80%	73.21%	70.78%	78.45
'VM-LIN'	72.84%	71.94%	72.11%	72.67%	72.39%	79.71
'TM-TREE'	78.47%	73.15%	74.43%	77.33%	75.80%	84.55
'TM-LIN'	68.61%	64.53%	65.83%	67.36%	66.57%	71.70
'TG-TREE'	85.51%	75.15%	77.41%	83.89%	80.32%	88.95
'TG-LIN'	71.83%	67.13%	68.52%	70.53%	69.48%	74.99
'VG-TREE'	83.70%	69.54%	73.24%	81.07%	76.61%	84.12
'VG-LIN'	74.25%	72.34%	72.78%	73.82%	73.29%	79.54
'SG-TREE'	92.56%	84.17%	85.34%	91.90%	88.35%	95.06
'SG-LIN'	84.31%	88.98%	88.40%	85.06%	86.65%	93.05
'MG-TREE'	80.68%	77.15%	77.86%	80.04%	78.92%	87.30
'MG-LIN'	72.84%	76.75%	75.73%	73.94%	74.80%	83.35

APPENDIX G

VIOLENT FLOWS DATASET RESULTS

Method	True Positive	True Negative	Specificity	Sensitivity	Accuracy	AUC
'SVTMG-TREE'	90.75%	51.64%	73.72%	78.89%	75.07%	80.74
'SVTMG-LIN'	83.45%	58.55%	75.05%	70.31%	73.47%	77.81
'G-TREE'	78.59%	68.73%	78.97%	68.23%	74.64%	76.58
'G-LIN'	70.80%	70.18%	78.02%	61.66%	70.55%	74.98
'S-TREE'	91.97%	50.55%	73.54%	80.81%	75.36%	78.75
'S-LIN'	80.05%	63.27%	76.51%	67.97%	73.32%	76.99
'M-TREE'	78.10%	38.91%	65.64%	54.31%	62.39%	63.38
'M-LIN'	66.18%	52.73%	67.66%	51.06%	60.79%	63.86
'V-TREE'	84.67%	49.82%	71.60%	68.50%	70.70%	75.73
'V-LIN'	75.91%	53.09%	70.75%	59.59%	66.76%	71.24
'T-TREE'	80.29%	60.00%	75.00%	67.07%	72.16%	79.03
'T-LIN'	74.21%	44.36%	66.59%	53.51%	62.24%	60.73
'SVTM-TREE'	91.48%	50.55%	73.44%	79.89%	75.07%	80.94
'SVTM-LIN'	84.43%	57.82%	74.95%	71.30%	73.76%	75.44
'SVTG-TREE'	91.24%	51.27%	73.67%	79.66%	75.22%	82.72
'SVTG-LIN'	83.45%	58.55%	75.05%	70.31%	73.47%	77.76
'SVGM-TREE'	93.67%	52.00%	74.47%	84.62%	76.97%	81.89
'SVGM-LIN'	80.54%	63.27%	76.62%	68.50%	73.62%	78.03
'SGTM-TREE'	91.48%	48.73%	72.73%	79.29%	74.34%	81.18
'SGTM-LIN'	83.21%	57.82%	74.67%	69.74%	73.03%	77.70
'GVTM-TREE'	87.35%	59.64%	76.38%	75.93%	76.24%	84.97
'GVTM-LIN'	75.67%	50.91%	69.73%	58.33%	65.74%	66.94
'SVT-TREE'	92.94%	48.00%	72.76%	81.99%	74.93%	81.05
'SVT-LIN'	84.43%	57.82%	74.95%	71.30%	73.76%	75.46
'SVM-TREE'	91.73%	48.73%	72.78%	79.76%	74.49%	79.20
'SVM-LIN'	81.02%	62.18%	76.20%	68.67%	73.47%	77.94
'STM-TREE'	91.48%	50.18%	73.29%	79.77%	74.93%	80.34
'STM-LIN'	84.43%	57.09%	74.62%	71.04%	73.47%	75.33
'VTM-TREE'	83.94%	56.36%	74.19%	70.14%	72.89%	81.26
'VTM-LIN'	74.21%	45.45%	67.03%	54.11%	62.68%	62.16
'SVG-TREE'	92.70%	50.55%	73.69%	82.25%	75.80%	80.94
'SVG-LIN'	80.29%	62.55%	76.21%	67.98%	73.18%	77.69
'SGT-TREE'	91.97%	51.27%	73.83%	81.03%	75.66%	81.83
'SGT-LIN'	83.45%	58.18%	74.89%	70.18%	73.32%	77.73
'GVT-TREE'	87.59%	60.00%	76.60%	76.39%	76.53%	84.23
'GVT-LIN'	75.43%	50.18%	69.35%	57.74%	65.31%	66.62
'SGM-TREE'	90.27%	51.64%	73.61%	78.02%	74.78%	82.09
'SGM-LIN'	79.81%	62.91%	76.28%	67.58%	73.03%	77.06
'GVM-TREE'	88.08%	60.36%	76.86%	77.21%	76.97%	81.83

'GVM-LIN'	82.97%	69.45%	80.24%	73.18%	77.55%	82.90
'TGM-TREE'	81.02%	60.00%	75.17%	67.90%	72.59%	82.48
'TGM-LIN'	75.18%	51.27%	69.75%	58.02%	65.60%	66.44
'SV-TREE'	92.21%	48.00%	72.61%	80.49%	74.49%	80.55
'SV-LIN'	80.78%	62.91%	76.50%	68.65%	73.62%	77.70
'ST-TREE'	91.48%	48.36%	72.59%	79.17%	74.20%	79.40
'ST-LIN'	84.43%	57.09%	74.62%	71.04%	73.47%	75.34
'SM-TREE'	90.75%	48.36%	72.43%	77.78%	73.76%	79.46
'SM-LIN'	80.05%	64.00%	76.87%	68.22%	73.62%	77.42
'VT-TREE'	83.94%	56.00%	74.03%	70.00%	72.74%	81.46
'VT-LIN'	74.94%	42.18%	65.95%	52.97%	61.81%	61.24
'VM-TREE'	82.24%	54.91%	73.16%	67.41%	71.28%	75.27
'VM-LIN'	73.97%	58.18%	72.55%	59.93%	67.64%	72.90
'TM-TREE'	79.56%	60.00%	74.83%	66.27%	71.72%	79.92
'TM-LIN'	73.48%	44.73%	66.52%	53.02%	61.95%	61.14
'TG-TREE'	81.27%	62.91%	76.61%	69.20%	73.91%	82.84
'TG-LIN'	75.43%	49.09%	68.89%	57.20%	64.87%	65.49
'VG-TREE'	87.10%	60.00%	76.50%	75.69%	76.24%	81.92
'VG-LIN'	81.75%	64.36%	77.42%	70.24%	74.78%	81.54
'SG-TREE'	91.97%	53.45%	74.70%	81.67%	76.53%	81.16
'SG-LIN'	80.05%	61.82%	75.81%	67.46%	72.74%	76.51
'MG-TREE'	82.97%	56.36%	73.97%	68.89%	72.30%	76.50
'MG-LIN'	77.37%	61.09%	74.82%	64.37%	70.85%	78.65

REFERENCES

F. Baumann, J.Liao, A.Ehlers, B. Rosenhahn. *Motion Binary Patters for Action Recognition*, 2014

E. Bermejo, O.Deniz, G.Bueno, R.Sukthankar. *Violence Detection in Video Using Computer Vision Techniques.* , In Computer Analysis of Images and Patterns, pages 332-339, 2011

G. Bradski , *OpenCV Library* , Dr. Dobb's Hournal of Software Tools, 2000.

C. Chang, C. Hsu, C. Lin. *A Practical Guide to Support Vector Classification*, 2010

C. Chang, C. Lin. LIBSVM: *A Library for Support Vector Machines*, 2001

T. Hassner, Y. Itcher, and O. Kliper-Gross, *Violent Flows: Real-Time Detection of Violent Crowd Behavior*, 3rd IEEE International Workshop on Socially Intelligent Surveillance and Monitoring (SISM) at the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Rhode Island, June 2012

S. Karlsson. *Tutorial on real-time optical flow - File Exchange - MATLAB Central*. [online] Mathworks.co.uk. Available at: http://www.mathworks.co.uk/matlabcentral/fileexchange/44400-tutorial-on-real-time-optical-flow [Accessed 13 February. 2014], 2014.

I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld. *Learning Realistic Human Actions from Movies* IEEE Conference on Computer Vision and Pattern Recognition, 2008

I. Laptev .O*n Space-Time Interest Points*. In International Journal of Computer Vision, vol 64, number 2/3, pp.107-123, 2005

S. Lazebnik, C. Schmid, J. Ponce. *Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scenes,* In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on pages 2169 – 2178, 2006.

Lindsey I. Smith, A Tutorial on Principal Component Analysis 2002 http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

K. Tapas Mount, D.M. ; Netanyahu, N.S. ; Piatko, C.D. ; Silverman, R. ; Wu, A.Y. *An Efficient K-Means Clustering Algorithm: Analysis and Implementation. Pattern Analysis and Machine Intelligence*, IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume 24, Issue 7), 2002

A. Vedaldi and B. Fulkerson *VL_Feat: An Open and Portable Library of Computer Vision Algorithms*, 2008, http://www.vlfeat.org/

H. Wang, A. Kläser, C. Schmid, C. Liu*. Action Recognition by Dense Trajectorie*s, IEEE Conference on Computer Vision and Pattern Recognition, pages 3169 – 3176, 2011