Anomaly Detection in Surveillance Videos

A Dissertation

Submitted in partial fulfilment of
the requirements for the award of the degree

of

Master of Technology

in

Electronics and Communication Engineering
(with specialization in Wireless Communication)

By

Prakhar Singh (12213014)

Under the Guidance of

Dr. Vinod Pankajakshan

Department of Electronics and Communication Engineering

Indian Institute of Technology, Roorkee

Roorkee, Uttarakhand, India – 247667

May, 2017

Declaration

I hereby declare that the work being presented in this dissertation entitled "Anomaly Detection in Surveillance Videos" towards the fulfilment of the requirements for the award of the degree of Integrated Dual Degree (B. Tech and M. Tech) submitted in the Department of Electronics and Communication Engineering, Indian Institute of Technology Roorkee, Roorkee, India, is an authentic record of my own work carried out during the period from July 2016 to May 2017, under the supervision of Vinod Pankajakshan, Assistant Professor, Department of Electronics and Communication Engineering, Indian Institute of Technology Roorkee, Roorkee, India.

The matter presented in this report has not been submitted by me for the award of any other degree at this or any other institute.

Date:

Place: Roorkee Prakhar Singh

Certificate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Dr. Vinod Pankajakshan

Assistant Professor

Department of Electronics and Communication Engineering

Indian Institute of Technology, Roorkee

Abstract

This work addresses the problem of anomaly detection in surveillance videos. To understand the challenges in this field, a comprehensive review of literature in the field was carried out. A suitable base system was selected from literature and analysed in depth. Then an approach utilizing the Histogram of Optical Flow (HOF) and Support Vector Data Description (SVDD) was proposed to overcome the shortcomings of the base system and improve its performance.

In the pre-processing stage, HOF was used to extract motion information ("events") from video data. These events were then described using a compact feature vector, which encoded both spatial and temporal information. An SVDD, with a non-linear kernel for increased flexibility, then learnt a spherically shaped boundary around the dataset, which was then used to identify anomalous behaviour.

The performance of the proposed approach was evaluated on a publicly available benchmark dataset. The strengths of the approach are its flexibility in detecting a broad range of anomalies, its unsupervised learning method and its ability to learn complex non-linear motion patterns.

Acknowledgements

I would like to express my deepest gratitude to Dr Vinod Pankajakshan, Assistant Professor, Department of Electronics and Communication Engineering, Indian Institute of Technology Roorkee for his invaluable cooperation, motivation and support. Without his active encouragement and assistance, I would not have made headway in this project. I am ineffably indebted to him for his conscientious guidance and encouragement.

I would also like to thank all my fellow scholars and lab staff for their suggestions and support that helped me with my work.

Last but not the least, I express a deep sense of gratitude to my family members and my friends who have been constant source of inspiration during the preparation of this project work.

Prakhar Singh

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AUC	Area Under Curve
FSM	Finite State Machine
GMM	Gaussian Mixture Model
HOF	Histogram of Optical Flow
KDE	Kernel Density Estimation
OF	Optical Flow
PCF	Pixel Change Frequency
PCR	Pixel Change Retainment
ROC	Receiver Operating Characteristic
SVDD	Support Vector Data Description
SVM	Support Vector Machine

Chapter 1 Introduction

1.1 Motivation

Nowadays, the need for automated anomaly detection systems is growing due to increased security concerns. Improvements in technology and reduction in costs have led to a rapid deployment of Closed-Circuit Television (CCTV) cameras. Earlier, the job of video surveillance was performed manually by humans who had to visually analyse video feeds from multiple cameras simultaneously. However, research shows that even professionals in this field suffer from a reduction in visual attention after analysing the same video monitors for long periods of time [1]. This can easily hinder their capability to spot and appropriately counter threats in real-time [2], turning existing surveillance systems into mere storage devices that can only be used for video analysis and evidence gathering after an event has occurred [3]. These reasons have motivated research into the field of anomaly detection in video surveillance over the last decade. Autonomous surveillance of videos can help human operators to efficiently counter possible threats in real-time.

To enhance the security and performance of surveillance systems, it is required to develop algorithms which can enable human operators to work more efficiently or even take over the task altogether. Such algorithms will allow monitoring systems to operate in real-time, and to analyse and interact with their surroundings, instead of just being used as forensic tools after the occurrence of an event. These algorithms will also increase the probability of detection as they can process hundreds of movement patterns in crowded scenes, a feat that cannot possibly be matched by humans. However, in order for these algorithms to be useful in the real-world, it is critical that they have a high probability of detection and a low probability of false alarm. Reliability and robustness are the two key metrics for evaluating the performance of such algorithms.

1.2 Comparison with Traditional Video Surveillance

In traditional techniques for video anomaly detection, modelling of anomalous events from training data that contains both normal and abnormal events is quite often the desired approach to detect anomalies and learn behaviour. Detection is then performed by finding and analysing new observations and testing whether they match

the behaviour model learned from the dataset in the training stage. In contrast, modern anomaly detection approaches attempt to learn only normal behaviour from datasets (“one-class datasets”). This learning is typically statistical. To detect anomalies, these algorithms then test whether the new data conforms to the normal behaviour learned earlier.

By aiming at interesting events directly, traditional methods for video surveillance typically learn very specific “high-level descriptors” that represent the events occurring in a video sequence. These methods then build models from training video sequences that contain the anomalies of interest together with normal data. They then classify the events that have not been seen before as anomalies. These approaches however have a lot of drawbacks. They are unable to cope with unknown behaviour, yielding unpredictable results in such scenarios. They can, in general, only be used on certain controlled video sequences.

Video anomaly detection techniques, on the other hand, are capable of detecting random and “undefined” anomalies as they only attempt to test whether the new events are different from some previously obtained model of normal behaviour. These approaches widen the number and types of anomalies that can be detected, but they pose some difficult challenges depending on the definition of normal behaviour. In addition, these methods cannot sufficiently explain "what" is occurring in the scene, as they never learn anything about anomalous behaviour in the training stage. To address this, additional high-level algorithms can then be used, building a system which can utilize the best of both worlds.

1.3 Defining an Anomaly

Despite the plethora of algorithms and their applications in anomaly detection literature, there is no consensus on how anomalous behaviour is defined. Approaches in the literature refer to anomalies as unusual events, abnormalities, suspicious or irregular events, etc.

On a broad level, an anomaly can be defined as “an observation that does not follow expected normal behaviour” [4]. For the purpose of anomaly detection in surveillance videos, anomalies can be considered “as sequences of motions that do not confer or stand out with their surroundings”. This allows statistical approaches to be used for anomaly detection. By this definition, events that have a low probability of

occurrence according to a pre-learnt “probabilistic model of normal behaviour” are referred to as anomalous.

This definition of anomaly detection suffers from certain drawbacks due to which anomalies that can found become limited. Firstly, this definition forces anomaly detection to inferred from a certain context. It is quite possible that an event that is normal at one time, may be anomalous at a different time. For example, in a video sequence containing traffic interactions, certain events like a U-turn or road crossing are dependent on the state of traffic lights. Secondly, in this definition, the features and the level (“scale”) at which normal behaviour is defined [5] limits the anomalous events that can be detected. For example, an event that would be considered as an anomaly at one level can be normal when considered at a different level. Because of the different methods of defining and categorizing anomalous behaviour, a lot of varied approaches that utilize different techniques can be found in the literature.

1.4 Challenges

Probabilistic definitions of anomalies are quite simple to understand and define intuitively, but numerous factors add significant technical hurdles in the field of anomaly detection.

Firstly, the very definition of an anomalous event is quite heavily dependent on how normal behaviour is modelled and which features are obtained from a training sequence. To be more specific, video context, extracted features and the scale of operation ultimately determine what types of anomalies can be detected.

Secondly, non-stationary contexts change what is “normal” at different moments in any particular scene. A robust and efficient system should be able to grasp the dynamics of a such a scenario and detect anomalies taking these into account.

Lastly, anomalies are generally “very infrequent, sparse, and hard to predict” [5]. The examples of anomalous behaviour found in training sequences are thus very limited, and quite often, not present at all. This makes validation of techniques for anomaly detection much harder.

1.5 Document Structure

Section 2 of this document provides a thorough literature review in the field of anomaly detection in surveillance videos. It first describes the types of approaches found in the literature and briefly introduces some of the most common techniques and compares them.

Section 3 gives a description of the base system that was chosen for implementation, providing an in-depth analysis of its working.

Section 4 analyses the performance of this system on standard datasets in anomaly detection literature.

Sections 5 proposes some enhancements to this base system to address its shortcomings and improve its performance.

Section 6 evaluates the efficacy of the proposed system and analyses its performance.

Section 7 provides a summary of this work and provides some scope for future research.

.

Chapter 2 Review of Literature

2.1 Overview of Approaches to Anomaly Detection

Among the different techniques found in the anomaly detection literature, “a distinction can be made between those approaches that rely on extraction of object trajectories, and those that do not” [6]. The former approaches use pre-processing techniques for object segmentation and tracking, while the latter work with different object and/or pixel properties. Different object-based techniques use features that are derived from objects present in a scene, while pixel techniques use features that are extracted on the pixel level.

Object-based techniques use features that are extracted from looks (“appearances”) or object motion. Among appearance based features, blob size and blob texture are frequently used. Motion features that are typically extracted via object tracking and are also quite popular in literature. These object-based techniques use object tracking which yields trajectories as series of object positions over a certain period of time. Using these trajectories, different features such as speed, direction of movement and orientation can be extracted and further utilized.

Pixel-based techniques include methods that extract “spatio-temporal features”, e.g., “pixel change frequency”, “histogram of optical flow”, “histogram of oriented gradients”, “filling ratio of foreground pixels”, “magnitude and orientation of gradients”, “accumulation of differences” etc.

2.2 Types of Anomalies

Depending on the context anomalous events can be classified in the following different types [4]:

Point anomalies, which test whether the computed (“extracted”) features at any particular place are different from normal behaviour. These anomalies take into account the current behaviour but do not depend on past behaviour. They also do not depend on the behaviour of nearby objects or points. As an example, if the velocities of moving objects at multiple locations are used to model normal behaviour, a vehicle in the testing sequence that is moving at a speed or direction that is not present in this model would be classified as anomalous. Applications of this technique include detecting and analysing movement of objects at different positions in a scene.

Contextual anomalies on the other hand take into account some sort of the temporal context, i.e., the past behaviour, or the spatial context, i.e., the behaviour of nearby objects. A particular class of contextual anomalies called “sequential anomalies” takes into account the both of these contexts while analysing anomalies in the data obtained from the extracted features. For example, in traffic surveillance data, a vehicle that makes a wrong turn at a crossing may be moving at a "normal" velocity as it passes, but it’s path of motion (“trajectory”) will not be present in “normal” traffic paths. As the path itself is anomalous, the motion pattern of this vehicle can be considered to deviate from normalcy and is considered to be anomalous.

These definitions bring to light the fact that the definition of anomalous behaviour is quite heavily dependent on context, and the complexity of anomaly detection depends on the kind of features being extracted and amount of information being obtained from a video sequence.

2.3 Common Features

Before a detailed review state-of-the-art video anomaly detection techniques found in the literature, a summary of common features that are used by these techniques is discussed.

Kernel Density Estimation (KDE) is “a method to non-parametrically estimate the probability density function of a random variable” [7]. To build a kernel density estimate, interpretations are found in fields that are not inside of a “density estimate”. KDEs are typically used to for accurately modelling path and motion data, and they yield superior results compared to fitting some classical parametric probability function to such data.

Hidden Markov Model (HMM) [8] are finite state machines (FSMs) that are quite commonly utilized to model sequences of data. An HMM is “a statistical model which considers a system to be a Markov process with some unobserved or hidden states”. An HMM abstracts the state of the system, while allowing the output of the system to be observed.

A Gaussian Mixture Model (GMM) is “a probabilistic model which assumes that all data points are generated from a mixture of a finite number of gaussian distributions with unknown parameters” [9]. GMMs can be thought of as generalizations of k-means clustering to include information about the “covariance structure” of the data,

together with some gaussian centres that are present in the data. Approaches can then employ a "likelihood strategy" to produce an anomaly likelihood map.

Optical Flow (OF) is "the pattern of apparent motion of objects in the image between two consecutive frames caused by motion of an object or the turning of the camera" [10]. It is two-dimensional vector field where each vector shows the motion of points of interest from the first image to the next. Quite often, optical flow is further quantized in orientation or magnitude or both to yield the Histogram of Optical Flow (HOF), or the Histogram of Optical Flow Orientation and Magnitude (HOFM).

2.2 Object Trajectory Based Techniques

Approaches that rely primarily on information extracted from object trajectories for anomaly detection purposes in surveillance videos typically fall under this category. These techniques require a pre-processing stage where motion detection is performed to track object movement in a video sequence. Background subtraction is quite often employed for the purpose of moving object detection. After that existing object-tracking techniques available in the literature can be utilized to find object trajectories.

The key benefit of these object trajectory based methods is that they allow building behavioural models in a fully unsupervised manner, i.e., labelled video sequences are not required. Bearing in mind that anomalies are "events that have a low probability of occurrence" [4], clustering methods can be applied to find anomalous events. This is achieved by clustering paths ("trajectories") to find normal behaviour in the data. Anomalies can then be found by finding the relative distance of new unobserved "test" trajectories and comparing them to known "normal" trajectories. Those test trajectories that are sufficiently dissimilar from all known trajectories (clusters) can then be labelled as anomalies.

In [11], Claudio Piciarelli *et al.* describe a trajectory based approach that utilizes Support Vector Machines (SVMs) to identify anomalous behaviour. They consider trajectories as "variable-length sequences of two dimensional coordinates". However, to work with kernels in SVM they require fixed dimensional feature vectors, which they extract using "trajectory subsampling". These are then further classified by them using an SVM classifier with a gaussian kernel utilizing a Euclidian distance metric:

$$k(x_1, x_2) = \exp\left(-\frac{||x_1 - x_2||^2}{2\sigma^2}\right) \quad (2.1)$$

where x_1, x_2 are the extracted fixed-dimension feature vectors.

In [12], F. Jiang *et al.* describe another trajectory based approach to detect different type of anomalies. The authors classify anomalies into three types: point anomalies, sequential anomalies and co-occurrence anomalies (also described in Section 2.2). For point anomalies, the authors observed that the "instantaneous motion of a single object follows certain rules". For example, road traffic on one side of the road has to move in fixed paths, or the traffic at a red light has to stop at certain spots. As many of these events "follow some regular motion rules", [12] detect normal and abnormal behaviour depending on the probability of occurrence (they refer to it as "frequency of appearance", a similar metric). However, a video anomaly may often contain other behaviour that cannot be characterized by the above method. For example, in the same example, consider these two actions: a) entering an intersection from a road and b) making a left turn at that intersection; they constitute normal behaviour individually. But if they occur together, they are anomalous. This anomaly is a typical example of a sequential anomaly. To detect such anomalies, similar to point anomaly detection, the authors find sequences that have a "low probability of occurrences", e.g., taking a right on a road without a turn will be rarer than most other paths in that area. A key difference between point and sequential anomalies is that the length of the latter can be random (possibly only a part of such an event may be observed in the complete sequence).

In [13], a method has been proposed by F. Jiang *et al.* in which object trajectories are extracted and modelled as Hidden Markov Models (HMM), which are then clubbed in groups via "hierarchical clustering". Object trajectories are found from the video and then represented as a time-sequence containing various features like speed, direction, etc for the objects. In a lot of cases, no prior information is available, so all the trajectories that have been extracted from available videos are analysed, and those trajectories that do not conform to normal behaviour, i.e. anomalous trajectories are then differentiated from normal trajectories. This approach utilizes the fact that "normal events demonstrate a commonality of behaviour while anomalous events indicate rareness".

In [14] W. Hu *et al.* extend the above to two levels of hierarchical clustering that utilizes different object properties like object position, speed, direction and size. The

authors put forward two metrics to find a) “point anomalies,” by calculating the likelihood of an anomaly occurring when a subject arrives or leaves a certain location and b) “contextual anomalies”, by calculating the likelihood of abnormal behaviour in a complete trajectory of motion.

However, in [15], T. Zhang *et al.* note that the methods proposed in [12] and [14] have several drawbacks: both of them lack “any probabilistic explanation for anomalous event detection”, and both of them “require the number of clusters in advance”. In [15], the authors then propose a novel approach to overcome these drawbacks. For each foreground object extracted from the training data, a “co-training classifier” is built to classify objects as vehicle or pedestrian (dataset specific classes in their work). Then, two labelled datasets are prepared from the original dataset, one for each of the classifiers. A graph is the then built to “cluster motion events” of each class based on trajectory. Clustering is achieved using a graph-cut algorithm. From the results which yield the corresponding motion sequences for both classes, trajectories are then grouped into clusters again. From every new cluster of trajectories, “entry points”, “exit points” and “primary trajectories” are found.

Object based techniques available in the literature are often affected by challenging scenarios in which basic feature extraction methods like background subtraction and object tracking are not suitable. Background subtraction performs abysmally in dense (crowded) scenes or scenes with changing (non-stationary) backgrounds and lighting changes. Object-based techniques also struggle in crowded scenarios where occlusions are very frequent, resulting in incorrect tracking that adversely affects the any further stages of an anomaly detection system. Computation complexity of tracking also increases exponentially with the number of objects, thus making it unsuitable for real-time applications.

2.3 Pixel Based Techniques

To address the drawbacks of object-trajectory based techniques that find movement information from object tracking, many recent approaches have been proposed that do not try to track objects but rather extract pixel-level or block-level features (by considering the image to be made up of overlapping or non-overlapping blocks). Some of these techniques utilize information from subjects that move across particular areas in a video frame, so they still have to apply some sort of object segmentation technique to extract these features.

In [16], H. Zhong *et al.* propose a very primitive approach to video anomaly detection. They describe a completely unsupervised technique for finding anomalous behaviour in video datasets set using multiple basic attributes. Their approach doesn't require any complex activity model or supervised feature selection technique. They simply split the training sequence into fixed time-length subsequences to extract features, then convert them into "prototypes" to compute a "prototype–segment co-occurrence matrix". They found the motivation for this from a matching problem in "document keyword analysis", which sought a "correspondence relationship" between these extracted features ("prototypes") and video subsequences that can satisfy a "transitive closure constraint".

They argued that this object model-based technique is very useful in scenarios where "normal behaviour" is well-understood and limited. However, in common real-life training sequences, the number of "normal behaviour" event categories that are observed can easily be surpassed by the number of "unusual behaviour" event categories. Hence, defining and modelling normal behaviour in an unlimited (unconstrained) environment is much more difficult than defining what is anomalous.

For this, they propose an approach to exploit the "hard to describe" but "easy to verify" property of anomalous behaviour without attempting to construct any particular models of "normal behaviour". They argue that "a comparison of each event with all other events that have been observed can be used to determine how many similar events are present". If an event is to be classified as normal, many such related events should be present in the larger data set. Otherwise, the event is considered to be anomalous, as even though the nature and type of the events are unknown, they do not constitute normal behaviour. Thus, detecting anomalous events in video sequences does not require any modelling of normal behaviour, but rather requires the ability to compare two events with a good and computationally fast metric.

To demonstrate this approach, they slice the video into fixed-time duration segments for simplicity (with overlapping windows). This slicing is quite often not perfect, but they argue that these video slices contain enough information in them to suitably determine the nature of activity present, e.g. in a nursing home video in their dataset, in a few seconds of video, events can be either people taking a walk for a few steps or lifting an object. Then, a "document clustering algorithm" is utilized by them to classify the segments. To extract motion information, they then apply the following spatial and temporal "motion thresholding filters":

$$F_t(x,y,t) = F(x,y,t) * X_t * X_{x,y} \tag{2.2}$$

$$X_t = te^{-\left(\frac{t}{\sigma_t}\right)^2}, X_{x,y} = e^{-\left\{\left(\frac{x}{\sigma_x}\right)^2 + \left(\frac{y}{\sigma_y}\right)^2\right\}} \tag{2.3}$$

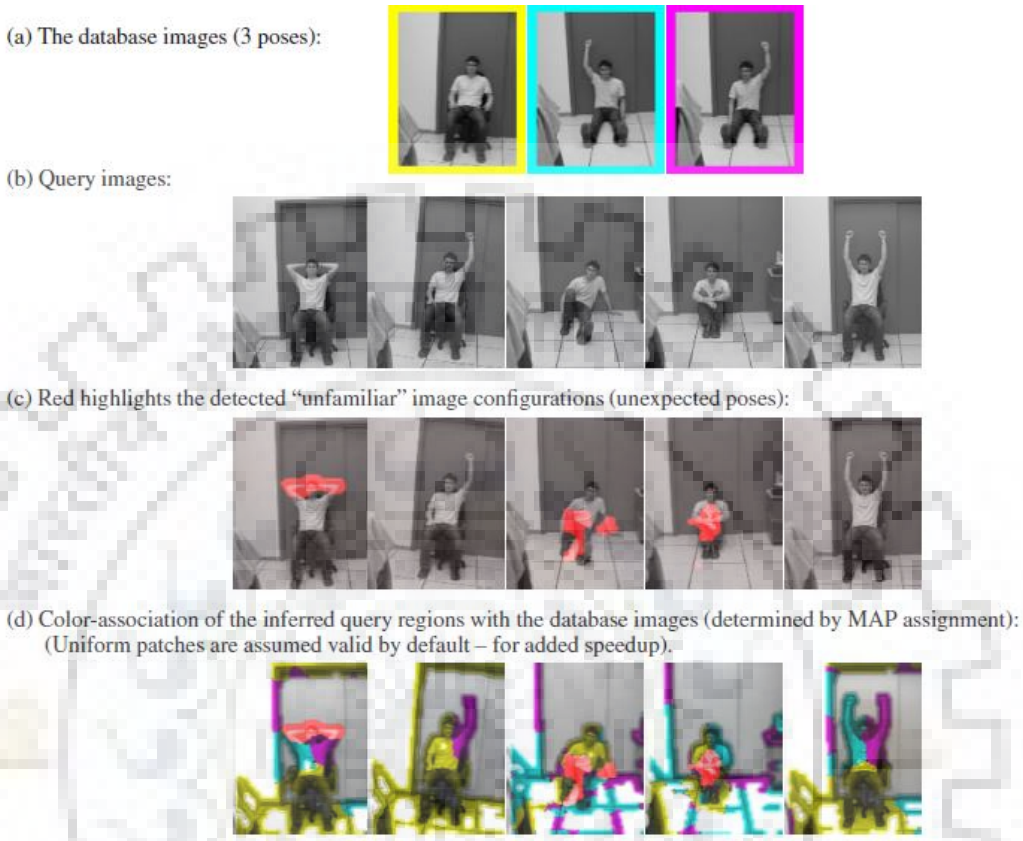

Figure 2.1 Illustration of Boiman and Irani's approach, source [17]

In another seminal paper [17], O. Boiman and M. Irani propose an approach that composes a test image or video sequence ("the query") by utilizing blobs of data ("pieces of puzzle") that were learned from the training data ("the database"). Those parts of the testing video sequences that can be built or composed from the database are considered likely to correspond to normal behaviour, whereas parts of the testing video sequences which cannot be built from the training sequence (or which can be built, but only by using small "fragmented" slices of data) are considered as anomalous. They describe their approach as "an inference process in a general probabilistic graphical model".

Their approach thus learns normal behaviour from very little data, and can be used to test much larger video sequence datasets, even if those particular video sequences are not present in the training sequences. "Local feature descriptors" extracted by them from small video sequences (that are them mixed together to form large video sequences) allow their approach to detect small nuances in normal behaviour, e.g.,

pedestrian walking vs. pedestrian walking while armed with a gun. Moreover, their method is capable of simultaneously identifying normal behaviour in one part of an input frame while detecting anomalous activity in a different part of the frame, achieving "localized detection". An example of their approach is provided in Figure 2.1.

In [18], P. Cui *et al.* describe a very different pixel based approach compared to the ones mentioned above. To reduce noise, they then down sampled features into a "super-pixel". The authors then calculate the likelihood for a video sequence to be anomalous given past events. They achieve this via a Sequential Monte Carlo (MC) framework that models events as Hidden Markov Models (HMMs). Apart from point anomalies, there are also able to find "contextual anomalies". For this however, supervised training data (i.e., labelled instances of normal behaviour) is required. The two major features pixel level features that they use are a) "Pixel Change Frequency" (PCF), which calculates the "changing times" of a pixel over a certain period, and b) "Pixel Change Retainment" (PCR), the amount of time when a pixel has had a different value from its background.

2.4 Anomaly Detection in Crowded Scenes

In contrast to sparse or traffic scenes, crowded scenes require very different approaches. This is because the approaches described above mainly rely on the presence of a static background, which is generally not available in crowded scenes.

A technique to detect anomalies in crowded scenes is proposed in [19] by F. Jiang *et al.* First, the authors extract motion information from input video sequences in the form of "spatio-temporal patches", characterized by dynamic texture. Next, they extract texture information from these patches in moving parts of the video sequence. They further cluster these patches into "behaviour categories" A and B. They note that these patches roughly correspond to motion patterns in the video sequence data. Contextual information for each motion blob is then extracted by analysing the "behaviour categories" of its neighbours.

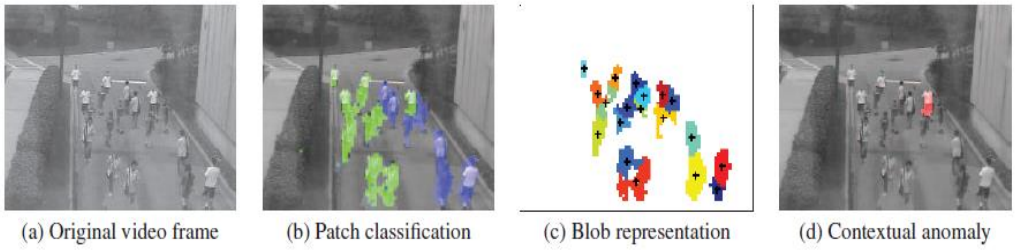

Figure 2.2 Example of contextual anomaly detection, source [19]

Then, based on the "spatial layout" of people walking in the frame, they can automatically gather important contextual information and find the motion patterns that are contextually anomalous. Their key contribution through their work is the introduction of the concept of a "contextual anomaly "into the field of anomaly detection in crowded scenes, as well as proposing a method to find those "contextual anomalies". An example from their work is illustrated in Figure 2.2.

Chapter 3 Description of Base System

3.1 Introduction

From the discussion in previous sections, it can be seen that a lot of authors approach the problem of anomaly detection using object-trajectory based techniques. These techniques typically analyse the scene by making abstractions at the object level, i.e. their basic features involve the extraction of object trajectories in some manner. Using such features in anomaly detection approaches yields several advantages; a) they allow training to be done in an unsupervised manner b) allow for statistical analysis c) ground truth is easier to generate as trajectories can be labelled manually d) these approaches are more intuitive in nature and can be visualized quite easily.

However, object-tracking based approaches have some serious disadvantages as well. They are computationally very expensive, and in general, computation complexity increases exponentially with the number of objects present in the scene. They also perform very poorly in crowded scenes, or even scenes with a large number of objects. In fact, inaccurate tracking can even confuse anomaly detection classifiers causing them to consider behaviour not present in training data as normal. Moreover, these inaccuracies typically accumulate over multiple steps in the training and detection framework.

From the recent trends in anomaly detection literature, it is quite easy to see that many state-of-the-art approaches rely on pixel-level features rather than object trajectories to exploit these advantages. Pixel-based approaches also work much better on crowded scenes as they do not attempt to track individual objects but rather analyse motion patterns as a whole.

For this work, a state-of-the-art (at the time) system that relies on pixel-level features was chosen for implementation and further analysis, and its performance was analysed and benchmarked in detail.

In the following section, the key objectives for this work are summarized, followed by a thorough review of the base system, including the feature extraction, training and detection processes.

3.2 Objectives

Before describing the anomaly detection system in detail, the planned technical objectives for anomaly detection are summarized in this section:

1. Detect suitably rare anomalies
2. Speed of anomaly detection (real time detection)
3. Type of scene:
    i. Basic surveillance (sparse object density)
    ii. Traffic Surveillance / Pedestrian Surveillance (medium object density)
    iii. Crowd Surveillance (high object density)

Point 1 describes the ability of the system to detect suitable rarely anomalies, i.e., the system should be able to learn and correctly predict statistical anomalies up to a predefined probability of occurrence (performance criterion).

Point 2 refers to the speed of anomaly detection i.e., how fast the system can process surveillance videos. A system that works in real time, e.g. a traffic surveillance system for law enforcement, must be able to detect erratic behaviour in live video. However, a system that is analysing past surveillance videos for rare anomalies doesn't need to work as fast.

Point 3, type of scene, refers to the scene depicted in the video. Techniques like background subtraction, object tracking, etc that are suitable for traffic and pedestrian surveillance (medium object density), are not suitable for crowd surveillance (high object density) and vice versa. Both traffic / pedestrian surveillance and crowd surveillance are active fields of research. For this study, the domain of traffic / pedestrian surveillance (medium object density) was chosen.

From literature review (Section 2), it was observed that while a lot of techniques for traffic surveillance are available, those that utilize object trajectories are not very efficient in speed (point 2), and in general, yield poorer performance compared to pixel-based techniques, as described in Section 3.1. So this work focuses on utilizing pixel-level features.

The rest of this section first describes the work of Hanlin Tan *et al.* [20], which utilizes the Lucas-Kanade's Sparse Optical Flow [21] to detect anomalies.

3.3 Overview of the Base System

In [20] the authors describe a framework for fast detection of abnormal events in a video sequence based on sparse optical flow. For this purpose, the low-level feature that they use is optical flow, which computes the speed and direction of movement of different pixels in the image. For feature extraction, they quantize this optical flow by direction and aggregate it at a block level. For training, they compute this feature vector for all input sequences and find the maximum optical flow per block of the image. For testing, they again compute this feature vector for the testing data and apply a simple max criterion filter to detect anomalies.

For this approach, they limit themselves to traffic surveillance videos. In addition, they consider anomalies to be defined only in probabilistic terms, i.e., training data is only assumed to contain the normal class. They also assume the input data, i.e., normal behaviour, to have "temporal stationarity", which includes certain behaviour such as fluttering leaves in the background, regular motion patterns of people, a fixed time of day etc.

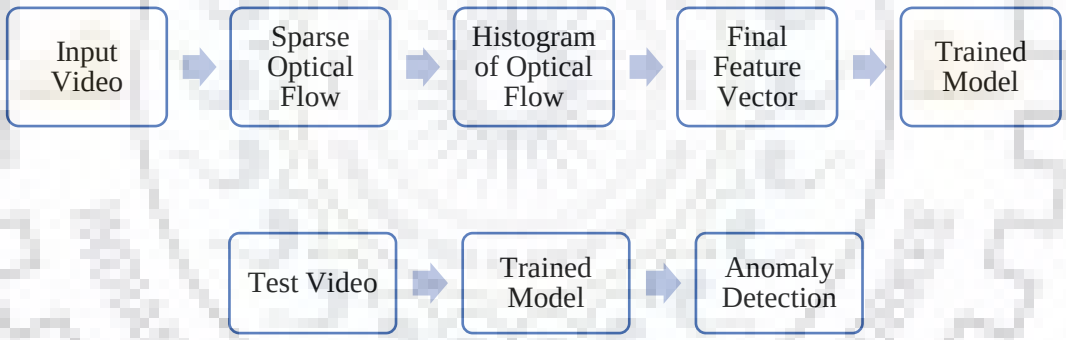

Figure 3.1 Overview of Base System

An overview of the base system is shown in Figure 3.1. The first step is to capture frames from a static camera and build a background model. Then the foreground is extracted. Here, the purpose of foreground extraction in simply to aid in the process of computation of optical flow. No effort is made to track objects in the foreground.

Then corner points are obtained using the "Good Features to Track" algorithm [22]. These corner points are essential to the computation of optical flow. These points yield all corners of an object needed to track it, thus allowing fast and efficient motion tracking (different from object tracking). Its alternative, the dense optical flow, which works on all pixels rather than corners could also have been used. However, as it

added significant computational complexity without any appreciable performance improvements, it was not considered.

The next step is to compute the “forward” optical flow between the current frame and the next frame utilizing the corner points, as well as a “reverse optical flow” between the next frame and the current frame. Then, a “similarity of flows” (and flow error) is computed, and the worst half of the flow is discarded. This flow is then converted to magnitude and angle. Finally, to extract the low-level optical flow feature, this flow is quantized into a fixed number of angular orientation bins (10 in their work). This results in a pixel level feature called the Histogram of Optical Flow (HOF).

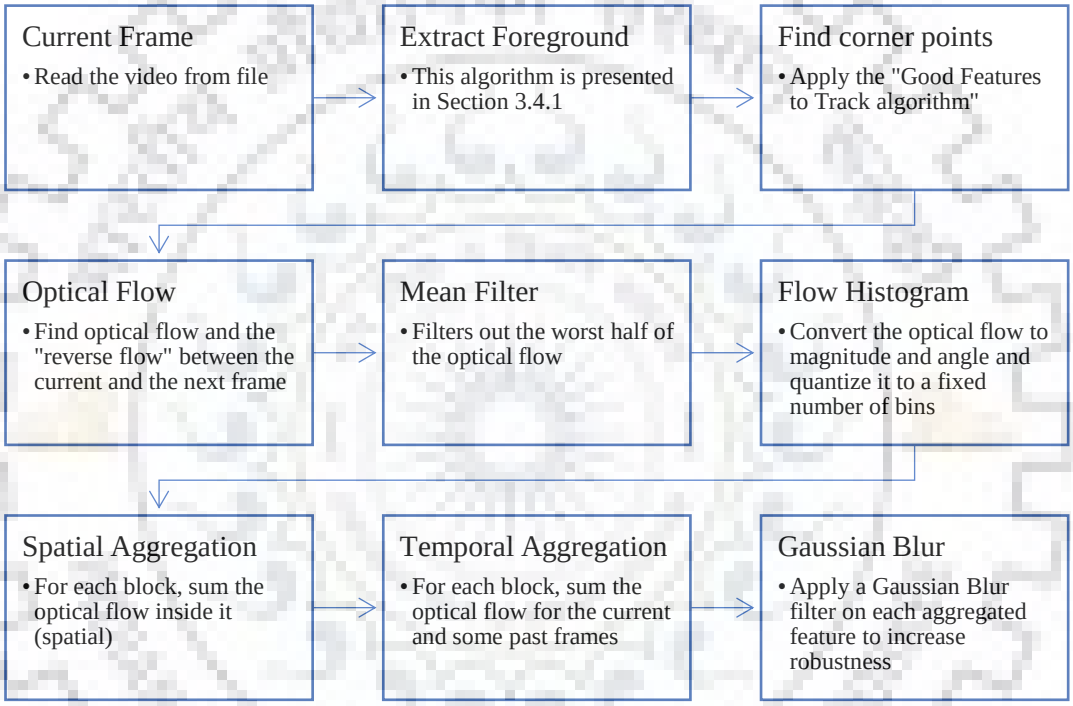

Figure 3.2 Illustration of the Feature Extraction Process

The input image is then divided into fixed size blocks. For each block, firstly, the flow is aggregated spatially for each angle bin separately. Then the flow is aggregated temporally for the previous few frames. After this spatio-temporal aggregation, the last step is to apply a Gaussian Blur to make the computed feature vector more robust.

For training, the authors use a max criterion to find the maximum optical flow per quantized direction per block in the form of a training matrix. For testing, the same feature is computed for the test videos, and the flows that exceed the trained max criterion matrix are considered to be anomalous.

The base system, described in detail below, was implemented using OpenCV [23] and MATLAB [24] and tested on the UCSD Anomaly Detection Dataset (Ped 1) [25].

The following sections describe each step of the base system along with experimental results and screenshots. Note that this section only describes the algorithm used by the base system and the experimental results obtained from intermediate steps. An analysis of the final performance of this system can be found in Section 4.

3.4 Feature Extraction

The first step in the base system is to extract features from the input dataset. An overview of this process is illustrated in Figure 3.2, with each sub-step described in the following sections.

3.4.1 Foreground Extraction

The standard approach described in [26] was used for background subtraction and foreground extraction. The following steps (Figure 3.3) detail this procedure:

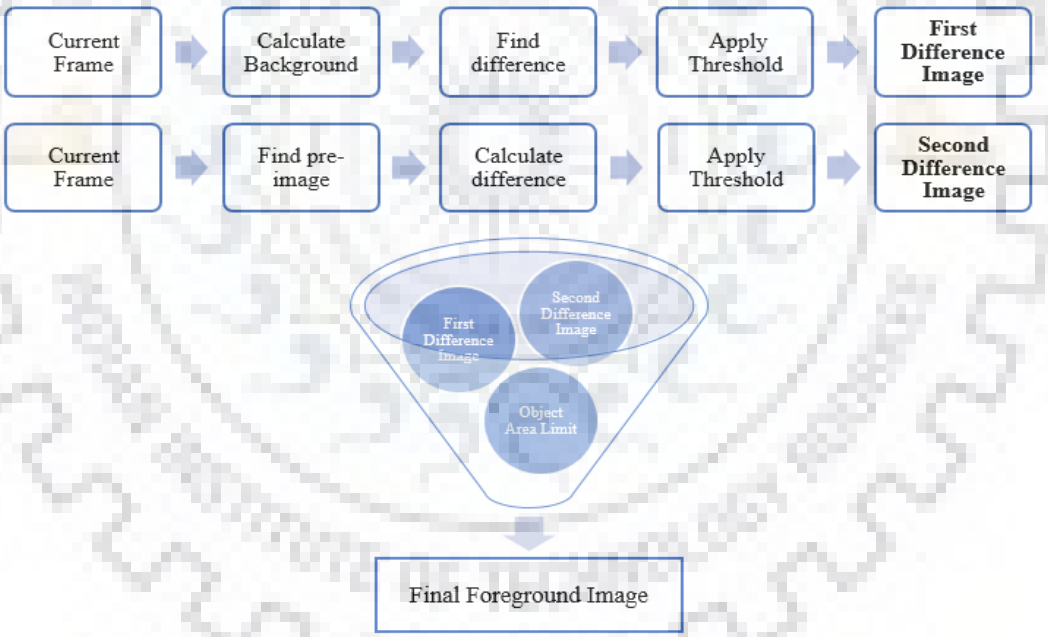

Figure 3.3 Foreground Extraction Process for the Base System

1. Build the histogram for every pixel (64 bins x 3 colours) and calculate the background image.
2. Calculate the first difference image from the current frame and the background.
3. Calculate the second difference image between the current frame and the first video frame ("pre-image").
4. Threshold both difference images and combine the results.

Figure 3.4 Foreground Extraction Process

The experimental results obtained for this step are shown in Figure 3.4. For building this background model experimentally, approximately 150-300 frames were used.

3.4.2 Corner Points

The "Good Features to Track" algorithm [22] by C. Tomasi and J. Shi is an optimal feature selection criterion for tracking objects. The algorithm works by sweeping a window $w(x, y)$ and scoring pixels to find corners.

Specifically, we need to maximize:

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u, y+v) - I(x,y)]^2 \quad (3.1)$$

which can also be expressed as:

$$E(u,v) \approx [u\ v]\ M \begin{bmatrix} u \\ v \end{bmatrix} \quad (3.2)$$

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} {I_x}^2 & I_x I_y \\ I_x I_y & {I_y}^2 \end{bmatrix} \quad (3.3)$$

This maximization was shown to be analogous to the optimization of Shi-Thomasi' metric:

$$R = \min(\lambda_1, \lambda_2) \quad (3.4)$$

The points that have this metric R above a certain threshold are considered to be good feature points. OpenCV [23] has library functions that automatically return N good points to track in an image, which was used in this work.

3.4.3 Optical Flow

Optical Flow (OF) is "the pattern of apparent motion of objects in the image between two consecutive frames caused by motion of object or turning of the camera" [10]. It is two-dimensional vector field where each vector shows the motion of points of interest from first image to the next. Quite often, optical flow is further quantized in orientation or magnitude or both to yield the Histogram of Optical Flow (HOF).

Several assumptions are made before Optical Flow is calculated:

i. Pixel values are constant between consecutive frames
ii. All pixels in a neighbourhood have same motion

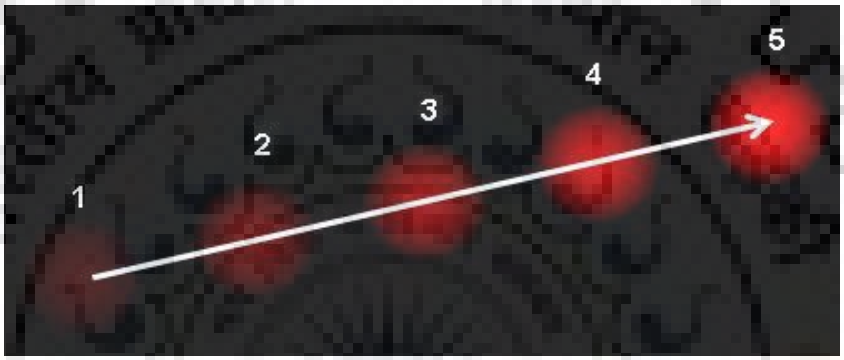

Figure 3.5 Illustration of Optical Flow

Considering a pixel $I(x, y, t)$ in first frame, which moves by distance (dx, dy) after dt time, we can say:

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \tag{3.5}$$

(intensity is assumed not to vary between consecutive images). By taking a Taylor series approximation and cancelling common terms:

$$f_x u + f_y v + f_t = 0 \tag{3.6}$$

$$f_x = \frac{\partial f}{\partial x};\ f_y = \frac{\partial f}{\partial y}$$

$$u = \frac{dx}{dt};\ v = \frac{\partial y}{\partial t} \tag{3.7}$$

Equation 3.6 is the Optical Flow equation. Here, fx, fy and ft are gradients in the spatio-temporal domains. (u, v) are the unknowns. So, this is a single equation with two unknowns, and thus cannot be solved directly.

Various techniques are available to that address this problem. One such solution was given by Lucas-Kanade [21].

Lucas-Kanade's algorithm calculates the above flow equation for a 9-point (8 around the point of interest and the point itself) sub-matrix. Because we assumed that neighbouring pixels have similar motion, these 9 points have the same displacement values. Now equation 3.6 is "over-determined", so the best solution, i.e., the least-mean square solution, is chosen. This gives the values for u and v:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_i f_{x_i}{}^2 & \sum_i f_{x_i} f_{y_i} \\ \sum_i f_{x_i} f_{y_i} & \sum_i f_y{}^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_i f_{x_i} f_{t_i} \\ \sum_i f_{y_i} f_{t_i} \end{bmatrix} \tag{3.8}$$

3.4.4 Forward-Backward Filtering

The optical flow algorithm provides only a least mean squared error solution. Quite often, this error is not acceptable limits needs to be discarded. For this, a forward-backward tracking filter [27] is used. This process is illustrated explained by in Figure 3.7 where:

Figure 3.6 Computation of Optical Flow

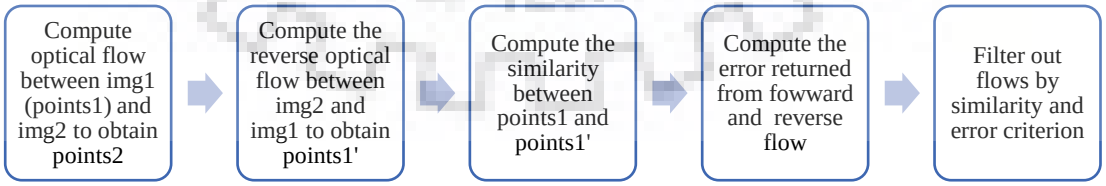

Figure 3.7 Flowchart Depicting Forward-Backward Filtering

- Points1: these are the points that we wish to calculate the optical flow of
- Points2: these are the resultant flow points returned by optical flow

- Points1': these are the estimated original flow points returned by the "reverse flow"
- Mean filter: Simple filter out the worst 50% flows for robustness

The optical flow results obtained experimentally are shown in Figure 3.6.

3.4.5 Histogram of Optical Flow

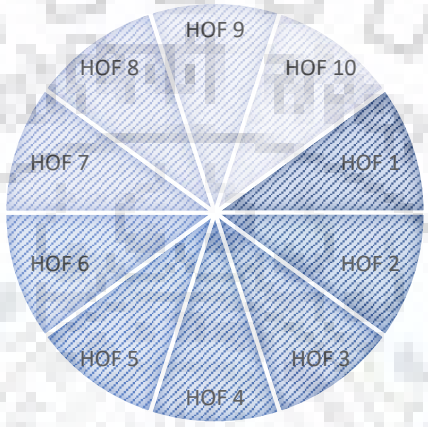

Figure 3.8 Directions for Quantization of Optical Flow

After the optical flow has been computed, the next step is to quantize it. For this, the optical flow is mapped onto a predetermined number of directions to find the Histogram of Optical Flow (HOF). In their work, [20] quantize the optical flow into 10 bins (Figure 3.8).

3.4.6 Integral Images

Integral images (Figure 3.9) are the quickest way to calculate the summation over any sub regions of the image. They can be built simply by computing the sum of all pixels below and the left of the current pixel.

Mathematically, this can be written as:

$$sum(X,Y) = \sum_{x<X,y<Y} image(x,y) \qquad (3.9)$$

Now the summation for any region of the image can be computed in constant time complexity as:

(a) (b)

Figure 3.9 Generation of Integral Image

$$\sum_{x_1 \le x < x_{2,}, y_1 \le y < y_{2,}} image(x,y) = sum(x_{2,}y_2) - sum(x_{1,}y_2) - sum(x_{2,}y_1) + sum(x_{1,}y_1) \quad (3.10)$$

3.5.7 Final Feature Vector

The final steps in the feature extraction first aggregate the Optical Flow (OF) feature:

i. Spatial Aggregation: Using the integral images built in the previous step, the quantized flow components are aggregated inside each block
ii. Temporal Aggregation: The optical flow is summed up per block per orientation for the past few frames

Next, [20] apply a Gaussian blur filter on the "aggregated feature", making the feature smoother and more stable.

Their experimental results show that this blurring of the aggregated feature not only reduces false alarm rate, but also increases the detection rate.

3.6 Training and Detection Framework

After extracting the final feature vector from the data, the next step is to train a model from a given input dataset.

For this, the authors in [20] propose a simple max criterion as follows that calculates the maximum value per feature channel per block. They note that even though the feature values vary dramatically, this method can extract the features.

To detect anomalies, they follow a very simple criterion:

1. HOF magnitudes that are found in training video sequences are normal
2. HOF magnitudes that exceed than the maximum value in the training video sequences are deemed anomalous.

Mathematically, if $F(b,t)$ denotes the extracted feature from a block b at time t, then the classifier would find a max boundary $B(b)$ for each HOF component:

$$B(b) = \max F(b,t) \tag{3.11}$$

where t is considered over all training frames in the input data.

For testing, suppose $f(b,t)$ is a feature vector extracted from the input data. Then, a "distance vector" $x(b,t)$ is be computed as:

$$x(b,t) = f(b,t) - B(b) \tag{3.12}$$

For deciding if a block is anomalous, thresholding can be applied on each HOF component of $x(b,t)$:

$$y(b,t) = \begin{cases} 0, & if\ x(b,t) > \theta \\ 1, & if\ x(b,t) < \theta \end{cases} \tag{3.13}$$

where θ is a predefined threshold vector, and $y(b,t)$ is the classifier output (0 a normal block and 1 for an anomalous block).

Chapter 4 Performance of Base System

4.1 Benchmarking Dataset

A number of video sequences were selected from the UCSD Anomaly Detection Dataset (Ped 1) [25]. This is one of the standard benchmark datasets in the anomaly detection in video literature and nearly every work tests their performance on this dataset. This dataset contains numerous instances of anomalous behaviour specific to the field of anomaly detection in videos, and is available publicly.

This dataset was acquired with a stationary camera that is overlooking a pedestrian walkway. It is divided into training and testing data, where the training data includes different video sequences containing different types of normal behaviour while the testing data contains abnormal behaviour or anomalies. In the "normal" data, the video contains only pedestrians. Abnormal events occur due to:

- movement of non-pedestrians on walkways
- abnormal movement trajectories of pedestrians

Anomalies that typically occur are bikers, skaters, small carts, and people walking across walkways or in the grass nearby. Some examples of people travelling in wheelchair are also present. All anomalous events in this dataset are natural, i.e. they were not staged for creating this dataset. The video footage recorded was then split into various clips of around 200 frames.

A disadvantage of this dataset however is its low quality. The resolution of this dataset is only 253x168, which makes detection of smaller anomalies like bikers and skaters among pedestrians much harder.

4.1.1 Normal Behaviour

All of the training data, and parts of the testing data contain only normal behaviour. Thus, only algorithms that are capable of learning from only one class of data can operate on this dataset. The most commonly occurring subjects in the dataset are pedestrians. Most of the time, pedestrians exhibit "normal" behaviour which includes movement along the walkway, stationary people, movement in groups, mixed movement in multiple directions. While this behaviour is common, there are significant challenges involved in developing a system that analyses this behaviour. The biggest difficulty arises due to the variable density of people along the walkways,

which can easily confuse classifiers. Another difficulty arises due to movement of people in close groups, making individual tracking very hard.

An example of pedestrians in the UCSD dataset is shown in Figure 4.1.

Figure 4.1 Example of Pedestrians in the UCSD dataset

4.1.2 Abnormal Behaviour

In the testing dataset, both normal and abnormal behaviour is present, and the challenge is to identify abnormal behaviour. Different types of anomalies that are present in the dataset generally offer different levels of difficulties as discussed.

Trucks, People in Wheelchairs

Examples of these anomalies are shown in Figure 4.2. These kinds of anomaly can be detected either via tracking individual people, or by analysing their motion pattern. They pose a challenge however, when individual objects are not tracked or analysed.

Figure 4.2 Examples of Vehicle anomalies

Bikers, Skaters

Examples of these are shown in Figure 4.3. These anomalies are similar in nature to the above, but due to them looking visibly similar to pedestrians and having much similar motion patterns, these anomalies are much harder to detect.

Figure 4.3 Example of a hard anomaly: Biker

As mentioned earlier, the major difficulty is faced in detecting bikers and skaters is due to the poor resolution of this dataset.

4.2 Benchmarks

4.2.1 Implementation

As mentioned earlier in Section 3, the base system was implemented using OpenCV [23] and MATLAB [24]. This section discusses the experimental performance of the implemented base system.

4.2.2 Procedure

For testing the implemented system, a training sequence of 300 frames was chosen to build a background model. Then multiple training sequences of minimum 1000 frames was chosen at random (the training dataset contains only normal behaviour) to train the classifier.

For testing, a subset of 5-10 training sequences was chosen from the available testing sequences.

4.2.3 Results

A few frames from the results from obtained from a particular set of training and testing sequences is shown in Fig 4.4.

Figure 4.4 Detection of Anomalies in the Implemented Base System

The implemented system uses a Histogram of Optical Flow (HOF) based model, and excels at detecting anomalies in scenes with lots of motion. It doesn't attempt to identify individual objects or any object-level features. This makes the detection of stationary anomalies that can only be classified by their type much harder. The base system also only employs a max criterion, which doesn't account for slow moving objects and is very susceptible to noise. The following sections discuss the successful and unsuccessful detections by the base system.

4.2.4 Anomalies Found

As the implemented base system utilizes Histogram of Optical Flow (HOF) at a block level, it was capable of detecting anomalies that were caused by subjects that:

i. were moving very fast
ii. had unusually large size
iii. were moving in anomalous directions

The first category of anomalies that were successfully detected, i.e. objects that were moving very fast, were very easily were efficiently picked up by the base system. This is because the base system is employing a max criterion on the HOF feature vector, which increases rapidly in response to fast moving objects.

The second category of anomalies, i.e., objects which had unusually large size, like trucks, were also detected. This was because of optical flow aggregation at the block level. Due to the unusually large size of these objects (compared to other pedestrians in the dataset), they had many more Optical Flow vectors which, when summed up, exceeded normal optical flow feature values.

The last category of anomalies that involved subjects moving in random directions were also efficiently detected by the base system. This was because of the quantized nature of the optical flow. For most blocks, optical flow is quite often close to null in random directions in the trained model. This allows even slightly anomalous behaviour in random directions to be easily detected.

4.2.5 Anomalies Missed

While the base system worked efficiently in certain scenarios, it failed to detect some anomalies present in the dataset, as discussed below. Most of these arise due to limitations of the Histogram of Optical Flow (HOF) feature vector and the somewhat simple training and detection framework. These typically include:

i. Anomalous objects exhibiting “normal” motion
ii. Partial or full eclipsing of anomalous objects by normal objects
iii. Detection in the presence of noisy behaviour
iv. Slow moving objects
v. Spatial learning

The base system is unable to detect those anomalous subjects that exhibit similar size and motion to “normal” subjects like pedestrians but are different visually (“appearance”). An example of such unsuccessful detections is skaters. As seen from Figure 4.5 when the skater is present in the group of people and moving at a similar velocity, he is very hard to detect via his motion pattern.

Figure 4.5 Example of an Anomalous Subject Depicting "Normal" Motion

The second type of unsuccessful detection cases occur when the test subject (anomaly) is eclipsed by other pedestrians in the dataset. In this scenario, the optical flow algorithm is only able to compute the optical flow of the pedestrians and the anomalous object is missed.

The third category of unsuccessful detections occur when there is noise present in the dataset. This is because the max criterion used by the base system of [20] is very susceptible to noise and any “incorrect” optical flow computation during the training or testing stage can lead to very significant drops in performance.

Another category of subjects not detected by the base system includes slow moving objects. As the base system only includes a max criterion, it is unable to find anomalous behaviour in slow moving objects. This choice by the authors of the base system is understandable as the types of anomalies in slow moving objects is typically complex and cannot be learnt by a max feature criterion.

Another key disadvantage of the base system is its inability to learn spatial information from one part of the input frame or region and apply it to other parts. In fact, it can be seen that the base system effectively learns different models for different parts of the image. This means that examples of normal behaviour learnt from one part of the image will be considered anomalous in nearby regions as the

models limit their learning within their own block. This is in fact a fundamental limitation of the base system.

Chapter 5 Proposed Enhancements

5.1 Introduction

The implemented system is not capable of detecting anomalies that arise from slow moving objects or complex motion patterns. In addition, the base system is not capable of spatial learning, which can allow the system to learn more information from the input videos to improve performance (See Section 4).

To solve these problems, this work proposes a modified feature vector that employs the same basic low-level feature i.e., Histogram of Optical Flow (HOF), but encodes additional spatial and net flow information. In addition, to take advantage of this modified feature vector, this work proposes the use of a support-vector based machine learning technique called Support Vector Data Description (SVDD) [28].

The remainder of this section gives a description of the proposed feature vector and the SVDD method.

5.2 Proposed Feature Vector

As detailed in Section 3.4, the feature descriptor of the base system only employs the components obtained from Histogram of Optical Flow (HOF). To compute this Lucas-Kanade's algorithm [21] is used on good feature points to compute the optical flow. This feature vector, however, limits the amount of spatial information that can be learnt. Also, this feature vector is only suitable for classifiers that require all features to be of similar type. By making improvements to this feature vector, more powerful classifiers can be use that are capable of exploiting this additional information.

The first proposed improvement is the addition of the spatial x and y coordinates to each block to the feature vector. Many algorithms like nearest neighbour search, One-Class SVM and SVDD can utilize these to learn additional behaviour from nearby blocks in addition to the block itself. Also, the addition of these coordinates within the feature vector itself will allow a single classifier model to be trained over all blocks of the image. If this were not the case then multiple models would be needed, which is the approach that the base system uses.

This work also proposes the addition of Net Optical Flow (NOF) magnitude and direction to the feature vector as well. This is because in some cases the individual quantized flow components may be small but when added they might yield a net optical flow which can be used for detection. The justification for this addition can be verified by experiments only as the relationships between NOF and anomalies is expected to be non-linear, and may not be intuitively obvious.

The next section introduces the problem of outlier detection and gives a brief overview of available techniques.

5.3 Outlier Detection (One-class datasets)

This section gives a brief description of the problem of outlier detection and data descriptions. Outlier detection simply refers to the identification of objects, events or observations which do not conform (cannot be explained) to an expected pattern or other known objects in a dataset. A data description then, as its name suggests, is simply a model that is capable of learning these expected patterns and rejecting unknown patterns.

The use of a data description arises when the input data has only one class, or when the other classes are too sparse ("undersampled") for other multi-class methods to learn from. But learning from only one class of data poses unique challenges that are not faced by other machine learning techniques. For example, when only one class is present, it's not clear what shape the data description must take, as only the "normal" behaviour is defined.

A simple approach is to simply generate outliers around a given dataset and then train a binary or multi-class classifier to build the data description. This doesn't work however when a set of "near-target" observations cannot be generated (highly-dimensional or complex data). More often than not, this problem is solved by the use of a probabilistic density model. There, methods attempt to estimate the probability density of the normal data (and sometimes the outlier data). Such an example specific to anomaly detection in videos can be found in [9]. But these typically require a lot of data and fail to work when the input data does not sufficiently represent the entirety of "normal" behaviour.

To overcome these drawbacks, Vapnik [29] argued that it is not necessary to solve a more general problem as a sub-problem for outlier detection, and attempts to estimate

complete densities instead of boundaries might require too much data and even then, yield poor results.

For learning boundaries, Sch¨olkopf [30] proposed a support vector based technique (known in the literature as One-Class SVM) that attempts to learn a hyperplane that separates the data from the origin. Another support-vector based method called Support-Vector Data Description (SVDD) by Tax and Duin [28] obtains a spherically based boundary around an input dataset with minimal volume. In this work, their description was used to learn from extracted features because of its robustness and efficiency compared to some of the methods mentioned above.

The following section briefly outlines their approach, and the subsequent sections describe its usage in this work.

5.4 Support Vector Data Description (SVDD)

The method of Support Vector Data Description by Tax and Duin (SVDD) [28] uses a spherical approach build a data description for a dataset. The algorithm "obtains a hyper-spherical boundary around the data in its feature space. The volume of this hypersphere is then minimized to reduce the effect of outliers"

Mathematically, consider a sphere with a centre $\boldsymbol{a}$ and a radius R which gives a closed boundary around a dataset in its feature space. Then the task is to minimize an error function:

$$F(R, \boldsymbol{a}) = R^2 \tag{5.1}$$

with the constraints:

$$\|x_i - \boldsymbol{a}\|^2 \leq R^2, \qquad \forall i \tag{5.2}$$

To allow outliers, a soft-margin SVM like penalty is introduced on distances greater than R^2:

$$F(R, \boldsymbol{a}) = R^2 + C \sum_i \xi_i \tag{5.3}$$

$$\|x_i - \boldsymbol{a}\|^2 \leq R^2 + \xi_i, \qquad \forall i \tag{5.4}$$

where $\xi_i \geq 0$ are the slack variables. These constraints can then be applied together using and solved to find Lagrange Multipliers α_i, which then yield the centre a and radius R as:

$$a = \sum_i \alpha_i x_i \tag{5.5}$$

$$R = (x_k \cdot x_k) - 2\sum_i \alpha_i (x_i \cdot x_k) + \sum_{i,j} \alpha_i \alpha_j (x_i \cdot x_j) \tag{5.6}$$

where x_k is any support vector. Flexibility in SVM can be introduced by using non-linear kernels as inner products with a simple substitution:

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \tag{5.7}$$

A few examples of common kernels are shown below.

Polynomial kernel (degree d)

$$K(x_i, x_j) = (x_i, x_j)^d \tag{5.8}$$

Gaussian Kernel (kernel parameter s)

$$K(x_i, x_j) = exp\left(-\frac{\|x_i - x_j\|^2}{s^2}\right) \tag{5.9}$$

Exponential kernel (kernel parameter γ)

$$K(x_i, x_j) = exp(-\gamma\|x_i - x_j\|), \qquad \gamma > 0 \tag{5.10}$$

To test whether a data-point is in-class or outlier, the distance to the centre of the hypersphere can be compared to its radius. Examples of these kernels are given in Figure 5.1.

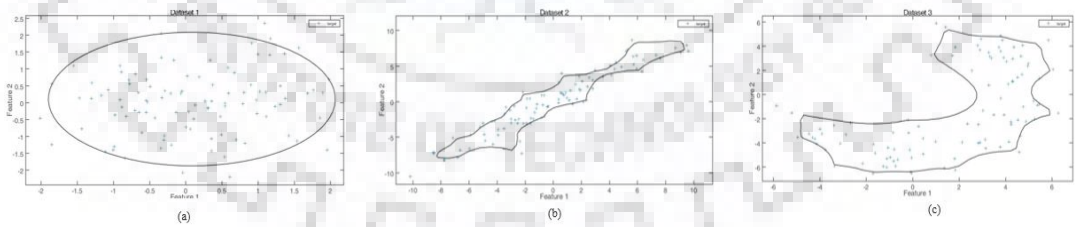

Figure 5.1 Examples of Different Kernels in SVDD
(a) Linear, (b)-(c) Exponential

5.5 Proposed Framework

A block diagram of the proposed training system is shown if Figure 5.2.

Figure 5.2 Proposed Model Training Framework

Starting with a given input video, first the Optical Flow (OF) is computed on good feature points [22] using Lucas-Kanade's algorithm [21] and quantized to obtain a Histogram of Optical Flow (HOF). Then the proposed spatio-temporal feature vector (Section 5.2), which encodes x, y block coordinates, net optical flow and magnitude and HOF components is computed. The final step is to train a Support Vector Data Description (SVDD), which learns a spherical boundary around an input dataset.

For testing the same spatio-temporal feature vector is computed for the test data (Figure 5.3). Then the trained SVDD model outputs the likelihood of a testing sequence being anomalous by computing the distance between the testing sequence and the centre of the hyper-sphere. By comparing this distance to the hyper-sphere's radius (see the above section) a likelihood of anomalous behaviour is calculated.

Figure 5.3 Proposed Testing Framework

Chapter 6 Experimental Work

6.1 Introduction

For testing the efficacy of the new system, its performance was evaluated on the same video sequences used for benchmarking the base system (see Section 4.1). In the subsequent section, the experimental work conducted to analyse and improve the base feature vector is documented. Then the procedure used to find optimum parameters for training Support Vector Data Description (SVDD) models has been described. Finally, the results that were obtained using the proposed method have been shown, along with a comparison to the state-of-the-art.

6.2 Analysis of the Feature Vector

To analyse the final feature vector, first an analysis of the Histogram of Optical Flow (HOF) feature is conducted, which was used by the base system. Then the experimental results from spatial feature coordinates are shown. Finally, the nature of the Net Optical Flow (NOF) features (magnitude and direction) are shown, along with their experimental performance. The theoretical descriptions and justifications for these features are described in Section 5.

Note that all of the following features are calculated per block per frame.

6.2.1 HOF Feature

The HOF feature was computed by quantizing the Optical Flow outputs into 10 channels.

To visualize this feature, a block was chosen at random from the dataset and the flow feature vector were plotted. Examples of this feature are plotted in Figure 6.1. As observable from this figure, these features have significant variations among themselves. A non-linear classifier is thus needed to fully learn the nuances of the training dataset video sequences from these features.

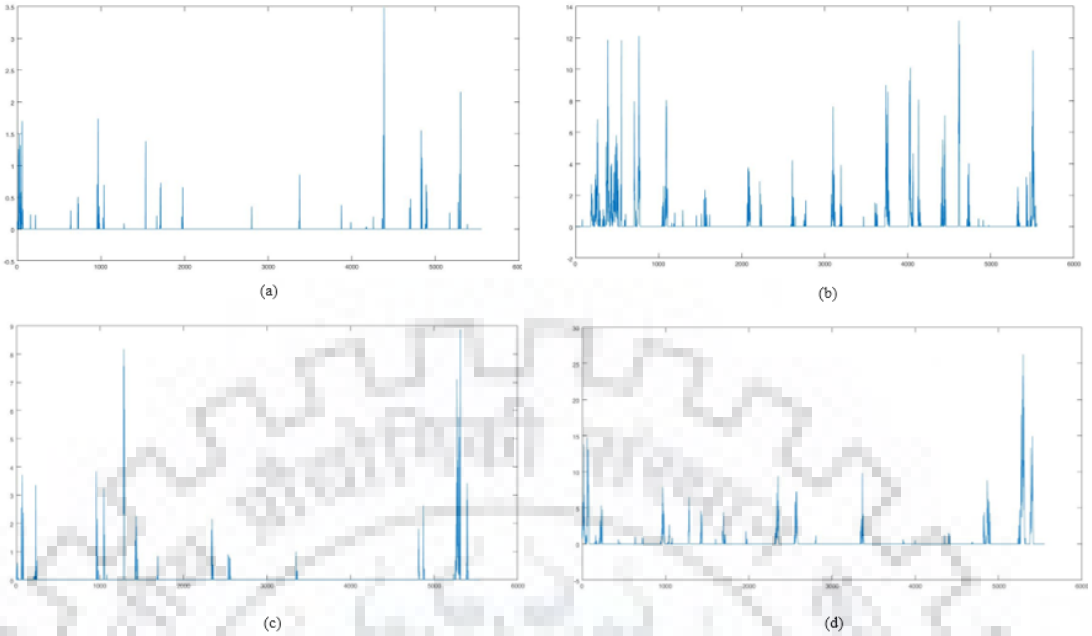

Figure 6.1 Plots of HOF feature channels

6.2.2 Spatial Coordinate Feature

In addition to the HOF components, the proposed feature vector also contains two spatial coordinates x and y. The max training criterion of the base system required each feature channel to be similar, but with SVDD, that restriction doesn't exit. The inherent linearity in the base system would not have allowed it to learn and utilize these features, but SVDD with non-linear kernels can exploit this additional information.

Table 6.1 Comparison of Different Feature Vectors

Feature	Dataset 1 (AUC)	Dataset 2 (AUC)
HOF	78.1	77.5
HOF + SC	80.2	81.4
HOF + NF	79.3	78.2

The motivation behind including spatial coordinates in the proposed feature vector stems from their ability to allow a classifier to learn spatial information from the video sequences directly, without having to train a separate classifier for different regions of the image.

Table 6.1 depicts the performance of spatial coordinate features when compared to a pure HOF based feature vector.

6.2.3 Net Optical Flow (NOF) Feature

The NOF features refer to the magnitude and orientation of the net optical flow in each block of the image. The performance of the NOF feature (magnitude and angle) is shown in Figure 6.2

Figure 6.2 NOF Feature Plots (Magnitude and Angle)

6.3 Kernel Selection

In a manner similar to SVM, SVDD is flexible in the choice of kernel, allowing it to model complex non-linear boundaries around a training dataset. A non-linear kernel effectively changes the "similarity measure" between input points, and allows SVDD to learn highly non-linear boundaries. Some common kernels usable in SVDD are

listed in Section 5.4. The choice of the best kernel in this work was found experimentally, as shown below (Table 6.2). Based on these results, an exponential kernel was chosen.

Table 6.2 Results for Kernel Selection

Kernel	AUC
lin	53.14
pol (deg 2)	76.23
pol (deg 3)	81.58
exp (par 10)	85.06
exp (par 15)	83.87

6.4 Experimental Results

The most common metric in the video anomaly detection literature is the Receiver Operating Characteristic (ROC) curve. For training around 2500 input frames were used. For testing a subset of video sequences were chosen from the UCSD Anomaly Detection Dataset (Ped 1) [25]. More details on the benchmarking technique are present in Section 4. The ROC curve obtained is shown in Figure 6.3.

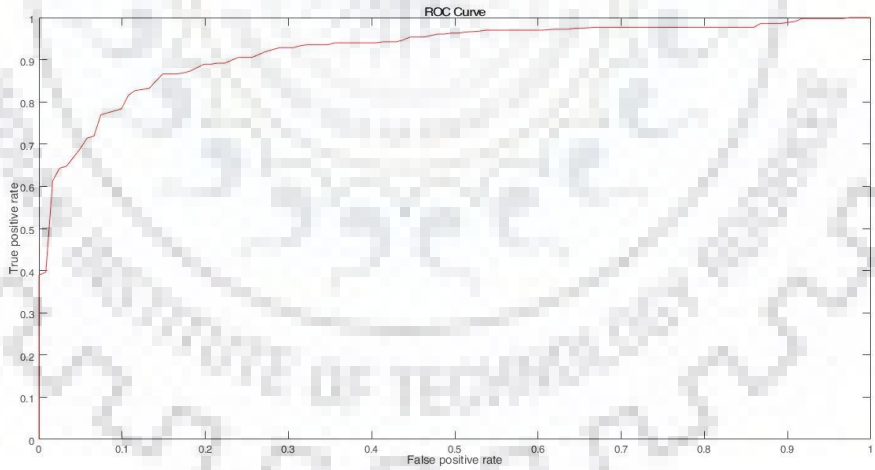

Figure 6.3 Receiver Operating Characteristic (ROC) Curve

To compare our work with the state-of-the-art, the following other approaches were considered: Mixture of Dynamic Texture (MDT) [31], Chong et al [32], Gaussian Process Regression (GPR) [33], Social Force (SF) [34], and Adam et al [35]. The results obtained by the proposed approach are illustrated in the following table.

As can be seen from Table 6.3, on UCSD Ped1, the proposed method performs comparable to state of-the-art in the frame level AUC metric. The detection part of this method runs in real time on the benchmark dataset. A comparison of absolute frames-per-second speeds to other methods in literature was not possible due to differences in available hardware and implementation platforms.

Table 6.3 Performance of Proposed Method

Method	**AUC (Frame)**
Proposed Method	86.9%
Implemented System	82.4%
MDT [31]	81.8%
Chong *et al* [32]	89.9%
GPR [33]	83.8%
SF [34]	67.5%
Adam et al [35]	65%

6.5 Improvements to Base System

In Section 4.2.5, may drawbacks of the base system were noted. Most of these have been improved upon by the proposed system, resulting in an AUC improvement of 4.5%, as discussed in the following paragraphs.

A key class of unsuccessful detections by the base system was those of anomalous objects that exhibited “normal” motion. This was because of the max criterion used by the base system. But due to the non-linear nature of learning by SVDD, complex motion patterns like this class can be differentiated from normal behaviour and such anomalies can be suitably detected.

Another disadvantage of the base system was its susceptibility to noise. The maximum threshold values in the max-criterion could easily be corrupted by noisy data. In SVDD however, a certain class of input is treated as erroneous (similar to soft-margin SVM), which easily gets rid of such noisy data.

The base system, however, still faced difficulty in processing scenes where the object motion is too slow; this is due to an inherent limitation of optical flow as a feature vector. A simple solution to this can be found in Zhang’s work [36], where they model objects separately to detect “appearance anomalies”. For this they learn object appearance features via a separate classifier and then combine the results from a motion classifier and appearance classifier to improve performance on such test cases.

Chapter 7 Summary and Future Work

7.1 Summary of this Work

The field of video anomaly detection has gained considerable attention due to the large number of surveillance cameras installed and the enormous computing resources available today.

In this work, an extensive study of literature on anomaly detection in surveillance videos was carried out, and a method was proposed for the purpose of anomaly detection in surveillance videos. The major contributions of this work have been summarized in the following paragraphs.

Firstly, the key challenges in the field of anomaly detection were identified by a thorough literature survey of existing works. These were found to be probabilistic definition of anomalies (as compared to pre-defined anomalies), one-class training datasets and a need for real time algorithms.

An existing framework that attempted to addressed these challenges was then chosen from literature and implemented. This framework modelled behaviour using pixel level abstractions by employing Histogram of Optical Flow (HOF) as a low-level feature. Then a set of video sequences were chosen from a publicly available dataset to benchmark this framework and find its merits and demerits.

After a thorough analysis of this chosen framework, enhancements were proposed to address its shortcomings. Enhancements to the feature vector included the addition of block coordinates and Net Optical Flow (NOF) magnitude and direction. The spatial coordinates encoded spatial information about blocks into the feature vector, while the NOF features added additional information that a classifier might not have inferred directly from HOF components. Then a support-vector based training (modelling) algorithm was also proposed exploit the non-linearity in the data.

Then the proposed method was benchmarked and it was found to improve upon many shortcomings of the base system like the ability to learn complex non-linear motion patterns. The proposed method also found to work well on crowded scenes where some other methods in the literature faced difficulty partly because this method only attempts to identify motion patterns in the foreground (pixel based abstractions), rather than individual objects (object based abstractions). A comparison to other

approaches in the literature showed the proposed system performing comparably to the state-of-the-art.

7.2 Future Work

A number of different areas and video sequences have been identified where further lines of research can be considered.

- Fixed spatio-temporal volumes. The temporal the duration of an event was fixed when extracting the Histogram of Optical Flow (HOF) feature, in both the base and the proposed system. Research can be conducted into extending this approach to learn variable time-length anomalies
- Detecting stationary anomalies. HOF is unable to detect objects that are moving too slow or are stationary. A lot of pixel-abstraction based techniques similar to HOF also require some sort of motion to work properly. To address these, other low-level features that can address both stationary and moving objects should be investigated.

Bibliography

[1] M. Green, J. Reno, R. Fisher and L. Robinson, "The appropriate and effective use of security technologies in U.S. schools: A guide for schools and law enforcement agencies series: Research report," *National Institute of Justice,* 1999.

[2] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, H.Merkl and S. Pankanti, "Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking," *IEEE Signal Process. Mag.*, vol. 22, no. 2, pp. 38-51, 2005.

[3] C. S. Regazzoni, A. Cavallaro, Y. Wu, J. Konrad and A. Hampapur, "Video Analytics for Surveillance: Theory and Practice," *IEEE Signal Process. Mag.*, vol. 27, no. 5, pp. 16-17, 2010.

[4] V. Chandola, A. Banerjee and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, pp. 1-58, July 2009.

[5] W. Li, V. Mahadevan and N. Vasconcelos, "Anomaly Detection and Localization in Crowded Scenes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. PP, no. 99, pp. 1-1, 2013.

[6] O. P. Popoola and K. Wang, "Video-Based Abnormal Human Behavior Recognition— A Review," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,* vol. 42, no. 6, pp. 865-878, 2012.

[7] B. W. Silverman, Density Estimation for Statistics and Data Analysis, London - New York: Chapman & Hall, 1986.

[8] S.S.Joshi and V.V.Phoha, "Investigating Hidden Markov Models Capabilities in Anomaly Detection," *Proceeding of 43'd ACM Southeeast Conference,* Mar 2005.

[9] F. Li, W. Yang and Q. Liao, "An efficient anomaly detection approach in surveillance video based on oriented GMM," *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on,* Mar 2016.

[10] "OpenCV: Optical Flow," OpenCV, [Online]. Available: http://docs.opencv.org/trunk/d7/d8b/tutorial_py_lucas_kanade.html.

[11] P. Claudio, M. Christian and F. G. Luca, "Trajectory-Based Anomalous Event Detection," *IEEE Transactions On Circuits and Systems for Video Technology,* vol. 18, no. 11, Nov 2008.

[12] F. Jiang, J. Yuan, S. A. Tsaftaris and A. K. Katsaggelos, “Anomalous video event detection using spatiotemporal context,” *Computer Vision and Image Understanding*, vol. pp, no. 115, pp. 323-333, Mar 2011.

[13] F. Jiang, Y. Wu and A. K. Katsaggelos, “A Dynamic Hierarchical Clustering Method for Trajectory-Based Unusual Video Event Detection,” *IEEE Transactions on Image Processing*, vol. 18, no. 4, p. 907–913, 2009.

[14] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan and S.Maybank, “A system for learning statistical motion patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, p. 1450–1464, 2006.

[15] T. Zhang, H. Lu and S. Z. Li, “Learning semantic scene models by object classification and trajectory clustering,” *IEEE Conference on Computer Vision and Pattern Recognition*, p. 1940–1947, 2009.

[16] H. Zhong, J. Shi and M. Visontai, “Detecting unusual activity in video,” *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, p. 819–826, 2004.

[17] O. Boiman and M. Irani, “Detecting Irregularities in Images and in Video,” *Int Journal on Computer Vision*, vol. 74, no. 1, pp. 17-31, 2007.

[18] P. Cui, L.-F. Sun, Z.-Q. Liu and S.-Q. Yang, “A Sequential Monte Carlo Approach to Anomaly Detection in Tracking Visual Events,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2007.

[19] F. Jiang, Y. Wu and A. K. Katsaggelos, “Detecting contextual anomalies of crowd motion in surveillance video,” *IEEE International Conference on Image Processing* , pp. 1117-1120, 2009.

[20] H. Tan, Y. Zhai, Y. Liu and M. Zhang, “Fast Anomaly Detection in Traffic Surveillance Video Based on Robust Sparse Optical Flow,” *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, 2016.

[21] B. Lucas and Kanade, “An iterative image registration technique with an application to stereo vision,” *IEEE Conference on,Computer Vision and Pattern Recognition*, pp. 674-679, 1981.

[22] C. Tomasi and J. Shi, “Good features to track,” *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 593-600, 1994.

[23] “OpenCV Framework,” OpenCV, [Online]. Available: http://opencv.org/.

[24] “MATLAB,” Mathworks, [Online]. Available: https://www.mathworks.com/products/matlab.html.

[25] W. Li and V. Mahadevan, “UCSD Anomaly Detection Dataset,” [Online]. Available: http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm.

[26] M. Nawaz, P. J. Cosmas, A. Adnan, M. I. U. Haq and E. Alazawi, "Foreground Detection using Background Subtraction with Histogram," *Broadband Multimedia Systems and Broadcasting, IEEE International Symposium on ,* June 2013.

[27] Z. Kalal, "Tracking learning detection," *PhD Thesis, University of Surrey,* vol. 30, pp. 555-560, Jan 2011.

[28] D. M. Tax and R. P. Duin, "Support Vector Data Description," *Machine Learning,* vol. 54, no. 1, pp. 45-66, Jan 2004.

[29] V. Vapnik, Statistical Learning Theory, Wiley, 1998.

[30] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor and J. C. Platt, "Support vector method for novelty detection," *Advances in neural information processing systems,* pp. 582-588, 2000.

[31] V. Mahadevan, W. Li, V. Bhalodia and N. Vasconcelos, "Anomaly detection in crowded scenes," *Computer Vision and Pattern Recognition (CVPR), International Conference On,* pp. 1975-1981, 2010.

[32] Y. Chong and Y. Tay, "Abnormal Event Detection in Videos using Spatiotemporal Autoencoder," *preprint arXiv:1701.01546 [cs.CV],* 2017.

[33] K.-W. Cheng, Y.-T. Chen and W.-H. Fang, "Video anomaly detection and localization using hierarchical feature representation and Gaussian process regression," *Computer Vision and Pattern Recognition (CVPR), International Conference On,* pp. 2909-2917, 2015.

[34] R. Mehran, A. Oyama and M. Shah, "Abnormal crowd behaviour detection using social force model," *Computer Vision and Pattern Recognition (CVPR), International Conference On,* 2009.

[35] A. Adam, E. Rivlin, I. Shimshoni and D. Reinitz, "Robust real-time unusual event detection using multiple fixed-location monitors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 30, no. 3, pp. 555-560, 2008.

[36] Y. Zhang, H. Lu, L. Zhang and R. Xiang, "Combining motion and appearance cues for anomaly detection," *Pattern Recognition,* vol. 51, pp. 443-452, 2016.

[37] R. V. H. M. Colque, C. A. C. Júnior and W. R. Schwartz, "Histograms of Optical Flow Orientation and Magnitude to Detect Anomalous Events in Videos," *Graphics, Patterns and Images (SIBGRAPI), SIBGRAPI Conference on,* Aug 2015.

[38] S. Wang, E. Zhu, J. Yin and F. Porikli, "Anomaly Detection in Crowded Scenes by SL-HOF Descriptor and Foreground Classification," *Pattern Recognition (ICPR), International Conference on,* Dec 2016.