

**ROBUST APPROACHES FOR ANOMALY
DETECTION APPLIED TO VIDEO
SURVEILLANCE**

RENSSO VICTOR HUGO MORA COLQUE

**ROBUST APPROACHES FOR ANOMALY
DETECTION APPLIED TO VIDEO
SURVEILLANCE**

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

**ORIENTADOR: WILLIAM ROBSON SCHWARTZ,
COORIENTADOR: GUILLERMO CAMARA CHAVEZ**

Belo Horizonte

Agosto de 2018

RENSSO VICTOR HUGO MORA COLQUE

**ROBUST APPROACHES FOR ANOMALY
DETECTION APPLIED TO VIDEO
SURVEILLANCE**

Thesis presented to the Graduate Program
in Ciência da Computação of the Universi-
dade Federal de Minas Gerais in partial ful-
fillment of the requirements for the degree
of Doctor in Ciência da Computação.

ADVISOR: WILLIAM ROBSON SCHWARTZ,
CO-ADVISOR: GUILLERMO CAMARA CHAVEZ

Belo Horizonte

August 2018

© 2018, Rensso Victor Hugo Mora Colque.
Todos os direitos reservados.

Mora Colque, Rensso Victor Hugo

M827r Robust Approaches For Anomaly Detection Applied
To Video Surveillance / Rensso Victor Hugo Mora
Colque. — Belo Horizonte, 2018
xi, 110 f. : il. ; 29cm

Tese (doutorado) — Universidade Federal de Minas
Gerais

Orientador: William Robson Schwartz,
Coorientador: Guillermo Camara Chavez

1. Computação - Teses. 2. Visão por computador.
3. Processamento de Imagens. 4. Sistemas eletrônicos
de segurança. I. Orientador. II. Coorientador.
III. Título.

CDU 519.6*82.10(043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

ROBUST APPROACHES FOR ANOMALY DETECTION APPLIED TO
VIDEO SURVEILLANCE


RENSSO VICTOR HUGO MORA COLQUE


Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:



PROF. WILLIAM ROBSON SCHWARTZ - Orientador
Departamento de Ciência da Computação - UFMG


PROF. GUILLERMO CÁMARA CHÁVEZ - Coorientador
Departamento de Computação - UFOP


PROF. ERICKSON RANGEL DO NASCIMENTO
Departamento de Ciência da Computação - UFMG


PROF. JEFERSSON ALEX DOS SANTOS
Departamento de Ciência da Computação - UFMG


PROF. MOACIR ANTONELLI PONTI
Departamento de Ciência da Computação - USP


PROF. CLÁUDIO ROSITO JUNG
Departamento de Informática Aplicada - UFRGS

Belo Horizonte, 24 de agosto de 2018.

*Dedico este trabajo a Dios quien me da la vida y fuerza para poder realizarlo.
A mis papas: Hugo, Elsa, Rensso, Luz y mi hermana Giovanna,
a mis profesores y amigos Guillermo y William,
y a mis amigos todos!*

Acknowledgments

I would like to express my sincere thanks and gratitude toward the following people who contributed their support and assistance to this thesis.

First and foremost, I am grateful with God, whose gifts me everything. Also, I am deeply grateful to my family, their inspiration and encouragement stimulated me to pursue a Ph.D.

This thesis would not have been possible without continuous support and guidance from my conscientious supervisors. I owe a debt of gratitude to my primary advisor, professor William for his patient supervision, constant encouragement, and profound knowledge, which will benefit me for a lifetime. At the same time, I thank to my co-advisor Guillermo, someday I hope to teach as well as he does.

In this walk, I made many good friends: the noble Victor Hugo, old friend Edward, Carlos from Turkey, Alberto Hideki, strong Karla, maddening Guillermo and all the friends in SSIG laboratory. They were patient with my mood. Antonio Carlos, finally the Peruvian is leaving.

I would like to thank the National Council for Scientific and Technological Development – CNPq (Grant 311053/2016-5), the Minas Gerais Research Foundation – FAPEMIG (Grants APQ-00567-14 and PPM-00540-17) and the Coordination for the Improvement of Higher Education Personnel – CAPES (DeepEyes Project).

Finally, besides the aforementioned people, I would like to thank all of those who has helped me complete my thesis no matter in whatever way.

Abstract

Modeling human behavior and activity patterns for detection of anomalous events has attracted significant research interest in recent years, particularly among the video surveillance community. An anomalous event might be characterized by the deviation from the normal or usual, but not necessarily in an undesirable manner. One of the main challenges of detecting such events is the difficulty to create models due to their unpredictability and their dependency on the context of the scene.

Anomalous events detection or anomaly recognition for surveillance videos is a very hard problem. Since anomalous events depend on the characteristic or the context of a specific scene. Although many contexts could be similar, the events that can be considered anomalous are also infinity, i.e., cannot be learned beforehand. In this dissertation, we propose three approaches to detect anomalous patterns in surveillance video sequences.

In the first approach, we present an approach based on a handcrafted feature descriptor that employs general concepts, such as orientation, velocity, and entropy to build a descriptor for spatiotemporal regions. With this histogram, we can compare them and detect anomalies in video sequences. The main advantage of this approach is its simplicity and promising results that will be show in the experimental results, where our descriptors had well performance in famous dataset as UCSD and Subway, reaching comparative results with the estate of the art, specially in UCSD peds2 view. This results show that this model fits well in scenes with crowds. In the second proposal, we develop an approach based on human-object interactions. This approach explores the scene context to determine normal patterns and finally detect whether some video segment contains a possible anomalous event. To validate this approach we proposed a novel dataset which contains anomalies based on the human object interactions, the results are promising, however, this approach must be extended to be robust to more situations and environments. In the third approach, we propose a novel method based on semantic information of people movement. While, most studies focus in information extracted from spatiotemporal regions, our approach detects anomalies

based on human trajectory. The results show that our model is suitable to detect anomalies in environments where trajectory of the people could be extracted.

The main difference among the proposed approaches is the source to describe the events in the scene. The first method intends to represent the scene from spatiotemporal regions, the second uses the human-object interactions and the third uses the people trajectory. Each approach is oriented to certain anomaly types, having advantages and disadvantages according to the inherent limitation of the source and to the subjective of normal and anomaly event definition in a determinate context.

List of Figures

1.1	Examples of classroom context. First row presents auditorium classrooms, second row contains examples of small classes. In all the images, many elements are similar as well as the people behavior.	5
2.1	The optical flow constraint equation defines a line velocity space.	15
2.2	Through apertures 1 and 3, only normal motions of the edges forming the square can be estimated, due to a lack of local structure. Inside aperture 2, at the corner point, the motion can be fully measured as there is sufficient local structure; both normal motions are visible.	16
2.3	Histogram composed by four bins, $B = 4$ [Chaudhry et al., 2009].	19
2.4	Classical CNN architecture proposed by Le Cun et al. [1990].	23
2.5	Alexnet architecture [Krizhevsky et al., 2012].	23
2.6	Example for unrolled Recurrent Neural Network (RNN) [Olah, 2018].	24
2.7	Unrolled Gated Recurrent Unit (GRU).	25
2.8	Gated Recurrent Unit (GRU).	26
2.9	Image representation of Autoencoder.	27
4.1	Diagram illustrating the proposed approach to performing anomalous event detection.	40
4.2	Image representation of a spatiotemporal region or cuboid C_i , which is composed of region R over c frames.	41
4.3	(a) and (b) are two consecutive frames. (c) is the frame difference between Frame $i - 1$ and Frame i , (d) is the optical flow representation, colored points represent the optical flow translation, green for previous frame, pink represents optical flow output. Then, image in (c) is a mask, thus, only white pixels this image are used to compute optical flow.	42

4.4	Example of feature vector extraction using orientation-magnitude descriptor. This figure shows a matrix presenting four magnitude ranges: $\{(0, 20], (20, 40], (40, 60], (60, \infty)\}$, named SR_1, SR_2, SR_3, SR_4 . All magnitudes are represented by colors blue, green, orange and red, respectively. Moreover, this figure also presents four ranges for orientations: $\{(0, 90], (90, 180], (180, 270], (270, 360]\}$, named as OC_1, OC_2, OC_3, OC_4 .	44
4.5	Matrix representation for four magnitude ranges: $\{(0, 20], (20, 40], (40, 60], (60, \infty)\}$, named SR_1, SR_2, SR_3, SR_4 . All magnitudes are represented by colors blue, green, orange and red, respectively. Moreover, this Figure also presents four ranges for orientations: $\{(0, 90], (90, 180], (180, 270], (270, 360]\}$, named as OC_1, OC_2, OC_3, OC_4 . Finally we can see the entropy as a third dimension based on four ranges also $\{(0, \frac{1}{2}], (\frac{1}{2}, 1], (1, \frac{3}{2}], (\frac{3}{2}, 2]\}$ labeled as $Epy_1, Epy_2, Epy_3, Epy_4$ respectively. Entropy value is between $[0, 2]$ given base of two.	46
4.6	Nearest neighbor search. Anomalous patterns is represented by point A, normal patterns is point B.	47
4.7	Results for Peds1.	51
4.8	Results for Peds2.	52
4.9	Examples of true positive matches for the exit gate clip.	53
4.10	Examples of true positive matches for the entrance gate clip.	54
4.11	Examples of false alarms samples for the subway clips. In first row two mistakes of position are presented, in second row the algorithm can recognize the anomaly, however after some time, in third the boy running which is anomalous, nonetheless it does not appear in the ground-truth.	55
4.12	ROC curve for exit sequence.	56
4.13	ROC curve for entrance sequence.	57
4.14	Examples analyzed through anomaly detection.	57
4.15	Results for Subway Entrance clip.	58
4.16	ROC curves for the Badminton dataset.	58
4.17	Examples of anomaly detections in the Badminton dataset.	59
5.1	Diagram illustrating the proposed approach to perform anomalous event detection using human-object interaction. The box in gray color indicates the steps for anomaly representation (training and testing) and the box in yellow indicates step for anomalous pattern detection (testing only).	62

5.2	Interaction representation from a person tracklet. Squares represent the person at determinate frame and circles represent the linked objects, different colors indicate different objects. Letters on top of structures are the word representation for this interaction. For instance, in Frame $i - 1$, there are two objects: A and B.	64
5.3	Graphical example of human-object interaction. It is a simple representation, where, there are only a single person with one object. The label for this interaction structure is [blank-chair-chair].	64
5.4	Examples analyzed through anomaly detection using human-object approach.	69
6.1	Overview of our approach. Given a body skeleton, we select reference points that are used to build trajectories. A sequence of such reference points consists of a trajectory. Then, we describe the normalized trajectories using two different techniques, a convolutional descriptor based on CNN or a recurrent descriptor modeled using a RNN. During the testing phase, we recognize anomalies and rare trajectories by comparing the descriptors extracted from each test sample regarding the trained model.	74
6.2	Score function examples. At left is the situation when the point p is near to point pr . At right is when point p is near enough to last point on tracklet and predicted point pr	77
6.3	Examples for point selection when the number of points have to be reduced. First row corresponds to the first derivative (number of points vs derivative value), circles show some interest points. In second row are marked the selected points in the frame, which for this case has a 1270×720 dimension.	78
6.4	Trajectory matrix representation, angular and radial respectively.	79
6.5	Architecture for convolutional autoencoder.	80
6.6	Architecture for recurrent autoencoder.	81
6.7	Experimental results and comparison with the state-of-the-art on the <i>Entrance</i> , <i>Exit</i> . (a) ROC results for Entrance clip; (b) results for the Exit clip.	85
6.8	ROC curves for Train sequence.	86
6.9	Example of normal cluster. which in this case contains 73 elements.	88
6.10	Example of anomalous cluster. which in this case contains 10 elements or trajectories.	89
6.11	Image visualization of normal and anomalous points using t-SNE. Purple points correspond to normal patterns, colored with yellow, green and blue correspond to anomaly patterns.	90

List of Tables

4.1	Anomaly detection AUC and ERR (%) results of HOFME on UCSD dataset. The results for [Li et al., 2014] was obtained from the original paper. . . .	51
4.2	Anomaly detection AUC and ERR (%) results of Subway dataset.	56
4.3	Precision Recall (P/R) results of Badminton dataset.	56
5.1	Precision and Recall results of human-object interaction dataset. Only unrecognized objects are present in these sequences.	70
5.2	Precision and Recall results of human-object interaction dataset. Only unknown sequences of interactions are presented in the clips.	70
6.1	AUC and ROC for Avenue sequences. Highlighted in bold, we present our best and worst result.	87
6.2	Clustering chart for Rare Trajectory Identification.	88

Contents

Acknowledgments	vii
Abstract	viii
List of Figures	x
List of Tables	xiii
1 Introduction	1
1.1 Research Motivation	4
1.2 Problem Description	6
1.3 Hypotheses	8
1.4 Goals	9
1.5 Contributions	10
1.6 Structure of the Text	11
2 Theoretical Background	12
2.1 Terminology	12
2.2 Optical Flow	14
2.3 Histogram of Optical Flow (HOF)	18
2.4 k-Nearest Neighbors (k-NN)	19
2.5 Affinity Propagation	20
2.6 Neural Networks	21
2.6.1 Convolutional Neural Network (CNN)	21
2.6.2 Recurrent Neural Network (RNN)	24
2.6.3 Autoencoder (AE)	26
3 Related Work	29
3.1 Feature Extraction	30

3.2	Learning Models	34
4	Anomaly Detection based on Handcrafted Feature Descriptor Ex- tracted from Spatiotemporal Regions	38
4.1	Overview	38
4.2	Proposed Approach	39
4.2.1	Descriptor Extraction	41
4.2.2	Detection of Anomalous Events	46
4.3	Experiments	47
4.3.1	Datasets	48
4.3.2	Parameter Setting	49
4.3.3	Evaluation on the UCSD Dataset	50
4.3.4	Evaluation on the Subway Dataset	52
4.3.5	Evaluation on Badminton Dataset	56
5	Anomaly Detection Based on Human-Object Interactions	60
5.1	Overview	60
5.2	Proposed Approach	61
5.3	Anomaly Representation	62
5.4	Anomalous Pattern Detection	64
5.4.1	Unrecognized Interactions	65
5.4.2	Sequence of Interactions	65
5.5	Experiments	66
5.5.1	Parameter setting.	67
5.5.2	Metrics.	67
5.5.3	Unrecognized Interactions	68
5.5.4	Sequence of Interactions	70
6	Anomaly Detection based on Trajectories	72
6.1	Overview	72
6.2	Proposed Approach	73
6.2.1	Pose Estimation	75
6.2.2	Trajectory Building	75
6.2.3	Feature Extraction	79
6.2.4	Anomaly Recognition and Rare Trajectory Identification	81
6.3	Experiments	82
6.3.1	General Settings	83
6.3.2	Anomaly Recognition	84

6.3.3	Rare Trajectory Identification	87
6.3.4	Discussion	89
7	Conclusions	91
7.0.1	Additional discuss	93
7.1	General Conclusions	95
7.2	Future Directions	96
	Bibliography	98

Acronyms

AE Autoencoder

AP Affinity Propagation

AUC Area Under Curve

ANN Artificial Neural Network

CAE Convolutional Auto Encoder

CAGR Compound Annual Growth Rate

CCTV Closed Circuit Television

CNN Convolutional Neural Network

CRF Conditional Random Field

CUHK Chinese University of Hong Kong

DNN Deep Neural Network

EER Equal Error Rate

FPR False Positive Rate

GAN Generative Adversarial Network

GRU Gated Recurrent Unit

GMM Gaussian Mixture Model

HMM Hidden Markov Model

HOF Histogram of Optical Flow

HOFM Histograms of Optical Flow Orientation and Magnitude

HOFME Histograms of Optical Flow Orientation, Magnitude and Entropy

HOG Histogram of Oriented Gradient

ILSVRC ImageNet Large Scale Visual Recognition Challenge

ISR Intelligence, Surveillance & Reconnaissance

IVS Intelligent Video Surveillance

IoT Internet of Things

KDT Kernel Dynamic Texture

k-NN k-Nearest Neighbors

LSTM Long Short-Term Memory

MBH Motion Boundary Histogram

MLP Multi-layer Neural Network

MM Markov Model

NN Neural Network

PCA Principal Component Analysis

UCSD University of California San Diego

UFMG Universidade Federal de Minas Gerais

ReLU Rectified Linear Unit

RNN Recurrent Neural Network

ROC Receiver Operating Characteristic

ROI Region of Interest

SCD Structural Context Descriptor

SHOF Selective Histogram of Optical Flow

SVM Support Vector Machine

STIP Space-Time Interest Points

TPR true Positive Rate

t-SNE t-Distributed Stochastic Neighbor Embedding

VA Video Analytics

Chapter 1

Introduction

In recent years, video surveillance systems have become very important due to heightened security concerns and low hardware costs. These type of systems are widely used in many applications such as nursing care institutions, law enforcement, building security, and traffic analysis. At the same time, the need of effective monitoring of public places such as airports, railway stations, shopping malls, crowded sport arenas, military installations is increasing [Popoola and Kejun Wang, 2012]. In view of that, many of these institutions are beginning to use video surveillance systems [Dautov et al., 2018]. Indeed, the more security is required, the better monitoring is needed.

Traditional surveillance systems have relied on network cameras monitored by a human operator that must be aware of the actions carried out by people who are in the camera's field of view. With the recent growth of the number of cameras to be analyzed, the efficiency and accuracy of human operators have reached a limit [Keval, 2006]. The need for several human operators and the difficulties in detecting events as they occur are the main difficulties faced in surveillance these days. Further, it is quite natural that human operators will not be able to continuously monitor the video footage due to fatigue. They also are not able to capture all the important content in the surveillance video due to the nature of human visual perception. This may cause them to miss the most informative content of the video, which eventually results in failures and holes in the surveillance system. Therefore, there is a great need for real-time automated systems that detect and locate suspicious behaviors and alert security agents. Hence, the rapid increase in the deployment of Closed Circuit Television (CCTV) systems and the challenges posed by direct human monitoring have led to a greater demand for computer algorithms that are able to process the video feeds to extract information of interest for human operators. In this way, detecting unusual or suspicious activities, uncommon behaviors, or irregular events in a scene can be seen as a primary objective of an automated video surveillance system.

A fundamental goal in intelligent video surveillance is to automatically detect

anomalous events in video streams [Vallejo et al., 2014]. However, the problem of anomaly detection is greatly open and research efforts are scattered not only in approaches, but also in the interpretation of the problem, assumptions and goals [Li et al., 2015]. Therefore, anomaly detection is a relevant problem, however it is not well defined [Del Giorno et al., 2016].

Qualifying an event as anomalous is largely subjective and depends on the intended application as well as the context. Hence, the scene context is highly related with the anomaly definition. The context involves the concept about a scene, for instance: “a small market in a neighborhood”. The concept about this scene regards the common elements and the behavior thus allowing to reason about them, for instance: people buying and selling products, cabinets, money, baskets, bags, people talking, among many elements that characterize the idea of the small supermarket. At the same time, the context bounds the elements within a scene, for instance, a small market in a neighborhood hardly has a cinema, many people, antibiotics, people running and jumping, vehicles, and an infinite list of things that particularly do not belong to these type of places.

The important characteristics of normal or anomalous activities are relative. For instance, an anomalous activity in one scenario may become normal in another [Hu et al., 2013]. For this reason, naming something anomalous is very difficult. We can label or measure how much anomalous is an observation based on its similarity with the given examples or given its compatibility with the model derived from the examples. Thus, the goal of anomaly detection system is not to analyze normal behavior, but to detect deviations from it [Duan et al., 2014]. This is an important aspect because it allows to differentiate the anomalous event detection with event recognition. Usually, event detection models intend to determine what type of event happens from a list of known events. In case of anomalous event recognition, the anomalies could be either previously known or could be unknown events.

Jiang et al. [2009] define anomaly detection as the identification of motion patterns that do not behave as the expected behavior. They also define anomaly as rare or infrequent behavior compared to all other behaviors. However, the identification of this concept requires semantic information and subjective knowledge regarding the scene and the expected behavior. Nonetheless, unknown patterns, in most cases, are very difficult to represent in automatic recognition models. Therefore, the modeling usually is built for the usual recurring patterns found in the scene and when there is no fitting to any usual pattern, one concludes a given event as anomalous. There are three common definitions and assumptions of anomalous event applied in research [Sodemann et al., 2012]: anomalous events occur infrequently in comparison to normal events, they have

significantly different characteristics from normal events and, they are events which have a specific meaning. Within the first category, anomalous events are events that are not sufficiently represented within the data available for modeling. These models require the availability of many data samples for modeling. Every event which is not to be flagged as anomalous must be well represented in the data used to create the model. Studies within the second category of assumptions define anomalous events as those which are significantly distinct from the normal events. A limitation of these models is the inability to detect anomalies which are not significantly distinct from normal events. The third category defines anomalous events *a priori* as specific meaningful actions or occurrences, like an event classification.

A common characteristic in anomalous event detection models is that the only information available are samples of the “normal” class. In general, anomaly detection settings cannot use traditional supervised approaches because it is impossible to find a sufficiently representative set of anomalies [Del Giorno et al., 2016]. The problem of determining an anomalous event is to define what is an anomaly in a given context. In this way, models must limit the scope of their methods to certain type of situations, without losing robustness. This challenging task increments its hardness as semantic information of anomaly is added.

Most studies are based on a typical pipeline which employs representations based on spatiotemporal features (low-level characteristics extracted from temporal regions) [Hasan et al., 2016a; Wang and Xu, 2016; Zhou et al., 2016; Cheng et al., 2015] followed by one-class classifier to determine whether an event is anomalous. These approaches model anomalies using characteristics such as velocity (magnitude, orientation), appearance, density and location. Nowadays, deep learning [Yuan et al., 2015] has become a hot topic, which learns features directly from video sequences. These studies [Xu et al., 2015; Feng et al., 2016], instead of using hand-crafted features to model event patterns, they based their models in discriminative feature representations of both appearance and motion patterns. However, all this information is not well-suited for solving the anomalous event detection problem since they are constrained to the same camera view, preventing the detection to be performed on different environments within the same context.

As aforementioned, common representations for anomaly event detection are based on low-level features, which fit well to recognize some particular types of anomalies, being mainly restrict to a single camera view. It means that the information captured by such features from a particular scene (i.e., same camera view) cannot be considered to detect anomaly in other scenes or even in different camera views within the same scene due to correlated information recorded from the same camera position.

Thus, since a more semantic information of anomalies is required to describe normal situations, these low-level characteristics are not enough to detect anomalies properly.

Given the previous limitation, we need other information type that allows to detect anomalies. For instance, consider the scenario in which the information of a specific context is repeated in many scenes. This information could capture the essence of a specific environment to be employed in similar environment, nevertheless at same time, this information must not depend on the camera position or be sensitive to the camera movement. For instance, in high-school classroom, we can find some common elements such as tables, chairs, a board, among others, also specific context activities happening such as walking, reading, writing, talking, etc. Figure 1.1 depicts examples of classroom context, however different scenes. On the other hand, other objects such as cars, shotguns and beds, or activities such as biking or fighting are not part of the high-school classroom context and could be characterized as an anomalous event. Other type of information that could be used to detect anomalies is based on the human trajectories. Many models focus on the movement information; however, they have problems about the illumination changes and the conception about the element that have been performed the anomaly. Introducing trajectory information implies a higher level of semantics. This can be beneficial since not all anomalies are detected using low-level characteristics.

In this study we address the anomalous event detection problem using three perspectives. First, we focus in common handcraft-feature representation in a more traditional approach (learning and detection of anomaly for a single camera view). Second, we propose a novel paradigm of anomaly detection based on human-object interactions able to work for different camera views being restrict only to be within the same context. Finally, the third approach detects anomalies based on the trajectory behavior of the people into the scene. Although this approach also is oriented from fixed view camera, the idea is to use information of more semantic level.

1.1 Research Motivation

There have been considerable efforts in the industry as well as academia, focusing on different algorithms, techniques and models to develop surveillance solutions [Chandola et al., 2009]. Following decades of slow market penetration and setbacks, the Intelligent Video Surveillance and Video Analytics industry is forecasted to experience decades of rapid growth. The Global Video Analytics, Intelligent Video Surveillance & Object Recognition Market-2015-2020 report indicates that the global Intelligent Video



(a)



(b)



(c)



(d)

Figure 1.1: Examples of classroom context. First row presents auditorium classrooms, second row contains examples of small classes. In all the images, many elements are similar as well as the people behavior.

Surveillance (IVS), Intelligence, Surveillance & Reconnaissance (ISR) and Video Analytics (VA) industry revenues are forecasted to grow at 14% Compound Annual Growth Rate (CAGR) from 2014 to 2020¹. Technology has introduced new tools such as: 4k videos, HD-CCTV cameras, more storage, new devices that allow demise pure server-based solutions, drones and integrated frameworks that join Internet of Things (IoT) with surveillance. In this context, anomalous event detection becomes an important and attractive topic to be studied and deepened.

Intelligent video surveillance using computer vision techniques is a popular research area and anomalous event detection is a sub-category of it, where the outcomes of this category can be used in various applications, including circumventing the security threats in public places and other day-to-day monitoring in elderly and patient care

¹MarketsandMarkets is the largest market research firm worldwide in terms of premium market research reports published annually. Serving 1,700 Fortune organizations globally with more than 1200 premium studies in a year, MarketsandMarkets caters to multitude of clients across 12 different industry verticals (www.marketsandmarkets.com).

centers. In recent years, a wealth of research has been undertaken in the domain of human behavior classification in automated surveillance. Behavior classification involves the categorization or classification of perceived behavioral events by an algorithm. Such research efforts have been driven by an increased concern for security and safety, coupled with an overabundance of available surveillance data relative to the amount of manpower available to process it.

Exploration of the unusual event detection in surveillance applications is based on two main arguments. First, the number of security cameras is growing, and the monitoring of these cameras is becoming increasingly difficult. Second, most of the solutions available to the public through commercialized products assume simple visual environments, and when more challenging environments are introduced, laborious and time intensive initialization procedures are required.

Though there have been lots of methods proposed to learn the activities present in the scene, there are still many challenges that need to be addressed, such as modeling various complex human activities, developing methods to model the activities not only in the constrained scenarios but also in the unconstrained scenarios. Apart from modeling, feature extraction also has a major role in the problem of anomalous event detection. As abnormality depends on the context under consideration, features must be carefully chosen to provide rich information about the normal activities present in a specific context. As there are many different contexts, extracting the right features for each and every context, which could provide abstract information relevant to the peculiar characteristics of the context, still remains a challenge. The above-mentioned reasons make video-based detection of normal and anomalous behavior of individuals a challenging task.

1.2 Problem Description

There are several problems related to the detection of anomalous events [Agrawal and Agrawal, 2015; Ahmed et al., 2016a; Akoglu et al., 2015; Ahmed et al., 2016b]. In this section we present the most representative for our research. For this it is important to delimit the scope of our work. Our study focuses on the detection of anomalous events in a context of surveillance videos.

One problem found in anomalous event detection is that the analysis of surveillance data is performed without the knowledge of when, where or even if an interesting event has occurred or is occurring. In this type of analysis, the analyst is interested in extraordinary events, something that deviates from the normal. Therefore, without

the suitable tools, it can be a daunting task for the analyst, consisting of sequentially viewing all raw video data and using his/her judgment to determine whether an event is unusual.

Another problem found in anomalous event detection is to determine what are the events which can be labeled as anomalous. In the literature, there have been a variety of terms used to refer to abnormal or anomalous events including interesting, irregular, suspicious, anomaly, uncommon, unusual, rare, atypical, salient and outliers. The definition of anomalous events has been causing much debate in the literature due to the subjective nature and complexity of human behaviors [Jiang et al., 2009]. In particular, an event is considered abnormal if there is deviation from observed or learned ordinary events (i.e., the event having low occurrence or statistical representation in the learned model) or the event is not known or it is outstanding. As aforementioned, another important aspect about the definition of anomalies is the context of the scene, which is directly related to the definition of anomaly in a certain environment or situation. In addition, the difficulty of semantic anomalous events increases as the semantics of events grows. In other words, the greater the semantics of an event, more complex are the situations that must need described to determine the anomalous events of the normal. Similarly, there is no clear distinction between abnormal activities, events and behaviors as their descriptions often overlap one another. In this study, we have opted for using the term anomalous event because abnormal might refer to a unusual event in a way that is undesirable, which is not our case since we do not capture enough semantic understand whether a given event is suspicious or just different from a normal recurring pattern.

To determine whether an event is anomalous or not, there are two main options: (i) the expected types of anomalies are known, (ii) only the normal patterns are known. In the first option, approaches have two classes, normal and abnormal. Usually, this problem is solved with action/activity recognition approaches [Popoola and Kejun Wang, 2012]. In the second option, there is only one known class. The main challenge of this latter option is to define features that allowing the detection of anomalous events and at the same time being suitable in many situations [Sodemann et al., 2012].

A representative characteristic of the anomalies is the difficulty in representing them. Once the model has been chosen, the goal is to find the outliers that represent the anomalies [Akoglu et al., 2015]. The problem is that the outliers do not have a specific spatial distribution and can be scattered in the space generated by the descriptor. It is natural that this happens since the spatial distribution depends on the descriptor and not necessarily the points that represent the anomalies must be together.

It is difficult to list all the normal situations and, depending on the context, it is even more complex to list the anomalous events. In other words, anomalous events are generally unknown so their position in the space generated by the descriptors is also unknown. There is a need for a prior analysis for each context, this must include anomalous situations to be able to determine if the descriptor is managing to represent the events, however, it is not possible to consider all the possible anomalous events, or in the case If necessary, it is a truly exhaustive task. Therefore, it is necessary to delimit the scope of the methods, which is also complex given the context and the final use in the field of surveillance videos.

Compared to other branches of research, the number of datasets for anomaly recognition is small. And in many cases the observation time of normal situations is short. This is controversial, because in surveillance videos, probably many normal situations are constantly repeated, however the recognition of what is normal can be confused with some abnormal situation. Apart from that, as already mentioned, the anomalies are unknown so it is difficult to define an observation time necessary to recognize all the normal events of a determined scene. This is again inherent in the context and therefore can only be configured according to the objective of surveillance in that particular scene.

1.3 Hypotheses

Before presenting our hypotheses, we present our main assumption which is: “*Something that has not being seen before is considered anomalous*”. This assumption is important because there are infinite situations that could be defined as anomalies. Hence, to delimit the scope of anomalies and at the same time give robustness to our model, we consider as normal any situation that is presented during training phase. Likewise, the model must secure that something that happening in training observations are normal events. In the following paragraphs, we present our hypotheses for the proposed approaches.

In our handcrafted approach, we define four characteristics to be used as clues to describe normal motion patterns in a particular region of the scene: i) velocity - speed of moving objects; ii) orientation - common flow of the objects; iii) appearance-texture of the objects; and (iv) density-number of moving objects. Hence, we hypothesize:

Hypothesis 1 (H1): *the use of low-level features extracted for movement patterns may characterize anomalies, thus, an histogram representation may represent the move-*

ment information, including also contextual information like orientation entropy extracted from optical flow vectors.

In second approach, human-object interactions, the main idea is to learn information regarding the context in a given environment and use that to detect anomalies in other scenes belonging to the same context. We hypothesize that

Hypothesis 2 (H2): *the human-object interactions might lead to the understanding of the scene. In consequence, normal patterns could be defined from the interaction representation.*

Furthermore, there are common patterns between scenes. Hence, we hypothesize

Hypothesis 3 (H3): *patterns learned in some scene might be used in another scene that shares the same context. This semantic information may help to detect anomalous patterns.*

In trajectory based approach we hypothesize:

Hypothesis 4 (H4): *human trajectories could provide relevant information to understand the movement behavior of the people in the scene. Thus, trajectory description characterizes the movement patterns normal and abnormal, at the same time they provide contextual information such as velocity, flow, direction and location, that could be used for behavior analysis.*

1.4 Goals

This Ph.D dissertation presents three approaches that aim to detect anomalous events in surveillance videos. The first, based on handcrafted features, attempts to recognize anomalies in scenes that are far from camera. This model has as goal to describe patterns based on movements to determine in which frames an anomaly is happening. The second, based on human-object interaction, collects interactions of a person and his/her surrounding objects to detect anomalous events. Finally, in our third approach, based on trajectories, the main goal is to describe the human trajectories in the video to find movement patterns that can represent the anomalous events.

The following specific goals are defined:

- Propose feature descriptors that employ optical flow information to capture movement information from spatiotemporal regions.

- Describe human-object interactions to discriminate between normal and anomalous patterns.
- Propose feature descriptors that explain the peculiar characteristics of the human trajectory patterns in a scene.
- Evaluate, validate and compare the proposed approaches.

1.5 Contributions

We propose three approaches for anomalous event detection. The first based on low-level (handcrafted) features, the second based on human-object interactions and the third a human trajectory based approach. The contributions of this proposal thesis are as follows.

- The use of low-level information regarding speed and orientation to describe the scene to determine whether a spatiotemporal region can be labeled as normal or anomalous [Colque et al., 2015], which has been extended in the work [Colque et al., 2017] with the addition of the entropy information. This model is oriented scenes with fixed camera.
- The use of human-object interactions to capture anomalies that take place in the same context (does not requiring the camera to be static of the scene been recorded in the same environment) [Colque et al., 2018].
- A simple model for multi-tracking of people in surveillance videos. This heuristic is an alternative to the complex data association models.
- The use of trajectory information to detect anomalous events based on people movement into the scene. In this approach, a higher semantic level is used to detect the anomalies. An advantage of this model over traditional methods, is that the information extracted can be used for a deep analysis of the scene. This analysis is focused on movement patterns that can be extracted from the set of trajectories formed by the people in the scene. This approach is oriented for fixed camera view.
- A brief analysis of trajectory charity which intuitively allows us to know the behavior of the costume of the people in the scene.

- Introduction of two new video dataset and a video sequence to contribute with literature. These datasets propose simple situations for the analysis of anomalous events, but at the same time they offer an alternative to the current datasets in the literature. In the case of the Laboratory dataset, the content is real without forcing any anomalous situation.

This research has the following published papers as contribution to literature:

- Rensso Mora Colque, Carlos Antonio Caetano Junior, William Robson Schwartz: “*Histograms of Optical Flow Orientation and Magnitude to Detect Anomalous Events in Videos*”. SIBGRAPI 2015: 126-133.
- Rensso Mora Colque; Carlos Caetano; Matheus Toledo; William R. Schwartz: “*Histograms of Optical Flow Orientation and Magnitude and Entropy to Detect Anomalous Events in Videos*,” in IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 3, pp. 673-682, 2017.
- Rensso Mora Colque; Carlos Caetano; Victor H. C. de Melo, Guillermo Camara Chavez; William R. Schwartz: “*Novel Anomalous Event Detection Based on Human-object Interactions*”. in VISAPP 2018: 293-300.

Other papers published during the PhD course:

- Rensso Mora Colque, Guillermo Camara Chavez, William Robson Schwartz: “*Detection of Groups of People in Surveillance Videos Based on Spatio-Temporal Clues*”. CIARP 2014: 948-955.

1.6 Structure of the Text

This dissertation is structured in the following manner. Chapter 3 contains a literature review concerning relevant topics and studies. It is followed by Chapter 4 which discusses the proposed approach based on handcrafted low-level feature descriptors. Chapter 5 describes our proposed approach for multiple view and same context based on human-object interactions. In Chapter 6 we describe our third approach based on human trajectories, then followed by Chapter ?? which discusses the results while keeping the original goals in mind. Chapter 7 presents the conclusions of our complete study.

Chapter 2

Theoretical Background

In this chapter we present important concepts and techniques that aims to clarify our proposed approaches. This should provide enough information to make the design of the approaches. Further, the section that refers to neural networks is deeper explained than the other concepts, due to two of our three approaches are based on neural networks.

2.1 Terminology

Some of the terms that are frequently used throughout the thesis are described below.

Activity refers to a sequence of atomic actions performed by a subject. An action, on the other hand, implies a sequence of fundamental movements performed by an object or person.

Event is the occurrence of an activity in a particular time and place.

Context is the semantic of a scene. This concept includes the subjectiveness of the observer and the broadly perception of the situation.

Anomalies are a deviation from the common rule, type, arrangement or form, where, the “anomalous” word is its adjective.

Anomalous event is a unusual, odd, out of the ordinary, peculiar, unexpected type of event. In technical sense, an anomalous event means the patterns of action that do not conform the normal behaviors that have been learned or expected in a given context. An event that is anomalous at a certain context may be perfectly normal in another scene. Anomalous events are generally infrequent, sparse, and unpredictable [Li et al., 2014]. In the literature, this term has been referred to as: unusual events Zhao et al. [2011], anomalous events [Jiang et al., 2011], abnormality [Xiang and Gong, 2005], suspicious activities or irregularities [Boiman and Irani, 2005].

To clarify the previous concepts, imagine the following scene “*an athlete running in a marathon, which is reaching the goal at first place*”. In this example, the activity is running and it is performed by the athlete, the event is reaching the goal and the context is the conception of the view at the goal, it means, the people waiting for the winner, cameras, the goal, people running, etc. The concept of context so important because it is based on the context that normal and anomalous situations are defined. Hence, anomalies are dependent of a given context. For instance, an anomaly could be a car traveling at great speed against the marathoners. Finally, anomalies are not necessarily hazardous situations, anomalies could be anything depending on the concept of normal in the determinate context.

Anomalous event detection refers to the problem of finding patterns in data that do not conform to expected behavior. These non-conforming patterns are often referred to as anomalies, outliers, discordant observations, exceptions, aberrations, surprises, peculiarities or contaminants in different application domains [Chandola et al., 2009]. Mathematically, outliers are also referred to as abnormalities, discordants, deviants, or anomalies in the data mining and statistics literature [Aggarwal, 2013].

Types of Anomalies Depending on the context and which anomalous events are modeled, types of anomalies might be divided in [Chandola et al., 2009]:

- *Point anomalies* indicate that the values of extracted features at a specific location deviate significantly from what is considered normal. Therefore, these anomalies do not consider past values or the information given by nearby objects or points. If one models the normal velocities of moving objects at all locations in the scene, any object that displays a velocity that does not fit the model can be considered an anomaly. This includes, for example, detecting motion of objects at unusual locations.
- *Contextual anomalies* consider information from the temporal context (the sequence of events), or the spatial context (nearby objects). Anomalies that take into account the temporal context, also called sequential anomalies, analyze irregularities in the temporal sequence of a given extracted feature. An example for temporal concept is when person jumps a ticket gate to avoid the payment, the normal sequence of events indicate that person should walk not jump. An example for spatial context could be a person manipulating some forbidden objects in a museum during visiting hours.
- *Collective anomalies* refer to collections of samples which are anomalous altogether. For example, people running in a main street city could be considered

as normal at morning when is late for work. However, all people in the street running rapidly in one direction for fire is an anomalous situation in a normal day in a main street context.

Spatiotemporal regions, also known as “cuboids” [Kratz and Nishino, 2009], are fixed regions in an video, the 3D conception of cube is given by the depth in time from consecutive frames.

Scene actor is an individual or object that performs a certain action.

2.2 Optical Flow

Optic flow is defined as the change of structured light in the image, e.g. on the retina or the sensor of a camera, due to a relative motion between the eyeball or camera and the scene.

The initial hypothesis in measuring image motion is that the intensity structures of local time-varying image regions are approximately constant under motion for at least a short duration [Horn and Schunck, 1981]. Formally, if $I(X, t)$ is the image intensity function, then:

$$I(X, t) = I(X + \delta X, t + \delta t) \quad (2.1)$$

where δX is the displacement of the local image region at (X, t) after time δt . Expanding the left-hand side of this equation in a Taylor series yields

$$I(X, t) = I(X, t) + \nabla I \delta X + \delta t I_t + O^2, \quad (2.2)$$

where $\nabla I = (I_x, I_y)$ and I_t are the first order partial derivatives of $I(X, t)$, and O^2 , the second and higher order terms, which are assumed negligible. Subtracting $I(X, t)$ on both sides, ignoring O^2 and dividing by δt yields

$$\nabla I \cdot V I_t = 0, \quad (2.3)$$

where $\nabla I = (I_x, I_y)$ is the spatial intensity gradient and $V = (u, v)$ is the image velocity. Equation 2.3 is known as the optical flow constraint equation, and defines a single local constraint on image motion (see Figure 2.1). In the figure the normal velocity v_1 is defined as the vector perpendicular to the constraint line, that is, the velocity with the smallest magnitude on the optical flow constraint line. This constraint is not sufficient to compute both components of V as the optical flow constraint equation is illposed.

That is to say, only v_1 , the motion component in the direction of the local gradient of the image intensity function, may be estimated. This phenomenon is known as the aperture problem. and only at image locations where there is sufficient intensity structure (or Gaussian curvature) can the motion be fully estimated with the use of the optical flow constraint equation (See Figure 2.2). For example, the velocity of a surface that is homogeneous or containing texture with a single orientation can not be recovered optically. Because the normal velocity is in the direction of the spatial gradient ∇I , Equation 2.3 permit to write

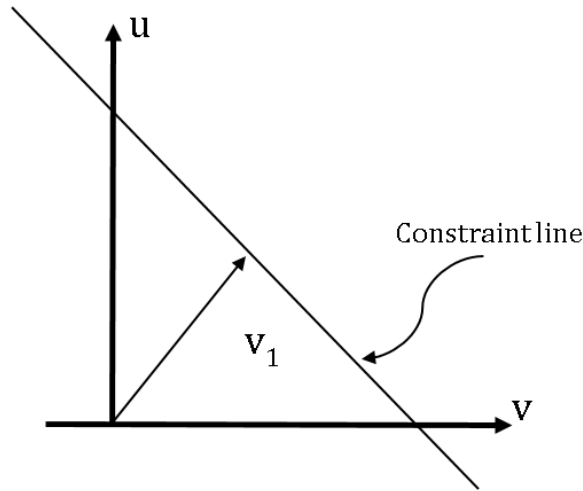


Figure 2.1: The optical flow constraint equation defines a line velocity space.

$$V_1 = \frac{-I_t \nabla I}{\|\nabla I\|_2^2}. \quad (2.4)$$

Thus, the measurement of spatiotemporal derivatives allows the recovery of normal image velocity.

From this definition, it becomes clear that for optical flow to be exactly image motion, a number of conditions have to be satisfied. These are: a) uniform illumination; b) Lambertian surface reflectance, and c) pure translation parallel to the image plane. Realistically, these conditions are never entirely satisfied in scenery.. Instead, it is assumed that these conditions hold locally in the scene and, therefore, locally the image on the image plane. The degree to which these conditions are satisfied partly determines the accuracy with which optical flow approximates image motion. Alternatively, the displacement of a small image patch can be measured, for instance using correlation in short images sequences (usually two or three frames). Such image displacements constitute a valuable approximation to image velocity when certain conditions are met. In particular, the ratio of sensor translational speed to absolute environmental

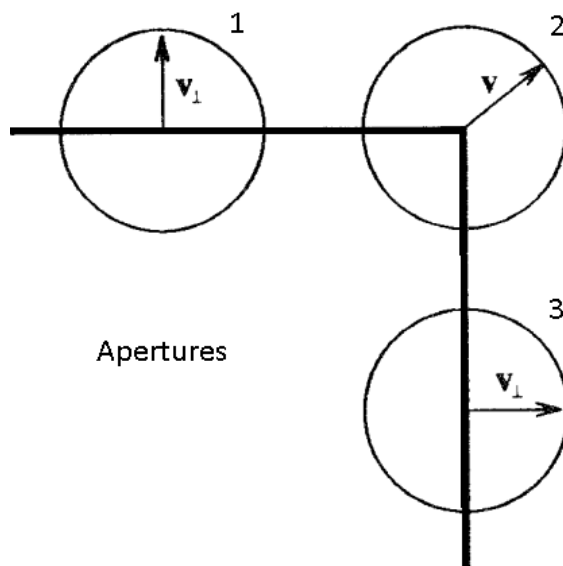


Figure 2.2: Through apertures 1 and 3, only normal motions of the edges forming the square can be estimated, due to a lack of local structure. Inside aperture 2, at the corner point, the motion can be fully measured as there is sufficient local structure; both normal motions are visible.

depth, the 3D vertical and horizontal sensor rotations, and the time interval between frames must be small quantities. Optical flow may also be computed as the disparity field where, given two stereo images or two adjacent images in some sequence, features of interest in the images are extracted and matched via a correspondence process.

Essentially, performing 2D motion detection involves the processing of scenes where the sensor is moving within an environment containing both stationary and non-stationary objects. Furthermore, visual events such as occlusion, transparent motions, and nonrigid objects increase the inherent complexity of the measurement of optical flow.

2.2.0.1 Lucas-Kanade method

In computer vision, the Lucas-Kanade method is a widely used differential method for optical flow estimation [Lucas and Kanade, 1981]. It assumes that the flow is essentially constant in a local neighborhood of the pixel under consideration, and solves the basic optical flow equations for all the pixels in that neighborhood, by the least squares criterion.

The Lucas-Kanade method assumes that the displacement of the image contents between two nearby instants (frames) is small and approximately constant within a neighborhood of the point p under consideration. Thus the optical flow equation can

be assumed to hold for all pixels within a window centered at p . Namely, the local image flow (velocity) vector (V_x, V_y) must satisfy

$$\begin{aligned} I_x(q_1)V_x + I_y(q_1)V_y &= -I_t(q_1) \\ I_x(q_2)V_x + I_y(q_2)V_y &= -I_t(q_2) \\ &\dots \\ I_x(q_n)V_x + I_y(q_n)V_y &= -I_t(q_n) \end{aligned}$$

where q_1, q_2, \dots, q_n are the pixels inside the window, and $I_x(q_i), I_y(q_i), I_t(q_i)$ are the partial derivatives of the image I with respect to position x, y and time t , evaluated at the point q_i and at the current time.

These equations can be written in matrix form $Av = b$ where

$$\begin{aligned} A &= \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_2) & I_y(q_2) \\ \dots & \dots \\ I_x(q_n) & I_y(q_n) \end{bmatrix} \\ v &= \begin{bmatrix} V_x \\ V_y \end{bmatrix} \\ b &= \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \dots - I_t(q_n) \end{bmatrix} \end{aligned}$$

This system has more equations than unknowns and thus it is usually over-determined. The Lucas-Kanade method obtains a compromise solution by the least squares principle. Namely, it solves the 2×2 system: $A^T Av = A^T b$ or $v = (A^T A)^{-1} A^T b$, where A^T is the transpose of matrix A . That is, it computes

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} \begin{bmatrix} \sum_i I_x(q_i)^2 & \sum_i I_x(q_i)I_y(q_i) \\ \sum_i I_y(q_i)I_x(q_i) & \sum_i I_y(q_i)^2 \end{bmatrix} \begin{bmatrix} -\sum I_x(q_i)I_t(q_i) \\ -\sum I_y(q_i)I_t(q_i) \end{bmatrix}$$

where the central matrix in the equation is an Inverse matrix. The sums are running from $i = 1$ to n .

The matrix $A^T A$ is often called the structure tensor of the image at the point p .

The plain least squares solution above gives the same importance to all n pixels q_i in the window. In practice it is usually better to give more weight to the pixels that are closer to the central pixel p . For that, one uses the weighted version of the least

squares equation, $A^T W A v = A^T W b$ or $v = (A^T W A)^{-1} A^T W b$, where W is an $n \times n$ diagonal matrix containing the weights $W_{ii} = w_i$ to be assigned to the equation of pixel q_i , follows

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} \begin{bmatrix} \sum_i w_i I_x(q_i)^2 & \sum_i w_i I_x(q_i) I_y(q_i) \\ \sum_i w_i I_y(q_i) I_x(q_i) & \sum_i w_i I_y(q_i)^2 \end{bmatrix} \begin{bmatrix} -\sum w_i I_x(q_i) I_t(q_i) \\ -\sum w_i I_y(q_i) I_t(q_i) \end{bmatrix}$$

The weight w_i is usually set a Gaussian function of the distance between q_i and p .

In order for equation $A^T A v = A^T b$ to be solvable, $A^T A$ should be invertible, or $A^T A$'s eigenvalues satisfy $\lambda_1 \geq \lambda_2 > 0$. To avoid noise issue, usually λ_2 is required to not be too small. Also, if $\frac{\lambda_1}{\lambda_2}$ is too large, this means that the point p is on an edge, and this method suffers from the aperture problem. So for this method to work properly, the condition is that λ_1 and λ_2 are large enough and have similar magnitude. This condition is also the one for corner detection. This observation shows that one can easily tell which pixel is suitable for the Lucas-Kanade method to work on by inspecting a single image.

One main assumption for this method is that the motion is small (less than one pixel between two images for example). If the motion is large and violates this assumption, one technique is to reduce the resolution of images first and then apply the Lucas-Kanade method.

The Lucas-Kanade method per se can be used only when the image flow vector V_x, V_y between the two frames is small enough for the differential equation of the optical flow to hold, which is often less than the pixel spacing. When the flow vector may exceed this limit, such as in stereo matching or warped document registration, the Lucas-Kanade method may still be used to refine some coarse estimate of the same, obtained by other means; for example, by extrapolating the flow vectors computed for previous frames, or by running the Lucas-Kanade algorithm on reduced-scale versions of the images. Indeed, the latter method is the basis of the popular Kanade-Lucas-Tomasi (KLT) feature matching algorithm.

2.3 Histogram of Optical Flow (HOF)

Histogram of Optical Flow (HOF) is a descriptor based on optical flow information. Proposed by Chaudhry et al. [2009], this descriptor stores the optical flow information (orientation and magnitude) distributing it in a histogram $h = [h_1, h_2, \dots, h_B]$, which

is composed of B bins. Thus, each flow vector is binned according to its primary angle from the horizontal axis and incremented according to its magnitude. Thus, every optical flow vector, $v = (x, y)'$, with direction $\theta = \tan^{-1}(\frac{y}{x})$, that contributes with its magnitude $m = \sqrt{x^2 + y^2}$ to the i -th bin of the histogram, where $1 \leq i \leq B$, B value is the quantized range. Figure 2.3 illustrates the procedure, where angle α contributes with its corresponding bin adding the magnitude value.

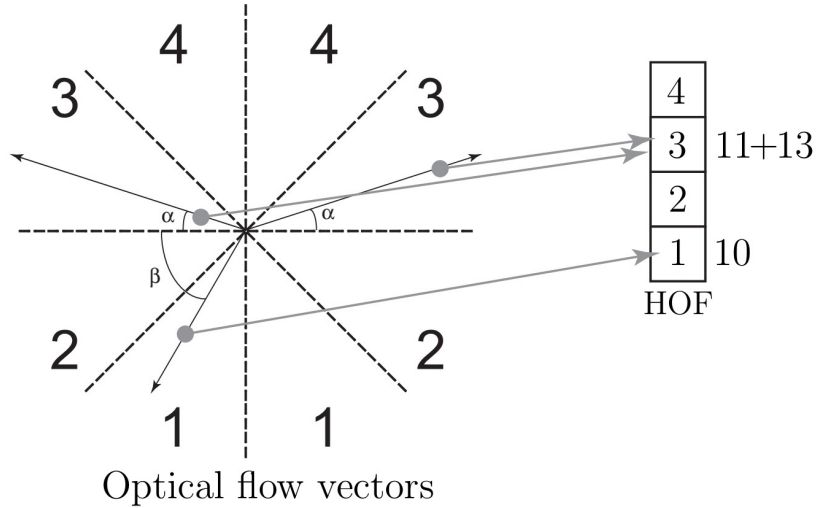


Figure 2.3: Histogram composed by four bins, $B = 4$ [Chaudhry et al., 2009].

2.4 k-Nearest Neighbors (k-NN)

k-Nearest Neighbors (k-NN) is a non-parametric and data-driven algorithm used for multi-class classification. Given a testing data, k-NN seeks to find out its k nearest neighbors from the entire labeled training set with a pre-defined distance metric [Larose, 2004]. The testing data is classified by a majority voting of its k nearest neighbors. In other words, the testing data is assigned to the dominant class among its k nearest neighbors. The special case is $k = 1$, where the testing data is simply assigned to the class of its nearest neighbor.

The advantages of k-NN are twofold. First, it is easy to implement as in the training phase the only step is storing the training set and their class labels. Second, it is flexible to handle diverse data by using specific distance metrics. However, it is computationally intensive when training set is very large because it must identify the k nearest neighbor for each testing data.

2.5 Affinity Propagation

Afinity Propagation (AP) clustering [Frey and Dueck, 2007] is a fast clustering algorithm especially in the case of large number of clusters. AP works based on similarities between pairs of data points (or $n \times n$ similarity matrix S for n data points) and simultaneously considers all the data points as potential cluster centers (called exemplars).

In the AP clustering algorithm, there are two important concepts: the responsibility $R(i, k)$ and availability $A(i, k)$ which represent two messages indicating how well-suited a data point is to be a potential exemplar. $R(i, k)$ is an accumulated value which reflects how well the point i is suited to be the candidate exemplar of data point i and then sends from the latter to the former; that is, compared to other potential exemplars, the point k is the best exemplar. The availability $A(i, k)$ is opposed to $R(i, k)$ and reflects how well-suited it is for the point i to choose point k as its exemplars. Based on the candidate exemplar point k , the accumulated message sent to the data point i indicating it that point k is more qualified as an exemplar than others.

The sum of the values of $R(i, k)$ and $A(i, k)$ is the evaluation basis whether the corresponding data point can be a candidate exemplar or not. Once a data point is chosen to be a candidate exemplar, those other data points with nearer distance will be assigned to this cluster. The similarity value between two data points x_i and x_j ($i \neq j$) is usually assigned the negative Euclidean distance, such as $S(i, j) = -\|x_i - x_j\|^2$. The algorithm uses an initial value called preference, which indicates the preference that the data point can be chosen as an exemplar, it is usually set by the median(s) of all distances. The following Algorithm 1 summarizes the process:

Algorithm 1 Afinity Propagation (AP).

```

1: procedure CLUSTERINGAP( $S$ )
2:    $R(i, k) = 0, A(i, j) = 0, \forall i, k$ 
3:   while Until converge do
4:      $R(i, k) = S(i, k) - \max(A(i, j) + S(i, j)) \mid (j \in [1, n]; j \neq k)$ 
5:      $A(i, k) = \min(0, R(k, k) + \sum_j \max(0, R(j, k))), \mid (j \in [1, n]; j \neq i; j \neq k)$ 
6:      $A(k, k) = \sum_i \max(0, R(i, k)), \mid (i \neq k)$ 
7:   return  $Trks$ 

```

The algorithm iterates until either the cluster boundaries remain unchanged over a number of iterations or after some predetermined number of iterations. The exemplars are extracted from the final matrices as those whose “responsibility + availability” for themselves is positive.

2.6 Neural Networks

Following the Caudill [1987] definition, Neural Network (NN) is “a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs”.

In simplest form a NN is expressed as linear regression equation $\hat{Y} = Wx + b$. The matrix W represents the weights of the neural model, x is the input data and b is the bias, this element usually is included in matrix W . The goal of the learning is to find a W matrix that allow to predict output \hat{Y} in order to increase the likelihood between output and real label Y . This formula corresponds to a single layer representation. In general architectures present many types of layers. A very intuitive form to visualize a NN is in a graph structure, where each layer contains neurons, and each neuron can be seen as the simple linear regression formula.

A famous algorithm to learn the weights is the backpropagation. Based on feed-forward strategy, this algorithm updates the weights iteratively. In forward step the result of activation function scrutinizes inputs and current weights and based on a loss function that compares the output of activation functions with the label assign to the input. In forward step the weights are updated by the Gradient Descent algorithm. This straightforward process is the core of many machine learning models based on NN. In this section we present a briefly explanation of the networks that have been used in our approaches [Caudill, 1987].

2.6.1 Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a category of neural networks that has proven very effective in areas such as image recognition and classification. CNNs have been successful in identifying faces, objects and traffic signs apart from powering vision in robots and self driving cars. CNNs have a different architecture than regular NN. Regular NNs transform an input by putting it through a series of hidden layers. Every layer is made up of a set of neurons, where each layer is fully connected to all neurons in the layer before. Finally, there is a completely connected last layer, the output layer, which represents the predictions. The CNNs are different. First, the layers are organized in three dimensions: width, height and depth. In addition, the neurons in one layer do not connect to all the neurons in the next layer, but only to a small region of the same. Finally, the final result will be reduced to a single vector of probability scores, organized along the depth dimension. CNNs have two components: feature extraction and classification steps.

The hidden layers or feature extraction layers perform a series of convolutions and pooling operations. The primary purpose of Convolution is to extract features from the input image. Convolution preserves the spatial relationship between pixels by learning image features using small squares of input data. Pooling operations aim at scaling the convolution output to represent global information. The matrix formed by sliding the filter over the image and computing the dot product is called activation map or feature map. It is important to note that filters act as feature detectors from the original input image.

In the classification part, the output from the convolutional and pooling layers represent high-level features of the input image. The purpose of the fully connected layer is to use these features for classifying the input image into various classes based on the training dataset. The fully connected layer is a traditional multilayer perceptron that uses a softmax activation function in the output layer, other models utilize SVM. The term “Fully Connected” implies that every neuron in the previous layer is connected to every neuron on the next layer. The size of the feature map is controlled by three parameters: depth, stride and padding. Depth corresponds to the number of filters for the convolution operation. Stride is the number of pixels by which the technique slides the filter matrix over the input matrix. When the stride is 1 then algorithm slides the filters one pixel at a time. When the stride is 2, then the filters jump 2 pixels at a time as algorithm slides them around. Having a larger stride will produce smaller feature maps. Padding adds zeros around the image border. Adding zero-padding is also called wide convolution, and not using zero-padding would be a narrow convolution.

The pioneering work with CNN is LeNet [Le Cun et al., 1990]. That net was used to classify digits and it was applied by several banks to recognize hand-written numbers on digitized checks in 32×32 pixel greyscale input images. The architecture for this convolutional neural network employs sequence of three layers of convolution and pooling. This architecture is shown in Figure 2.4. The convolution layers extract spatial features, pooling layers subsample using spatial average of maps. Multi-layer Neural Network (MLP) is the final classifier. The ability to process higher resolution images requires larger and more convolutional layers, so this technique is constrained by the availability of computing resources. During some years, CNNs and neural networks went unnoticed, until computing power rose, CPUs were becoming faster, and GPUs became a general-purpose computing tool. Specifically, CNN reappeared in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The ImageNet project is a large visual database designed for use in visual object recognition software research. In this contest, models compete to correctly classify and detect objects and scenes. In 2012, AlexNet [Krizhevsky et al., 2012] significantly outperformed all

the prior competitors and won the challenge by reducing the top-5 error from 26% to 15.3%. The network had a very similar architecture as LeNet by Yann LeCun et al but it was deeper, with more filters per layer, and with stacked convolutional layers. It consisted 11×11 , 5×5 , 3×3 , convolutions, max pooling, dropout, data augmentation, Rectified Linear Unit (ReLU) activations, SGD with momentum. It attached ReLU activations after every convolutional and fully-connected layer. Figure 2.5 presents this architecture.

In past years, other important architectures appear, improving drawbacks from predecessors, for instance: VGG16 proposed by [Simonyan and Zisserman, 2014], GoogleNet [Szegedy et al., 2015], Microsoft Res-Net [He et al., 2016], focusing mainly on image classification. However, novel approaches focus on object detection which comprises a high difficult level. In this group we can mention: Faster R-CNN [Ren et al., 2017], Yolo9000 [Redmon and Farhadi, 2017], Single Shot Detector (SSD) [Liu et al., 2016] and Feature Pyramid Network [Lin et al., 2017]. All these models show the the effectiveness of CNNs and the architectures are also utilized in many other studies [Li Yandong, 2016], including anomaly and crowd behavior analysis approaches [Tripathi et al., 2018].

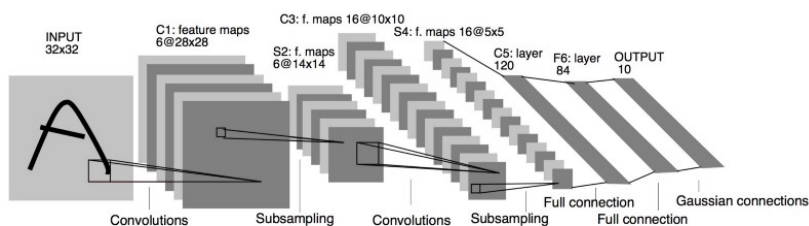


Figure 2.4: Clasiical CNN architecture proposed by Le Cun et al. [1990].

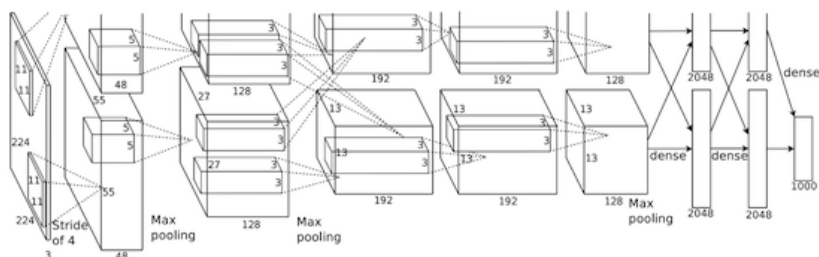


Figure 2.5: Alexnet architecture [Krizhevsky et al., 2012].

2.6.2 Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) is a powerful and robust type of neural network and is employed in many promising algorithms because they are the only NN type with an internal memory. RNNs are able to remember important information about the input that they received, which enables them to be very precise in predicting next information states. These models are well suited for sequential data such as time series, speech, text, financial data, audio, video and weather. An particular advantage is that they can form a much deeper understanding of a sequence and its context, compared to other algorithms.

In a RNN, the information feeds cells through a loop. When it makes a decision, it takes into consideration the current input and also what it has learned from the inputs it received previously. A usual RNN has a short-term memory. Thus, recurrent neural network has two inputs, the present and the recent past. This is important because the sequence of data contains crucial information about future information. This NN assigns, as all other deep learning algorithms, a weight matrix to its inputs and then produces the output. Note that RNNs apply weights to the current and also to the previous input. Furthermore, they also update their weights for both: through gradient descent and backpropagation through time, this latter term is basically back-propagation algorithm on an unrolled RNN. Figure 2.6 presents an example of unrolled RNN. In the left, you can see the RNN, which is unrolled after the equal sign. Note that there is no cycle after the equal sign since the different time steps are visualized and information is passed from one time step to the next. Classical RNN deals with two major problems: exploding and vanish gradient. The first problem refers when algorithm assigns high importance to the weights, without much reason. The second issue occurs when the values of a gradient are too small and the model stops learning or takes way too long because of that. This was a major problem in the 1990s decade and much harder to solve than the exploding gradients. Fortunately, it was solved by Hochreiter and Schmidhuber [1997], whose introduced the Long Short-Term Memory (LSTM).

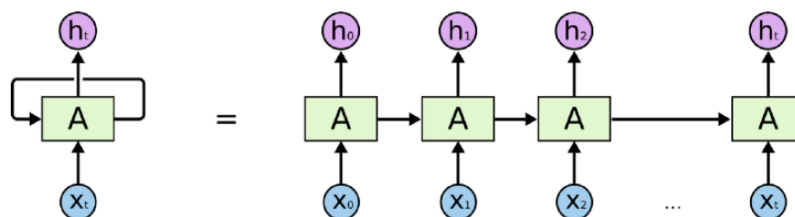


Figure 2.6: Example for unrolled Recurrent Neural Network (RNN) [Olah, 2018].

LSTM is special kind of RNN, capable of learning long-term dependencies. In standard RNNs, the main unit is composed by a simple structure, such as a single tanh layer. However, LSTM introduces a structure based on four gates, which are used to remember important information, to forget irrelevant information and to select which type of information is used in each iteration during the learning. This revolutionary technique aims to solve many problems improving the classical models.

2.6.2.1 Gated Recurrent Unit (GRU)

Proposed by Chung et al. [2014], Gated Recurrent Unit (GRU) solves the vanishing gradient problem which comes with a standard recurrent neural network. GRU can also be considered as a variation on the LSTM because both are designed similarly and, in some cases, produce equally excellent results. To solve the vanishing gradient problem of a standard RNN, GRU uses an update gate and a reset gate. Basically, these are two vectors which decide what information should be passed to the output. An important characteristic about them is that they can be trained to keep information from long ago without losing it through time or removing information which is irrelevant to the prediction. Figure 2.7 depicts the unrolled GRU and Figure 2.8 presents the GRU cell layout, where, plus symbol represents the plus operation, sigma box is the sigmoid function, circle is the Hadamard product \odot and \tanh is the tangent hyperbolic.

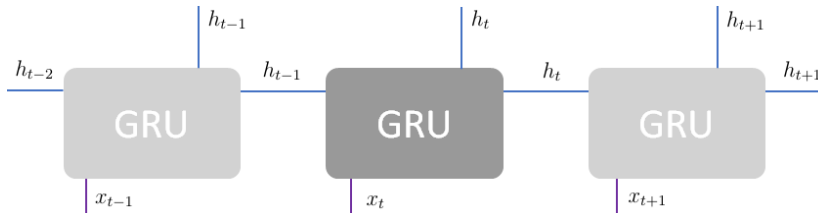


Figure 2.7: Unrolled Gated Recurrent Unit (GRU).

The update gate helps the model to determine how much of the past information (from previous time steps) need to be passed along to the future. It represents an important advantage because the model can decide to copy all the information from the past and eliminate the risk of vanishing gradient problem. This process follows:

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1}). \quad (2.5)$$

Given an input x_t , it is multiplied by its own weight $W^{(z)}$. Similar with h_{t-1} , which holds the information for the previous $t - 1$ units and it is also multiplied by its own weight $U^{(z)}$. Both results are added together and a sigmoid activation function is applied to squash the result between 0 and 1.

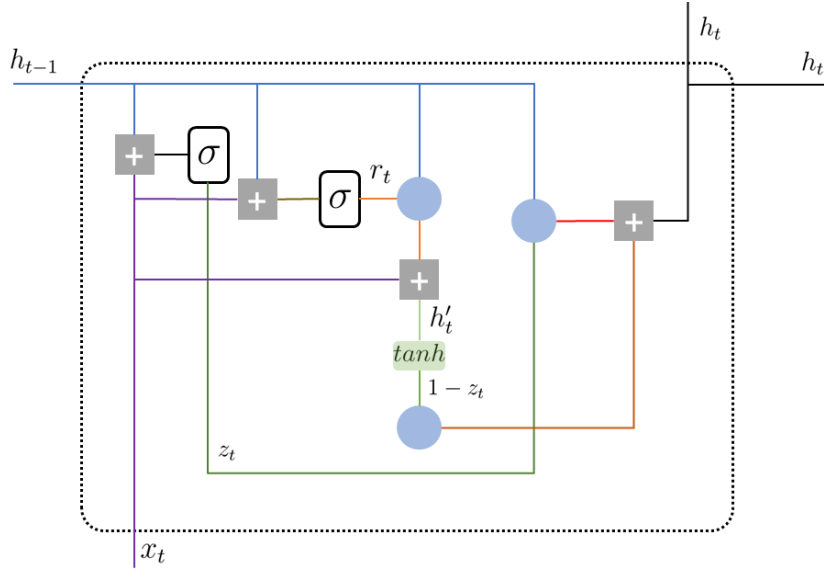


Figure 2.8: Gated Recurrent Unit (GRU).

Reset gate is used to decide how much of the past information to forget. The reset value is computed following the equation 2.5. This formula is the same as update gate, the difference is the weights and the usage of the gate. Once the model computed the information of each gate, the algorithm uses the reset gate to store the relevant information from the past. As:

$$h'_t = \tanh(Wx_t + r_t \odot Uh_{t-1}).$$

The network computes h_t vector which contains information about the current unit and passes it to the network. Update gate determines what to collect from the current memory content h'_t and what from the previous steps h_{t-1} . This process is computed as follows:

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h'_t.$$

GRU eliminates the vanishing gradient problem because it keeps the relevant information and passes it to the next time steps of the network.

2.6.3 Autoencoder (AE)

Autoencoders are a subcategory of (feed-forward) Artificial Neural Network (ANN) which possesses auto-association property. It is an unsupervised learning algorithm trained by using backpropagation method. Autoencoders are similar to dimensionality reduction techniques like Principal Component Analysis (PCA). They project the data

from a higher dimension to a lower dimension using linear transformation and try to preserve the important features of the data while removing the non-essential parts. Autoencoders usually have a hidden layer which has a smaller number of neurons compared to visible layers. The main goal of this particular type of networks is to learn how to reconstruct the data from a lower dimensional space representation.

The AE tries to learn a function $A_{W,b}(x) \approx x$. In other words, it is trying to learn an approximation to the identity function, so as to output \hat{x} that is similar to x . The identity function seems a particularly trivial function to be trying to learn; but by placing constraints on the network, such as by limiting the number of hidden units, interesting structure about the data could be discovered.

AE can be divided into three parts: encoder, code and decoder. Figure 2.9 depicts the classical architecture for AEs. The encoder compresses or down-samples the input into a lower dimension. The space represented by this new dimensionality is often called the latent-space or bottleneck and contains the semantic representation or the code of the input. The decoder intends to reconstruct the input using only the encoding of the input. AEs have generalized the idea of encoder and decoder beyond deterministic functions to stochastic mappings $f_{encoder}(h|x)$ and $g_{decoder}(x|h)$. The goal is to minimize $argmin_{f,g} ||x - (f \circ g)(x)||^2$. In the Figure 2.9, the AE contains three fully connected hidden layers. Thus, given an input $x \in R^d$, the encoder maps into $h \in R^p$, where, $h = \rho(Wx + b)$. After that, decoder looks for rebuilt the representation $\hat{x} = \rho(W'h + b')$. Activation function ρ usually is a ReLu or a Sigmoid. The loss function for this representation is given by

$$L(x, \hat{x}) = ||x - \rho(W'(\rho(Wx + b)) + b')||^2. \quad (2.6)$$

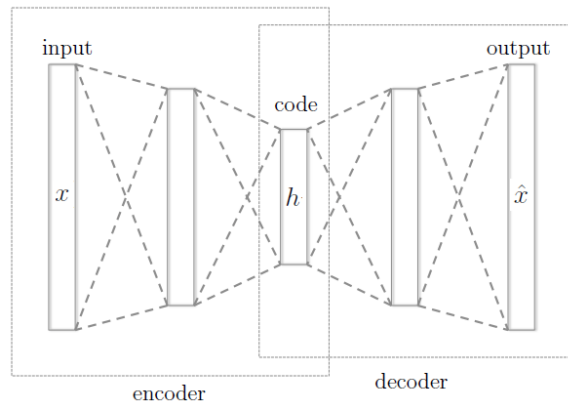


Figure 2.9: Image representation of Autoencoder.

Due to their capacity to build features based on only normal patterns, AEs have

been used in many anomaly recognition approaches [Chong and Tay, 2017]. Thus, AEs offer an alternative of unsupervised feature extraction, where, approaches train with normal representations (images, trajectories, optical flow, among others), therefore, during the test characteristics for the anomalous events, the representations are sufficiently different from the characteristics of the normal representations.

Chapter 3

Related Work

Anomaly detection involves many research areas and application domains, such as cyber-intrusion detection, fraud detection, medical anomaly detection, industrial damage detection, image processing, textual anomaly detection and sensor networks.

Anomalous events are something that people have learned to identify based on their experiences and abilities to understand the context in which things occur. For instance, there is nothing anomalous when car is on a road, however, a car driving on a playground is very strange, and should be considered an anomaly. On the other hand, there is nothing abnormal about a child playing on a playground, but it certainly would be if the child were on the road. Generally, if we see some event, we will consider its context, which will help us to decide whether we should be concerned or not. Thus, an anomalous event depends on the environment and mainly the context. Furthermore, we can define an anomalous event as “*something that is inconsistent with or deviates from what is usual, normal, or expected*”.

It is important to emphasize that an anomalous event is not necessarily an abnormal situation, it only deviates from the normal or expected. An algorithm can be designed to recognize what is normal and anomalous, however it is not currently capable of solving this problem as humans do because a person has years of experience and knowledge regarding many domains, rendering the automatic anomaly event detection a hard and challenging task.

In the following sections, we present the two steps of anomalous event detection: (i) feature extraction and (ii) learning models. Our main goal is to give a brief overview of models and relating them with our study. For further details, please refer to works such as [Lavee et al., 2009; Chandola et al., 2009; Li et al., 2015].

3.1 Feature Extraction

A feature, in computer vision, is a mapping of an image or patch from raw data to an often smaller, however, rich information representation. This should be done by extracting only the key information of the raw data and leaving out the redundant and unimportant information. In general, a system for activity understanding and unusual event detection in surveillance videos involves several key components for processing videos: (i) low-level components for background modeling, feature extraction, and object tracking; (ii) middle-level components for object and action description, e.g. object and action classification; and (iii) high-level components for semantic interpretation, e.g. activity understanding and unusual event detection. Our proposed models focus on low level and high level representations.

A very common category of anomaly detection methods found in the literature addressed the problem by learning activity patterns from low-level handcrafted visual features from spatiotemporal regions. They describe regions using Histogram of Optical Flow (HOF) [Chaudhry et al., 2009], Histogram of Oriented Gradient (HOG) [Dalal and Triggs, 2005], Space-Time Interest Points (STIP) [Laptev and Lindeberg, 2003], and some variations of these descriptors. Ye and Li [2017] proposed a sparse representation based on HOG and HOF to build a semantic dictionary, which is used to compute a score based on each descriptor. Leyva et al. [2017] proposed a model based on optical flow information and HOF, which is utilized in a voting inference scheme from Gaussian Mixture Model (GMM) and Markov Model (MM) Cheng et al. [2016] proposed a Bayesian approach to model HOF and HOG features, classified using a one-class Support Vector Machine (SVM). Cheng et al. [2015] address the problem using local and global analysis, mid-level features (codebooks) extracted from STIP descriptor to perform anomaly recognition.

Other models explore the scene using real world points. De Almeida et al. [2017] proposed a model that describe using real world movement flow, to create a 2D histogram representation. Afterwards, it groups features to determine change in crowd movement. This, model takes in consideration the complete frame. Shi et al. [2016] employed STIP and Multiscale HOF to build a salience map from temporal blocks with different structure, anomalies are detected when testing sequences exceed a threshold in map representation. Besides movement, other approaches focus on appearance, such as Wang and Xu [2016] which employs the wavelet transform to create a saliency map from texture information.

Our approach based on low-level features differs from literature since it introduces a descriptor that explores the subject movement and captures the information in form

of ranges. These ranges aim at detecting strong variations in the orientation or speed of the subjects in the scene. Studies based on descriptors commonly extract texture and movement information and combine them in a descriptor or create codebooks. Instead of that, our model distributes the information in a single descriptor, which can be used directly to detect anomalous events.

A common characteristic of the previously mentioned methods is that features are basically extracted from spatiotemporal regions, defined by the location of interest points. Nevertheless, there are approaches that extract features from different sources. For instance, Xiang and Gong [2008] utilized the blob information extracted from actor (subjects and objects) bounding boxes. Other models employ texture information to create saliency maps. For instance, Duan et al. [2014] proposed a feature named Kernel Dynamic Texture (KDT), which is a statistical model that transforms the video sequence to represent the appearance and dynamics of the video. Yuan et al. [2015], not only employed KDT, but also proposed a novel structure to model crowd behavior, called Structural Context Descriptor (SCD). Then, anomaly are localized using an online spatiotemporal analyzing the SCD variation of the crowd. Finally, Nallaivarothayan et al. [2012] mixed the information regarding blobs, motion (optical flow of regions) and textures extracted from motion to update a double Hidden Markov Model (HMM). In surveillance context, these models present a great advantage over spatiotemporal approaches, which is the specific target location, and in case of trajectories, the complete event performed by the actor. This important characteristic is explored by our approaches that focus on the individual.

The majority of methods model activity patterns only considering local or global context, and not both. In view of that, Xu et al. [2013] proposed a hierarchical framework that considers both global and local spatiotemporal contexts. Regarding the global context, the authors extract atomic activity patterns from low level optical flow features, and the distributions of atomic activity patterns are modeled for higher-level activity representation. Cheng et al. [2015] also proposed a unified framework to detect both local and global anomalies using a sparse set of Space-Time Interest Points (STIP) [Laptev and Lindeberg, 2003]. They identify local anomalies as STIP features with low-likelihood visual patterns and global anomalies as interactions that have either dissimilar semantics or misaligned structures with respect to a probabilistic normal model. We adopt the robustness of such approaches in our third model based on neural networks, which utilizes semantic information extracted from trajectories.

With the success of Deep Neural Network (DNN), researchers have started to use Convolutional Neural Network (CNN), Autoencoder (AE), Generative Adversarial Network (GAN) and Recurrent Neural Network (RNN) to solve the anomaly recognition

problem [Kiran et al., 2018]. Such architectures learn hierarchical layers of representations to perform pattern recognition and have demonstrated impressive results on many tasks. Commonly, the techniques either employ DNN in part of their approach or present a complete model based on DNN.

CNN approaches [Sabokrou et al., 2018; Shao et al., 2016; Zhou et al., 2016] look to describe anomalies, creating models that combine optical flow and texture information from spatiotemporal regions. Most of these models present an end-to-end architecture. Models that use AE or Convolutional Auto Encoder (CAE) [Feng et al., 2016; Chong and Tay, 2017; Qiao et al., 2017; Ribeiro et al., 2017] describe events in non-supervised fashion. Thus, anomalies are representations that differ from normal because they cannot be reconstructed by the AE. Similar to AE, GAN approaches [Ravanbakhsh et al., 2017; Lawson et al., 2017] learn the normal behavior using a generative model. Anomalies are recognized by the discriminator since the generator built an anomaly representation based on normal situations, it means trying to fit something that is different using only the normal. RNN models [Feng et al., 2016; Chong and Tay, 2017] usually appear accompanied with DNN, specially for movement data. The idea is to combine the recurrent information of what is considered normal and create a representation of it. Nevertheless, most of these models depend on the camera position. Thus, these models learn specific patterns of the camera view which cannot be transferred to other views without retraining. Our second approach (human-object interaction) has a significantly advantage from traditional models and its main difference is how it collects the information and uses it in other scenes. Instead of using correlated information between camera and scene, our approach looks for patterns that can be used in other scenes. Thus, human-object iterations aim to determine anomalous events in other scenes that belonging to same context.

Xu et al. [2015] proposed a novel Appearance and Motion Deep-Net (AMDN) framework for discovering anomalous activities. Their model learns discriminative feature representations of both appearance and motion patterns in a fully unsupervised manner. Patches from still images and dynamic motion fields represented with optical flow are used as input of two separate networks, to learn appearance and motion features, respectively. Then, early fusion is performed by combining pixels with their corresponding optical flow to learn a joint representation. Finally, a late fusion strategy is introduced to combine the anomaly scores predicted by multiple one-class SVM classifiers. Similarly to handcrafted features, these techniques also extract texture (appearance) and movement (flow) information. In our model based on trajectories, the source of information for anomaly representation is different. Specifically, our model extracts information from trajectories which implying semantic information about a

person in the scene, for instance the location of the person, its speed, its orientation, the performed path. An important difference with these models is the fact that our model (trajectory based), is not affected with high color intensity changes. Obviously, the number of persons and the distance from camera is an important limitation of our model, compared with crowded models.

To take advantage of the best of both trends (handcrafted and deep features), Hasan et al. [2016a] used an Autoencoder (AE), based on the two types of features, to learn regularity in video sequences. According to the authors, the AE can model the complex distribution of the regular dynamics of appearance changes. As an input to the AE, they used handcrafted features (HOG and HOF) with improved trajectories and learn the regular motion signatures by a AE based on seven stacked fully connected layers. Since the features were not designed or optimized for their problem, the authors claim that it may be suboptimal for learning temporal regularity, thus they used the video as an input and learn both local motion features and the AE by an end-to-end learning model based on a fully convolutional neural network.

During many years, the researchers employ region based models due to the ease of adapting the model in environments with few people and crowds, avoiding using trajectories due to the difficulty of extracting them in environments with a high density of people. However, with the recent developments in object detection and recognition, including new technologies, this issue has been progressively reduced. Pioneering studies on the recognition of anomalies based their extraction of characteristics on trajectories [Wang et al., 2008]. New trends are resuming the use of trajectories to detect anomalous events. Bera et al. [2016] restricted their work to trajectory-level behaviors or movement features per agents, including current position, average velocity (including speed and direction), cluster flow, and the intermediate goal position. In view of that, they developed a pedestrian behavior feature interactively computed from tracked trajectories. In contrast with this approach, our trajectory based approach uses information extracted with neural model, these features contains semantic information about movement. The model proposed by Cosar et al. [2017] takes information of trajectories to build regions, such regions are examined in a time lapse to find texture and movement information. Li et al. [2013] proposed a technique that describes the scene using a sparse representation of overlapping trajectories, which are grouped and abnormal events are recognized when they differs much from any cluster. Chebiyyam et al. [2017] built a graph representation based on trajectories. Saini et al. [2018] proposed a different way to determine anomalies, their algorithm uses the trajectories to train a Hidden Markov Model (HMM), combined with genetic algorithm, which detect anomalies by their low probability. Zhou et al. [2015] proposed a method based on

HMM and feature clustering. An important difference with these approaches and ours is that our trajectory model does not segment the trajectories in parts or blocks, it means our model focus in complete trajectory.

Other studies focus on the trajectory orientation, for instance Dee and Caplier [2010] whose proposed a model based on trajectories representing the information in histograms. This model is very similar to our handcrafted approach, however we introduce entropy information, as well as they take information about the trajectory instead of this, our handcrafted approach extracts information from spatiotemporal region. About our third trajectory based approach, the difference is the semantic information extracted from the neural networks.

Trajectory analysis is based on object tracking and typically requires an uncrowded environment to operate, while Motion analysis, is better suited for crowded scenes by analyzing patterns of movement rather than attempting to distinguish objects [Ryan et al., 2011; Adam et al., 2008] individually. The difficulty of the trajectory approach increases proportionally to the number of individuals in the scene [Popoola and Kejun Wang, 2012; Vishwakarma and Agrawal, 2013]. Most of Motion analysis methods tend to handle a crowd as short groups of people [Choi and Savarese, 2012]. The object-centered approaches [Shao et al., 2014; Mehran et al., 2009] require explicit detection and segmentation of individuals. These techniques, in some cases are not feasible due to severe inter-object occlusion, especially in highly crowded scenes. Other models prefer to represent the crowd scenes using dense information rather than determining interest objects (groups of people). These models focus on scenes where the people movement covers most of the vision field.

3.2 Learning Models

Learning models address the anomaly detection step by learning the normal patterns and using this knowledge to classify normal from anomalous patterns. These approaches are trained in an unsupervised manner using videos containing only normal event, and an incoming video is classified as either normal or anomalous based on the likelihood of the clip according to the trained model, i.e outliers of the model are classified as abnormal while the other are classified as normal [Nallaivarothayan et al., 2013].

Zhang et al. [2015] proposed a model based on Locality Sensitive Hashing Filters (LSHF), in which, the idea is to hash the data into a low-dimensional binary (Hamming) space, where similar data points are mapped into the same bucket with a high

probability while dissimilar data points are hashed into the same bucket with a low probability. Wang and Xu [2016] presented a model that extracts dynamic textures from spatiotemporal regions, this information is built using steerable pyramid wavelet transform. The final detection is based on Gaussian probability function. Andersson et al. [2013] presented an approach where K-means and Hidden Markov Model are trained and recomputed every time that an anomaly is recognized. Zhang et al. [2005] proposed a model that trains an HMM using a large amount of normal situations. It performs iterations of likelihood test and Viterbi decoding on unlabeled video sequences, a number of unusual models is derived using the unlabeled sequences via Bayesian adaptation. These approaches have the benefit of not requiring any training data from anomalous events, which are often poorly represented within the data available for training. However, this approach may also suffer from a high rate of false positives, since any event not sufficiently represented in the training data will be detected as anomalous.

Xu et al. [2015] proposed a model based on patterns learned from one-class SVM classifier. These representations are combined in a weight vector which is the result of linear combination of one class SVM scores. Del Giorno et al. [2016] proposed a model based on classical Histogram of Oriented Gradient (HOG), Histogram of Optical Flow (HOF) and Motion Boundary Histogram (MBH), this study determines the anomalous frames defining probabilities extracted from one-class classifier. Fang et al. [2016] proposed a model based on Principal Component Analysis (PCA) neural network. This model uses optical flow information of patches, a set of PCANet models are learned separately based on them. For each patch, this model removes block-wise histograms represent each block in the patch, as opposed to linking them all to a single vector. After obtaining training features, one-class classifiers are trained to determine anomalous patches. Liu et al. [2014] proposed a model based on a temporal dynamic textures to describe spatiotemporal volumes of events in videos. Using this information an sparse coding is utilized for reconstructing the test data and determining if a volume is anomalous or not. Similar model was presented by Li et al. [2014]. They used dynamic textures as features, however, instead using space coding, they utilized saliency maps to detect where occurs an anomaly. These maps are built from statistical model based on Conditional Random Field (CRF) filters. A feature shared by these methods is the use of local (patches or regions) and global features. Antić and Ommer [2011], proposed a pixel level model, instead using patches, this model determines the abnormality firstly computing foreground pixels, after that, the model creates hypothesis about them. This hypothesis is used to train a probabilistic graphical model to determine if a video parse is normal or not. These methods have the benefit of not requiring labeling the

multiple classes of training data, but suffer from the drawbacks associated with the assumption that all rare events are anomalous. In the results for our trajectory based approach, we present an analysis for this type of situations.

To model the events, the majority of techniques employ Gaussian Mixture Model (GMM) [Pathan et al., 2010] and Hidden Markov Model (HMM) [Zhang et al., 2005]. In [Kratz and Nishino, 2009], a multi-level HMM is used to predict the anomalous events in specific regions of the crowd. In [Andrade et al., 2006; Kim and Grauman, 2009], Markov models allow the analysis of the scene. The expectation maximization algorithm has been also employed as predictor for anomaly [Mehran et al., 2009]. Another statistical model was employed in [Antić and Ommer, 2011], where each pixel has a estimated probability to belong to foreground (there is no movement at that particular location), then by using inference techniques, it determines whether a pixel is an anomaly signal. In [Li et al., 2014], a robust approach uses a hierarchical mixture of dynamic textures to describe the frame. Despite stated in several papers, the models based on the crowd trajectories are hard to accomplish, due to aspects such as occlusion and video quality. Xiang and Gong [2008] proposed a model based on group profiling that is considerably different from common models in the literature. Their model is based on group modeling, where a map of four descriptors define the anomalies. Then, this information is quantized using a bag of words technique and the events are classified as anomalous or not using a Support Vector Machine (SVM). The popularity of these models is due to their ability to model events by means of probabilities. Many of them even consider temporal sequences. However, with the tendency and success of convolutional and recurrent networks, the use of these models has declined in recent years.

A straightforward, however successful technique is based on k-Nearest Neighbors (k-NN) ranking. For instance, Saligrama and Chen [2012] proposed a model based on handcrafted features, where the relation between spatiotemporal regions is given by a MM, and the local analysis is performed using k-NN in each region. In our handcrafted approach, we utilize a similar pipeline. In our trajectory based approach, the learning model is based on the own trajectory descriptors, nevertheless, the final recognition is also performed using k-NN technique.

Many models based on Neural Networks [Kiran et al., 2018] present an end to end architecture, it means, the complete process of learning is part of the architecture, this includes the feature extraction and the anomaly detection. In last few years, this type of models has achieved great popularity among researchers, due to the success of the networks. As aforementioned, in our approach based on trajectories, we employ neural networks to create the feature representation, the learning process is performed

by an Autoencoder (AE). We employ also NNs in our study, however, a difference with conventional neural network models is that our model focus in trajectory information, while classical models focus on spatiotemporal information.

Chapter 4

Anomaly Detection based on Handcrafted Feature Descriptor Extracted from Spatiotemporal Regions

Anomalous event detection in video sequences has proven to be challenging because of the variations in the environment, appearance of actors, the way that same action is performed by different people, speed, duration and points of view from which the event is observed. This variability renders the difficulty of defining effective descriptors. Many studies focus on extracting information that suits in many context scenes. Thus, handcrafted features from spatiotemporal regions have been a effective tool in most studies in literature.

In this chapter, we present two representations for anomalous event detection approaches, both based on handcrafted feature descriptors. Inspired on the Histogram of Optical Flow (HOF) [Chaudhry et al., 2009]. While the first proposition considers the magnitude and orientation of the optical flow, the second incorporates entropy information to better detect possible anomalous events. This approach aims to detect anomalies in static camera view and collects only information from moving objects.

4.1 Overview

Handcrafted features extracted from a spatiotemporal region are presented in many classical studies [Li et al., 2015]. The goal of these methods is to extract low-level

features from motion, color, and other fundamental image properties. The following paragraphs presents some important characteristics of such methods.

The main advantage of using information from low level features is that the majority of activities, which are part of any event, present movement. Thus, models based on this type of information are robust to image-processing difficulties, for instance, occlusion. Further, since no objects are explicitly detected, approaches based on spatiotemporal regions are able to operate successfully even with large numbers of targets in view, for instance crowded scenes [Sodemann et al., 2012]. It is for this reason that these types of models are popular in recognition of anomalous events containing crowds. Finally, these models are suitable for real time applications [Roshtkhari and Levine, 2013]. These are important advantages of classical handcrafted feature descriptors, due to their simple model structure.

Generally, the main drawback associated with these methods is their locality. Since the activity pattern of a pixel cannot be used for behavioral understanding, their applicability in surveillance systems is restricted to the detection of local temporal phenomena [Mehran et al., 2009]. Even though, this disadvantage depends on the objective of anomaly detection, many cases suit well with this type of approaches.

The following sections describe our approach to detect anomalous event based on handcrafted features extracted from spatiotemporal regions. Our main contribution are two feature descriptors. The first descriptor is the Histograms of Optical Flow Orientation and Magnitude (HOFM) and the second is an extension, called Histograms of Optical Flow Orientation, Magnitude and Entropy (HOFME). These descriptors aim to capture anomalies based on the movement of a defined region. Therefore, the anomalies that can be detected only consider situations that involve movement. Our models are oriented to situations with crowds and, where in general, the camera has a wide view of the scene, which allows to analyze by regions the scene.

4.2 Proposed Approach

Based on common anomalous events, such as pedestrians moving with excessive speed, spatial anomaly (intruders in restricted areas or unusual locations), and presence of non-human objects in unusual locations [Ryan et al., 2011], we define four characteristics that could be extracted from spatiotemporal regions, to describe normal motion patterns in a particular region of the scene: i) *velocity* - speed of moving objects; ii) *orientation* - common flow of the objects; iii) *appearance* - texture of the objects; and (iv) *density* - number of moving objects. We hypothesize that the use of such

characteristics allow us to capture information regarding anomaly. Note that these characteristics allow the detection of anomalies based on movement and appearance. Hence, we propose a spatiotemporal feature descriptor based on *orientation* and *velocity* information extracted from optical flow. The goal of this descriptor is to split movement information to discriminate speed and orientation of components in the region. Therefore, we can determine anomalous event comprising fast/slow movements and unknown direction of actors (people, cars, bicycles) in the scene, thus the feature extraction step must describe the movement in certain regions in the frame sequence.

Our method is composed of two main steps: (i) descriptor extraction and (ii) anomalous event detection. It divides the video into non-overlapping spatiotemporal regions and builds an orientation-magnitude representation for each region (*cuboid*). On the training step, the histograms extracted from cuboids are assumed to contain normal patterns. In other words, our model collects the histograms in a specific spatiotemporal region to be able to capture the normal patterns. During tests, for a specific cuboid, our approach performs a simple nearest-neighbor search to find similar patterns stored during training at that region. If none is found, we consider it as anomalous. An overview of the approach is illustrated in Figure 4.1.

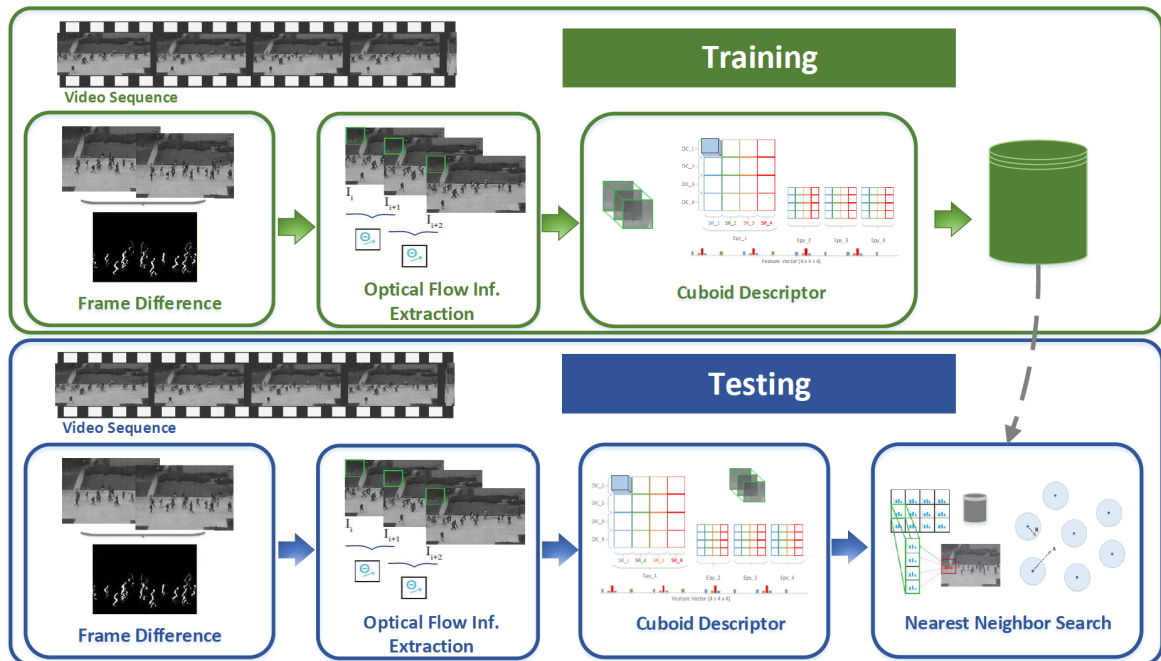


Figure 4.1: Diagram illustrating the proposed approach to performing anomalous event detection.

4.2.1 Descriptor Extraction

Our proposed descriptor is based on two main steps: (i) Frame difference and Optical flow extraction; and (ii) histogram building, described in the next paragraphs.

Frame Difference and Optical Flow Extraction

Since the extraction of optical flow for the whole image can be computationally expensive [Ryan et al., 2011], we first create a binary mask using image subtraction between the frame I_i and the frame I_{i+t} (where i is the frame number). Given a threshold d , if the resulting difference is smaller than d , then the pixel is discarded; otherwise, this pixel p is set to its corresponding local cuboid C_i . The spatiotemporal region C_i is a tri-dimensional representation of a region R_i in c consecutive frames. c is considered as temporal length, and region R is a patch of the frame. The optical flow is computed between $frame_{i-1}$ and $frame_i$ as illustrated in Figure 4.2. It is important to mention, that consecutive frames are not necessarily immediately consecutive frames, in this way, discarding intermediate frames allows us to capture more movement information. Thus, model can take two frames discarding n frames between them.

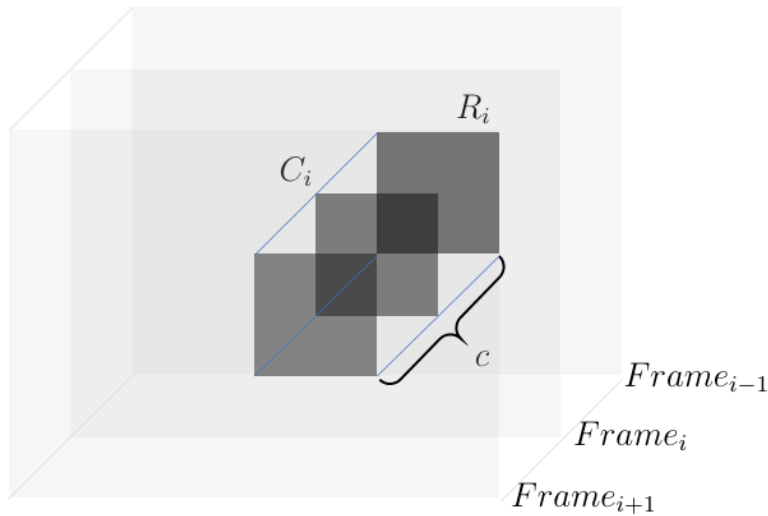


Figure 4.2: Image representation of a spatiotemporal region or cuboid C_i , which is composed of region R over c frames.

Figure 4.3 shows an example of this stage. We can see two consecutive frames and the resulting frame difference image. In Figure Figure 4.3(d), the colored pixels show optical flow output position, in green the past frame, and in pink the optical flow vector position. An important aspect to mention is that the quantity of information to be processed has been reduced whit this pre-processing step.

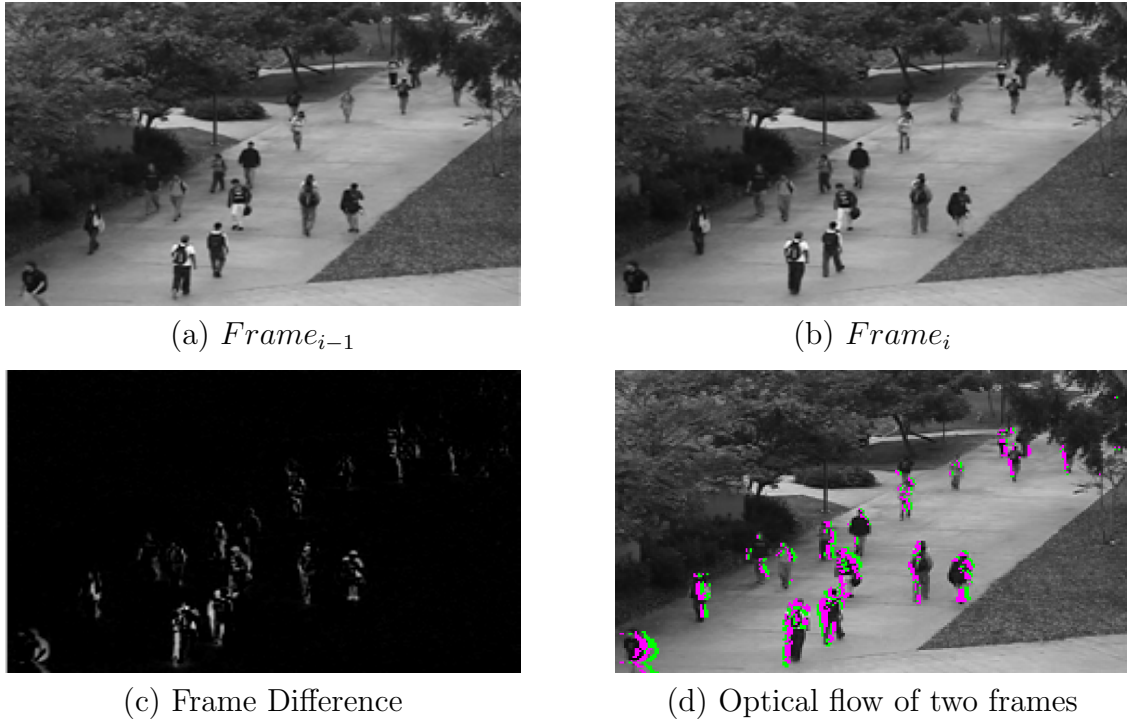


Figure 4.3: (a) and (b) are two consecutive frames. (c) is the frame difference between Frame $i - 1$ and Frame i , (d) is the optical flow representation, colored points represent the optical flow translation, green for previous frame, pink represents optical flow output. Then, image in (c) is a mask, thus, only white pixels this image are used to compute optical flow.

Histogram Building

In this section, we present and discuss our proposed feature descriptors. First, we detail the Histograms of Optical Flow Orientation and Magnitude (HOFM) descriptor algorithm, afterwards, the Histograms of Optical Flow Orientation, Magnitude and Entropy (HOFME). The main motivation of our proposed descriptor is to encode the movement of a determinate region, with the aim of capturing information about the normal flow of subjects that pass through the regions. Inspired in HOF the idea is to distribute the flow information in order to differentiate direction and velocity (magnitude) in each region, it means from any object that presents movement. Thus, scenes where the context presents slow movement like people walking (university, public institutions, etc.) our descriptor and our model in general gets to differentiate this type of events from people running.

Let be $F_{S \times B}$ a matrix, where S is the number of orientation ranges and B the number of HOF magnitude ranges. Similar to the original HOF, we build an accumulation matrix based on the orientation of the vector, but we incorporate information

of magnitude provided by the vector field resultant of optical flow (note that the magnitude of the optical flow indicates the velocity that the pixel is moving). Thus, given the pixels $p(x, y, t)$ and $p'(x, y, t + 1)$ belonging to the cuboid C_i , $1 < i < n_c$ is the id position of cuboids in the grid created by cuboid layout. For instance, similar to matrix representation, a grid with 4×5 layout will have $n_c = 20$ cuboids, where $i = 5$ is the position $(2, 1)$ in this grid. The vector field \vec{v} between p and p' is composed by the magnitude m and orientation θ . In this way, for each cuboid at time t , we compute the cumulating matrix F using

$$F(s, b) = \sum_{\vec{v} \rightarrow C_i^t} \begin{cases} 1 & \text{if } (s = \text{mod}(m, M)) \text{ and} \\ & (b = \text{mod}(\theta, B)) \\ 0 & \text{otherwise} \end{cases}, \quad (4.1)$$

where $s \in \{1, 2, \dots, S\}$ and $b \in \{1, 2, \dots, B\}$ denote orientation and magnitude ranges, respectively. The spatiotemporal descriptors are computed for each cuboid C_i .

Figure 4.4 shows a matrix presenting four magnitude and orientation ranges. Each pixel in the cuboid C_i increments the occurrence of a specific bin in the accumulation matrix. In this way, our feature vector can be seen as a matrix, where each line corresponds to a orientation range and each column corresponds to the magnitude ranges. For instance, the example pixel in figure 4.4 has $(50, 17)$, orientation and magnitude values, this pixel increments the value in $M_{1 \times 1}$, since the angle 50 is in OC_1 range and its speed is between $(0, 20]$, corresponding to first column. In this example for simplicity, the temporal depth is $c = 2$. In case of $c > 2$, there will be more optical flow vectors for each pixel position, e.g., $t = 4$ yields three optical flow images.

The Histograms of Optical Flow Orientation, Magnitude and Entropy (HOFME) is a direct extension of the Histograms of HOFM. Where, we introduce orientation entropy to the descriptor. The idea is to collect more information regarding texture in the spatiotemporal regions. The HOFME steps are very similar to the HOFM. The main difference is in the histogram building stage.

For the HOFME histogram, we define a cube $F_{S \times B \times E}$, where S is the number of orientation ranges, B the number of magnitude ranges and E is the number of entropy ranges. We build a 3D matrix based on the orientation and magnitude information provided by the optical flow. We refer to *cube*, as a graphical representation of the cumulating matrix, the word *cuboid* refers to a spatiotemporal region. For the entropy information extraction, we use the orientation information, thus, different flow orientations in the spatiotemporal region may indicate the presence of distinct objects moving in different directions, it is important because it provides to the model information

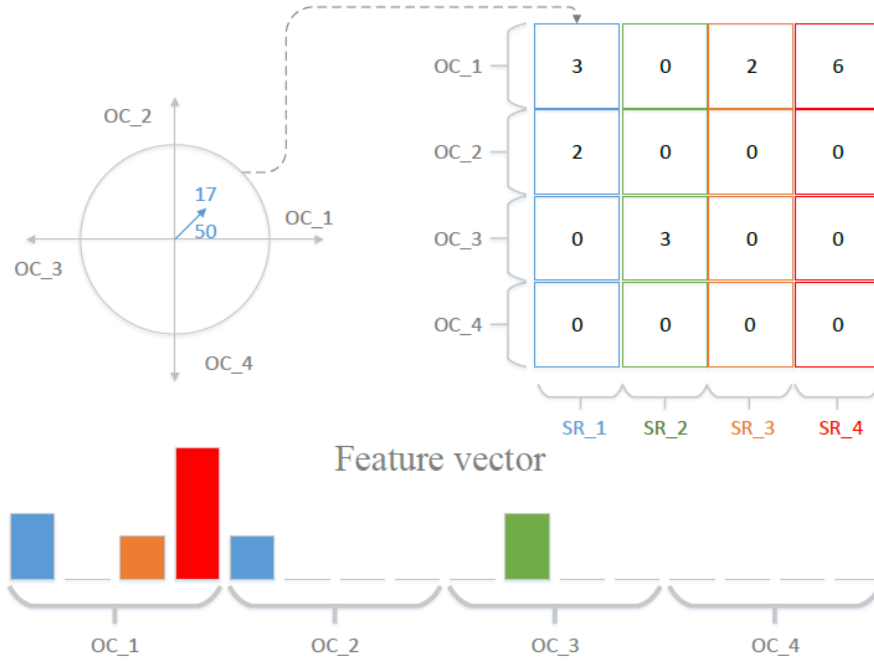


Figure 4.4: Example of feature vector extraction using orientation-magnitude descriptor. This figure shows a matrix presenting four magnitude ranges: $\{(0, 20], (20, 40], (40, 60], (60, \infty)\}$, named SR_1, SR_2, SR_3, SR_4 . All magnitudes are represented by colors blue, green, orange and red, respectively. Moreover, this figure also presents four ranges for orientations: $\{(0, 90], (90, 180], (180, 270], (270, 360]\}$, named as OC_1, OC_2, OC_3, OC_4 .

about different actors in the region or objects with different size. In this way, if some cuboid presents different orientations, it is more likely that “subjects” or “objects” on it have a distinct movement, because, commonly moving objects present the same direction. For instance, a vehicle passing through the view presents low entropy in its orientation flow.

Given pixels $p \in C_i^t$, the vector \vec{v}_p is composed by magnitude m and orientation θ . For each cuboid at time t , we compute the cube feature F using the Algorithm 2.

Algorithm 2 HOFME Descriptor algorithm.

- 1: **procedure** HOFME(C_i^t)
 - 2: C_i^t is the cuboid i at time t
 - 3: **for** each pixel $p \in C_i^t$ **do**
 - 4: $e \leftarrow Entropy(p)/E$
 - 5: $s \leftarrow p_\theta/S$
 - 6: $b \leftarrow p_m/B$
 - 7: $F[e, s, b] \leftarrow F[e, s, b] + 1$
 - 8: **return** F
-

In Algorithm 2, $s \in \{0, 1..(S-1)\}$, $b \in \{0, 1..B\}$ and $e \in \{1, 2..(E-1)\}$ represent the bins in the cuboid, respectively. S , B and E are the factors for the number of bins (*e.g.*, if we use 4 bins for orientation, the range B is 90° or $2\pi/4$).

The magnitude range is a variable that depends on the scene. For instance, when moving objects are far from camera, the optical flow vector presents a small magnitude, in contrast, when the moving objects are near the camera, the optical flow vector presents a high magnitude. Although, this variable is set according to the scene, the number of ranges is fixed. Magnitudes that exceed the maximum value in the range are truncated. Thus, pixels that present higher magnitudes, we increment the bin corresponding with the last magnitude range. Thus, the distribution in the histogram is clearly divided or separated by small or large magnitudes.

To compute the entropy, we use a 3D patch around the pixel p , similar to cuboid however much tiny. The first step is to build the orientation distribution around the pixel p . The resultant histogram O_p is normalized to get the probability of each quantized orientation for pixel p , finally the entropy is computed by

$$Entropy(p) = - \sum_i^S O_p(i) \log[O_p(i)],$$

where S is the number of quantization ranges as well as orientation used to build the cumulative cube.

Figure 4.5 shows a matrix presenting four magnitude and orientation ranges. Similar to HOFM descriptor, each pixel in the cuboid C increments the occurrence of corresponding bin in the histogram. The feature vector could be represented as a cube, where each line corresponds to a determinate orientation range, each column corresponds to the magnitude ranges and the deep of the cube represents the entropy measure. In the example, the pixel increments a unity in $M_{1 \times 1}$, since the angle 50 is in OC_1 range and its speed is between $(0, 20]$, corresponding to first column. Finally, we compute the entropy orientation measure. Around the pixel $p(50, 17)$, all the adjacent pixels have the same orientation, then the histogram for orientation distribution is accumulating in just single bin range, using equation 4.2.1, the entropy is 0. Hence, zero entropy corresponds to Epy_1 . In this example the depth of cuboid is $c = 2$, then, there is only one optical flow matrix (magnitude and orientation each pixel).

Other studies [Wang et al., 2013; Laptev et al., 2008], similar to ours, are also based on optical flow information, nevertheless, there are some fundamental differences. First, our descriptor instead of accumulating the magnitude value in the orientation bin, it represents the magnitudes in a different axis, quantizing the magnitude in ranges,

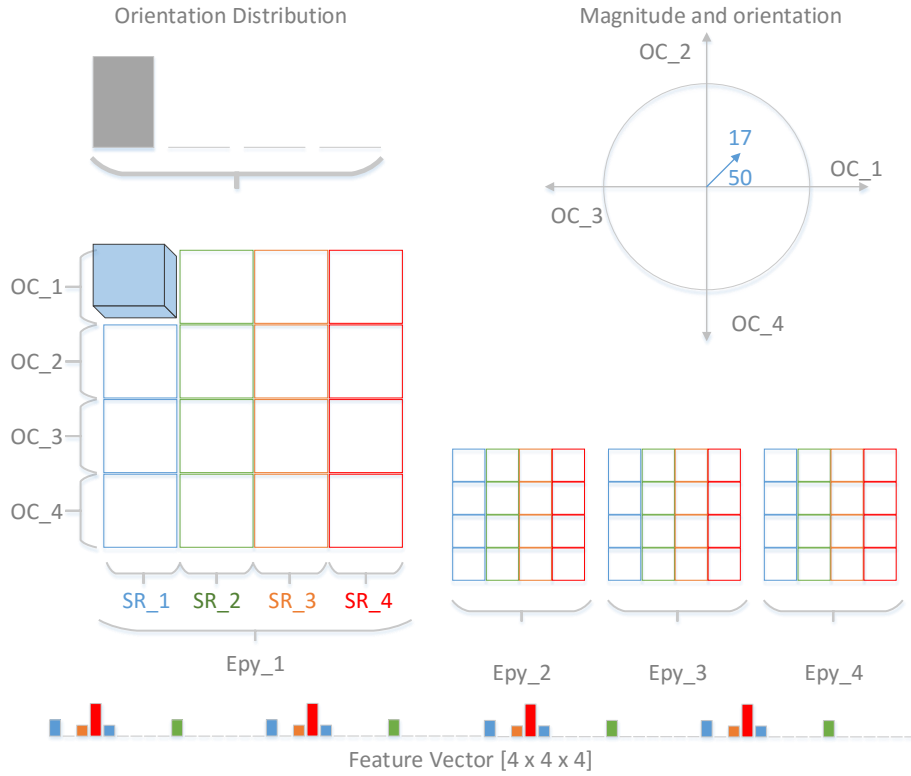


Figure 4.5: Matrix representation for four magnitude ranges: $\{(0, 20], (20, 40], (40, 60], (60, \infty)\}$, named SR_1, SR_2, SR_3, SR_4 . All magnitudes are represented by colors blue, green, orange and red, respectively. Moreover, this Figure also presents four ranges for orientations: $\{(0, 90], (90, 180], (180, 270], (270, 360)\}$, named as OC_1, OC_2, OC_3, OC_4 . Finally we can see the entropy as a third dimension based on four ranges also $\{(0, \frac{1}{2}], (\frac{1}{2}, 1], (1, \frac{3}{2}], (\frac{3}{2}, 2)\}$ labeled as $Epy_1, Epy_2, Epy_3, Epy_4$ respectively. Entropy value is between $[0, 2]$ given base of two.

providing information regarding to orientation and velocity separately. The second important difference is the entropy. Our descriptor also includes another axis, which represents the orientation variation. Finally, the descriptors proposed in [Wang et al., 2013; Laptev et al., 2008] concatenate other descriptors HOF, HOG and MBH, all to compose the feature vector.

4.2.2 Detection of Anomalous Events

In this step, for each cuboid, the feature vectors for testing are compared with normal feature vectors, collected in the training. Similar to Saligrama and Chen [2012] model, the goal of this process is to search in the normal pool some feature vector that are similar to the sample test. Therefore, if a sample test is similar to a normal pattern, then, it is classified as normal. Otherwise, it is considered as an anomalous pattern.

This process is explained in Algorithm 3, where, for a specific cuboid C_i , T is the feature vector for a test sample, N is the set of features vector extracted from C_i . In each iteration, the algorithm computes the distance $\|T - n\|^2$, where n is a normal pattern in N . If the distance is less than a threshold τ , then the algorithm stops and returns false, considering the sample a normal pattern. Otherwise, the algorithm returns true, which indicates that the test sample T is an anomalous pattern.

Algorithm 3 Anomaly detection with nearest-neighbor search.

```

1: procedure NEAREST NEIGHBOR SEARCH( $T, C_i$ )
2:    $T$  is a test feature vector at cuboid  $i$ 
3:    $N$  is a set o normal feature vectors cuboid  $C_i$ 
4:   for  $n \in N$  do
5:     if  $\|T - n\|^2 < \tau$  then
6:       return False
7:   return True

```

Figure 4.6 illustrates this step by using blue points to represent patterns seen during training and orange points to represent test samples.

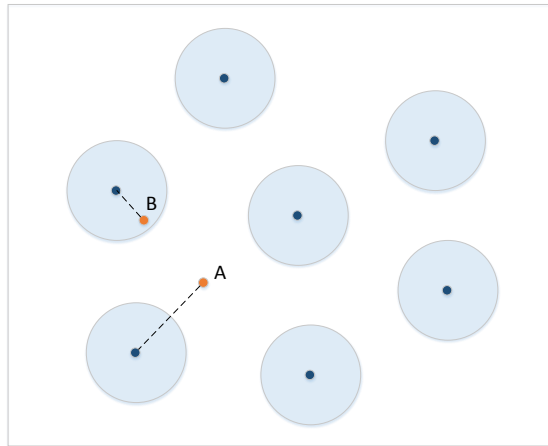


Figure 4.6: Nearest neighbor search. Anomalous patterns is represented by point A, normal patterns is point B.

4.3 Experiments

In this section, we evaluate our anomaly detection approach based on handcrafted features using the well-known UCSD [Lab, 2014] and Subway [Adam et al., 2008] datasets and a proposed dataset called Badminton. The criterion used to evaluate anomaly detection accuracy was based on frame-level (*i.e.*, there is an anomaly in a given frame),

as most works in the literature, the algorithm predicts the frames containing anomalous events and those predictions are compared to the ground-truth annotations.

4.3.1 Datasets

UCSD Dataset

UCSD [Lab, 2014] is an annotated publicly available dataset for the evaluation of anomalous event detection and localization in crowded scenarios featuring pedestrian walkways [Mahadevan et al., 2010]. The dataset was acquired with a stationary camera, each frame has 238×158 pixels and it was recorded with frame rate of 10 frames per second. Anomalous events are due to either the circulation of non-pedestrian entities in the walkways or anomalous pedestrian motion patterns.

The UCSD videos are divided into two scenarios: *Peds1* and *Peds2*, each captured by a camera at a different location. The videos recorded from each scenario were split into various video sequences (clips) containing around 200 frames. The number of training sequences is 27 and 16 for *Peds1* and *Peds2*, respectively.

Subway Dataset

This dataset is composed of two sequences in a set of videos proposed by Adam et al. [2008]. The first video sequence, known as *Entrance Gate*, has a time length of one hour and 36 minutes and the second video, called *Exit Gate* has length of 43 minutes. These sequences correspond to a ticket gate in a subway entrance and exit. The original ground-truth provided by the authors, containing the initial frame of anomalous events, focuses on two specific anomaly types: walking in wrong direction and jumping the ticket gate.

Entrance Gate is a sequence recorded from a subway entrance gate view. The training phase considers the initial 20 minutes (first 30,000 frames) and the remaining of the clip for test (approximately one hour and 16 minutes), where the ground-truth presented two types of anomalies: walking on wrong way and jumping the ticket gate.

The *Exit Gate* clip contains data recorded from a subway exit. In this case, the ground-truth considers only people walking in wrong way. The training set considers only the first five minutes (first 8,000 frames) and the rest of video is used to test.

Badminton Dataset

This clip was recorded from a badminton game. In this video, with a total of 345 seconds captured at 30 frames per second and with a frame size of 640×360 pixels.

The scene for this video sequence is simple; there is a grandstand with people watching the badminton game, people are sitting in the grandstand. We built the ground-truth focusing on motion activities occurred in the grandstand, especially three situations, when people jump the stands, people run in the corridor and people celebrate a game point. Along the video, the initial 56 seconds were used for training to determine what were the normal activities, such as people climbing up the stairs or walking from the right side to the left side of the camera and vice-versa. Activities which occurred in the rest of the video that are different from those previously described were considered anomalies. The anomalous events detected were people running, which occurred three times, and individuals walking down the stairs, which occurred several times. In this video sequence, we define a Region of Interest (ROI), focusing only on the grandstand, discarding the region where players are competing.

4.3.2 Parameter Setting

Before explaining the experiments, we present the hardware configuration and the parameter tuning. Our experiments were carried out using a Intel Core I7 (3.2 GHz) processor and 8 Gbs of memory. Most of the processing has two bottleneck, first in the optical flow extraction, and second in anomaly detection using the K-nn strategy.

In first stage, for frame difference and optical flow extraction we changed the frame size, duplicating its size, with the aim of getting more motion optical flow information since the optical flow vectors present larger magnitudes. For optical flow extraction we used the pyramidal implementation [yves Bouguet, 2000] presented in the OpenCV library [Bradski, 2000]. Specifically, *calcOpticalFlowPyrLK* function receives as input the consecutive frames, the list of points to determine the optical flow vectors (in our case the points that are segmented with the frame difference step), and finally a windows size which is the motion boundary (30×30 in all our experiments).

The first descriptor HOFM uses four bins for each orientation and six for magnitude. HOFME uses four bins for each orientation, magnitude and entropy ranges. The model also filters the pixels with small moving magnitude by thresholding values lower than 0.5 and removing the respective pixels, thus with the aim to filter small magnitudes that could correspond to intensity variations due to video compression. These values were chosen in empirical way, trying to obtain the better results.

Both descriptors share two main parameters: (i) the threshold τ , for the nearest neighbor search; and (ii) the cuboids size ($n \times m \times t$), cuboid spatial dimension (width and height) were set to 30×30 pixels and 5 frames as a temporal depth for both HOFM and HOFME descriptors. Here, we changed the τ value to generate the Receiver

Operating Characteristic (ROC) curves, Equal Error Rate (EER) and Area Under Curve (AUC). ROC curve is utilized to measure the detection accuracy and based on the ROC curve, there are two evaluation criteria: AUC and EER which is the ratio of misclassified frames at which False Positive Rate (FPR) = $1 - \text{true Positive Rate (TPR)}$.

The setup of this parameters is highly dependent of the camera view. Hence, the values must be set according to the scene view, for instance, camera view distance, the image size, the people size in the scene, among others. In our experiments we chose parameter values after many empirical tests, where, we concluded that a good size for cuboids is when a person in the center of the frame image fits at least into two cuboids.

For our experiments, we employed Euclidean distance to determine the similarity between patterns. This is due to other performed experiments during testing the Euclidean distance was the one that achieved the best results (we carried concept test using hamming, cosine and the mahalanobis distance).

Also, we performed several tests varying the number of bins of the descriptor, increasing or decreasing the orientation and magnitude ranges. In the results, we present only the best configuration.

4.3.3 Evaluation on the UCSD Dataset

Table 4.1 shows the results considering the UCSD dataset. On Peds1 scenario, HOFME approach achieved an EER of 32.0% and an AUC of 0.727, being competitive to most of the reported methods on the literature, HOFM achieves 0.727 and 33.3%, AUC and EER respectively. On the other hand, on Peds2, we achieved an EER of 20% and AUC of 0.875, outperforming all reported results. HOFM descriptor obtains 20.7% of EER and 0.87 of AUC. The ROC curves for the two scenarios are shown in Figures 4.7 and 4.8. ROC curves and the AUC (curves and values were gently provided by Li et al. [2014], we included our results on the figure). In this experiment, we compared our descriptors with other handcrafted descriptors. Figures 4.7 and 4.8 present HOF descriptor results. Although they have almost random behavior, we present them because our descriptor is inspired on them. In literature, new approaches using neural networks as they employ other type of features, they are not present in this comparison.

According to the results, HOFM and HOFME achieved similar results in this dataset. The entropy information was able to incorporate information only in few cases where car is passing through. We investigated the cases where our approaches failed. Most of the undetected anomalous frames correspond to very challenging cases, such as a skateboarder or a wheelchair going in an almost similar velocity of the pedestrians

Table 4.1: Anomaly detection AUC and ERR (%) results of HOFME on UCSD dataset. The results for [Li et al., 2014] was obtained from the original paper.

Approach		Peds1		Peds2	
		AUC	ERR (%)	AUC	ERR (%)
	MDT-temporal [Li et al., 2014]	0.827	25.4	0.775	25.9
	HOG3D [Kläser et al., 2008]	0.52	50.0	0.61	47.7
	MBH [Wang et al., 2013]	0.57	43.4	0.55	45.0
	HOOF [Chaudhry et al., 2009]	0.69	36.4	0.82	25.9
Our	HOFM	0.727	33.3	0.87	20.7
Results	HOFME	0.727	33.1	0.875	20.0

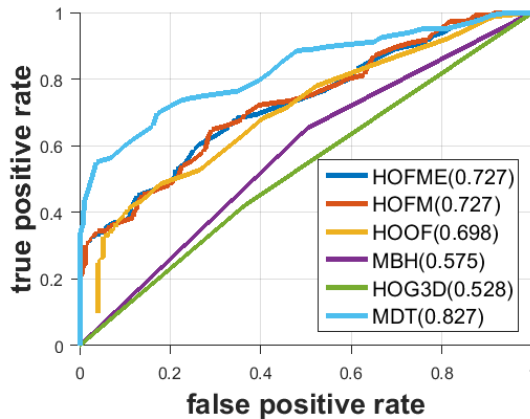


Figure 4.7: Results for Peds1.

and with partial occlusions, as shown in Figures 4.14(b) and 4.14(c). These errors occurred in sequences 21 and 12 of Peds1 and Peds2, respectively. We can see that appearance is an important criterion in UCSD dataset. The entropy information is computed over the orientation information and it is used to filter anomalies, especially in very crowded scenes. For instance, in the middle of the street where people are walking when a car passes, the entropy of the car is smaller than normal events where people is walking in different directions.

Although our model does not include explicit appearance information, it incorporates spatial characteristic and orientation entropy information. Thus, higher entropy values may mean that many pixels are moving with different velocities and, consequently, the objects in the scene do not have a regular texture, which captures some information regarding appearance and density of the region. For instance, some locations in the scene may not present movement on the training sequences however, during test, subjects might appear in those regions, which should be considered as an anomaly since such informations were not present in training. In addition, depending

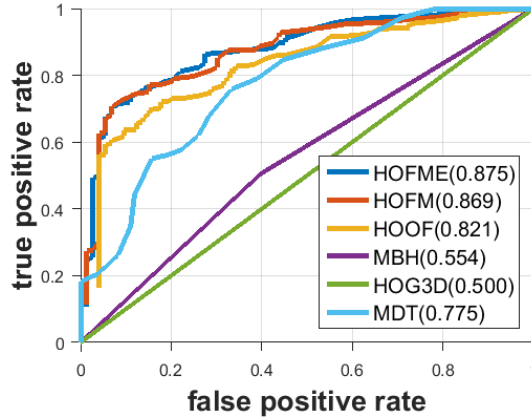


Figure 4.8: Results for Ped2.

on how the ground-truth was labeled, these regions may be omitted or considered as a normal situation. Figure 4.14(d) illustrates one of such cases.

It is important to highlight the premise used by our approaches, we use patterns to determine the anomalous cases, *i.e.*, patterns that do not occur during the training phase are considered as anomalous during the test phase. In this way, our model intended to be as general as possible, even when it is impossible to train any model with all possibilities of “normal” motion. Another important remark of our proposal is that it does not use appearance information. The main reason for this is because the appearance of the unknown is difficult to quantized. For instance, when a histogram based on codebooks is constructed, patterns fit in determinate bins, however, in case of anomalous patterns they also have to fit in other bins and cannot be differentiated from normal ones. We may specialize our descriptor for particular scenes or environments, but this would reduce the generalization capability of our proposal.

Finally, entropy does not provide appearance information directly, it introduces information regarding region texture when something is moving in the cuboid (solid), the orientation of the block will be similar and the entropy will be low. In case of many orientations, for instance, people walking in different directions, the entropy will be high. This type of information helps to differentiate some regions, for instance where there are different flows or when the objects are very different. Our experiments show that entropy provides marginal improvements to the recognition model.

4.3.4 Evaluation on the Subway Dataset

Exit Gate. For the Exit Gate clip, we counted the matches and false alarms manually. We performed various experiments and the best configuration achieved 100% of

accuracy recognizing all anomalous events. Nonetheless, our descriptor also reported 40 false alarms. Experimentally, we chose six bins for orientation, four ranges for magnitude and five frames of temporal depth for HOFM and HOFME. Figure 4.9 shows examples of true positive matches using HOFME. Figure 4.9 presents some examples of correct matches using the HOFME descriptor.



Figure 4.9: Examples of true positive matches for the exit gate clip.

Entrance Gate. This clip contains sequences for the subway entrance gate. In total, we have 31 anomalous events. Following the protocol presented by the authors, we trained with the initial 20 minutes and tested our model using the rest of the clip, which is approximately one hour and 16 minutes. We obtained 83% of accuracy. Figure 4.10 presents some examples of correct matches using the HOFME descriptor.

Although our model recognized most of the anomalous events, it also presented many false alarms (67 for the entrance gate clip and 40 for the exit gate clip). During the training phase, people move in few directions in the entire scene. Then, in the test, our model detects anomalies of people walking in different directions from the learned one in the training phase. The dataset ground-truth only considers as anomaly people jumping over the ticket gate. Thus, the false alarms obtained by our method are due to the creation of the ground-truth assuming that anomalies could take place only near the ticket gate.

A second aspect to discuss is the anomaly based on jumping the ticket gate. In this case, the anomaly contains high semantic information. Our descriptor looks for atomic information (such as orientation, entropy and velocity), discarding explicit modeling of appearance. When people jump the ticket gate, velocity and appearance



Figure 4.10: Examples of true positive matches for the entrance gate clip.

information may be used to recognize the action. However, during the training, velocity and direction in this specific region is the same when passes thorough the ticket gate in much cases. In this way, we are not able to recognize it as anomalous since during the training, this type of orientation and magnitude appears.

Figures 4.11(a) and 4.11(b) show some of aforementioned events. Most of them correspond to people that walk by ignoring the ticket gate. Another important aspect to consider is the velocity. For instance, in Figures 4.11(c) and 4.11(d), the person jumped the ticket gate but our model did not detect this action. However, after the man started running our model detected it as anomalous since nobody runs in that direction during the training phase. A very similar case happens in Figures 4.11(e) and 4.11(f), where the person appears running out of the scene, this will be considered as anomalous event.

The New Ground Truth for the Subway Dataset Since the ground-truth in subway clips focuses on events that involved the ticket gate, many other events may happen in the whole scene. For instance, a young boy running in the corridor or people walking in forbidden areas.

Now, we propose an alternative ground-truth for the subway dataset. The criterion used to determine the anomalies in the clip is based on the following premises: any event that have not occurred during the training stage is reported as anomalous. We considered as “event”: the directions, the speed, the location, and also the original subway ground-truth. Therefore, for instance, if someone runs in testing and nobody ran during the training, that event will be considered as anomalous.

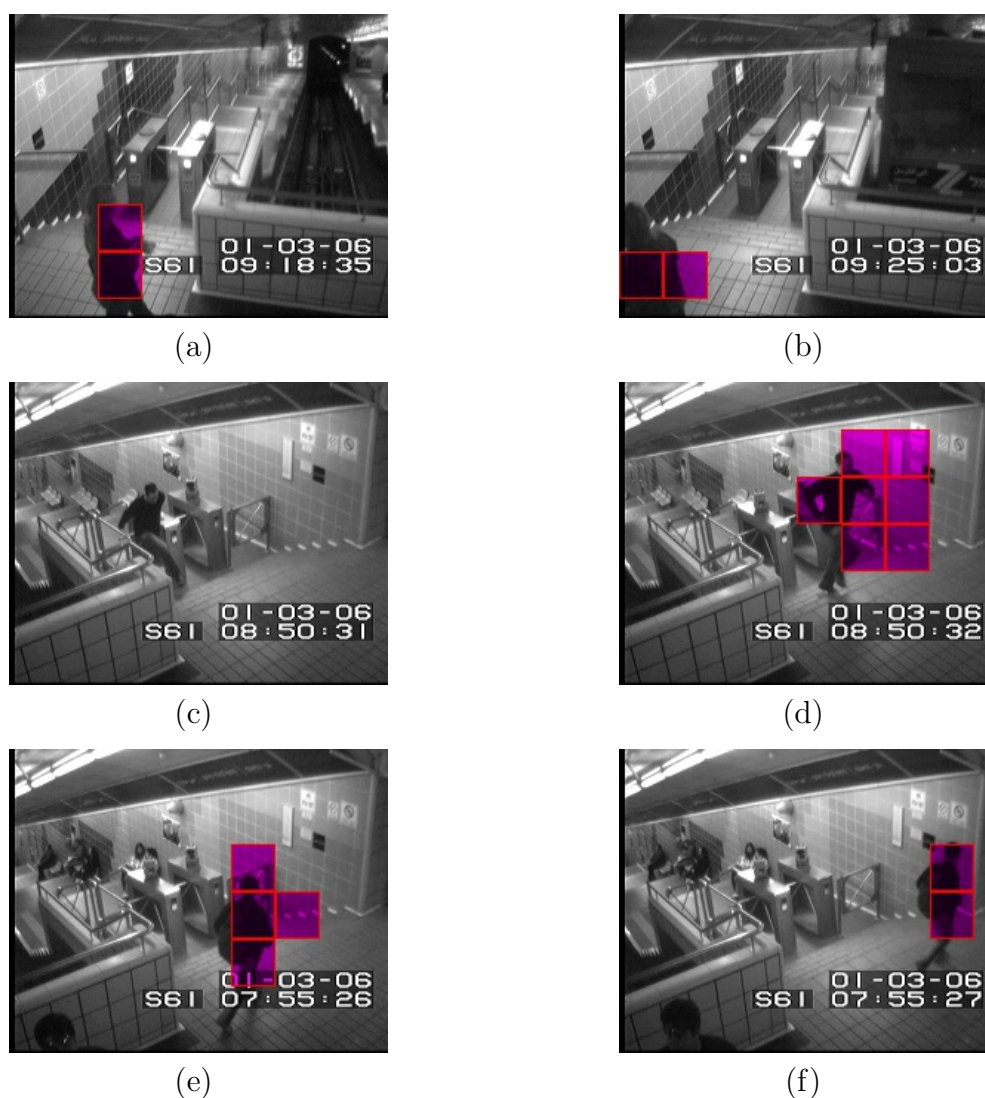


Figure 4.11: Examples of false alarms samples for the subway clips. In first row two mistakes of position are presented, in second row the algorithm can recognize the anomaly, however after some time, in third the boy running which is anomalous, nonetheless it does not appear in the ground-truth.

Table 4.2 shows our results and the results achieved using different local feature extraction approaches. On the exit scenario, HOFME achieved an equal error rate (EER) of 17.8% and an AUC of 0.849, outperforming the other descriptors. In this case, HOFM achieved 18.8% and AUC of 0.845. On the entrance clip, HOFME achieved an EER of 22.8% and AUC of 0.816, while HOFM 23.5% and AUC of 0.815. The ROC curves for the two scenarios are shown in Figures 4.13 and 4.12. HOFME outperforms the HOFM, HOOF, MBH and HOG3D descriptors.

Table 4.2: Anomaly detection AUC and ERR (%) results of Subway dataset.

Approach	Exit		Entrance		
	AUC	ERR (%)	AUC	ERR (%)	
HOG3D [Kläser et al., 2008]	0.524	48.6.3	0.497	50.1	
MBH [Wang et al., 2013]	0.61	43.4	0.519	48.7	
HOOF [Chaudhry et al., 2009]	0.8	25.1	0.774	24.4	
Our	HOFM	0.845	18.8	0.815	23.5
Results	HOFME	0.849	17.8	0.816	22.8

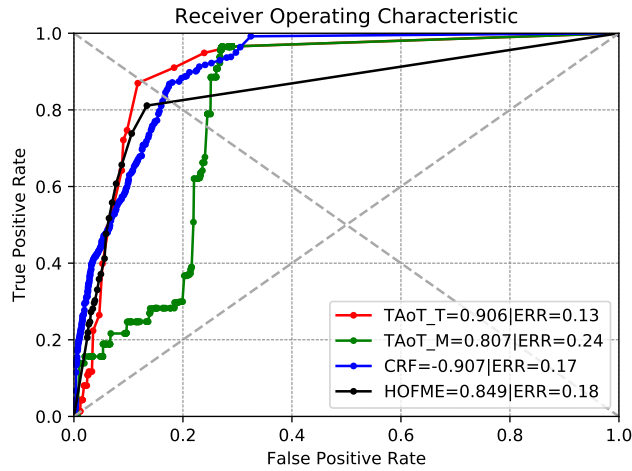


Figure 4.12: ROC curve for exit sequence.

4.3.5 Evaluation on Badminton Dataset

Table 4.3 shows the AUC and EER rate for the Badminton dataset. According to the results, HOFME achieved an AUC very similar to the HOFM and a smaller EER than the HOOF. In Figure 4.16, we can see the respective ROC curve. In this case, the HOFM descriptor achieves the best results because the entropy does not add much discriminative information, since the anomalies are activities such as running and moving in wrong direction, which can be well captured by the HOFM.

Table 4.3: Precision Recall (P/R) results of Badminton dataset.

Approach	Peds1		
	AUC	ERR (%)	
HOG3D [Kläser et al., 2008]	0.5	50.0	
MBH [Wang et al., 2013]	0.539	48.7	
HOOF [Chaudhry et al., 2009]	0.765	26.2	
Our	HOFM	0.806	28.6
results	HOFME	0.798	28.0

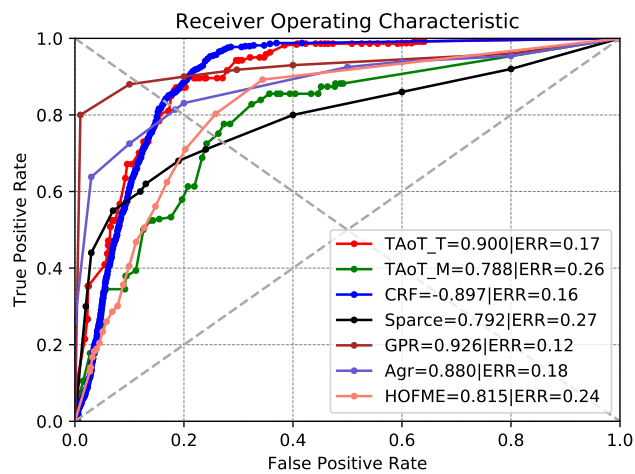


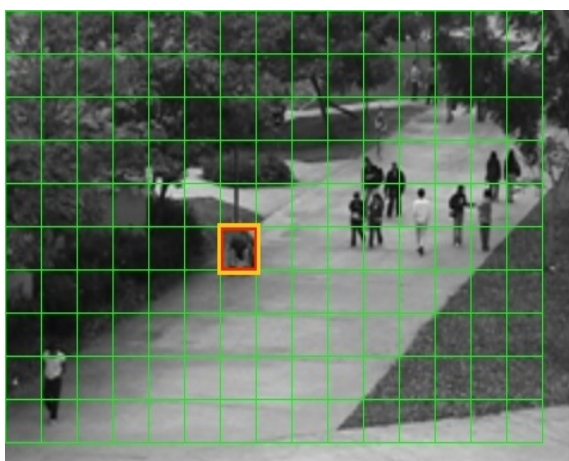
Figure 4.13: ROC curve for entrance sequence.



(a) True positive



(b) False negative



(c) False positive



(d) False positive

Figure 4.14: Examples analyzed through anomaly detection.

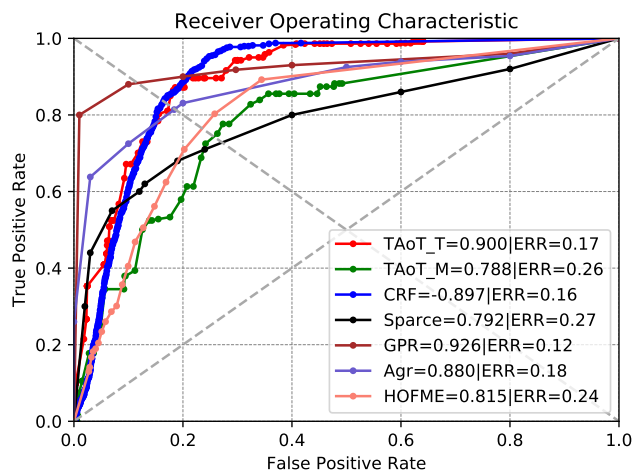


Figure 4.15: Results for Subway Entrance clip.

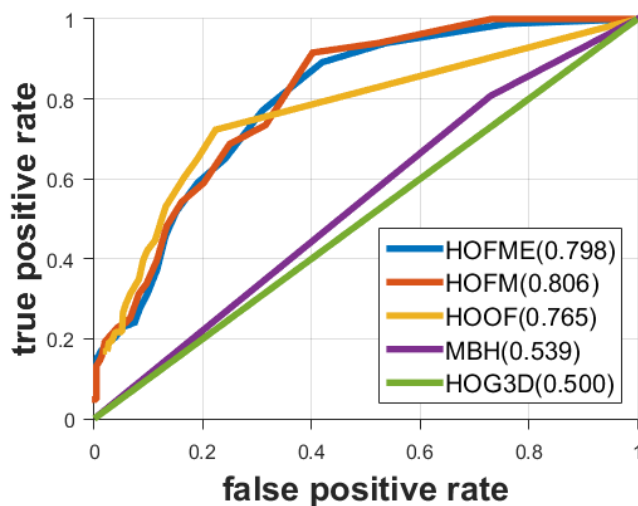


Figure 4.16: ROC curves for the Badminton dataset.

With this experiment, we intended to present a non-controlled environment. Although, the labels are simple, our model recognized accurately most of labeled events. We focused on the region with people on the grandstand, discarding the players in the game and the people that cross in front of the camera. Some examples are showed in Figure 4.17.



(a)



(b)



(c)



(d)

Figure 4.17: Examples of anomaly detections in the Badminton dataset.

Chapter 5

Anomaly Detection Based on Human-Object Interactions

In this chapter, we present our second approach to detect anomalous events. In this model we focus on discovering context patterns that will be used to detect anomalies. The main idea is to create a list of relations between human and objects in many scenes belonging to some context (for instance, a comic book store, computer laboratory). This list contains only normal interactions. In the detection step, two strategies are used to determine the anomalous patterns in the scene.

5.1 Overview

Generally, the intuition to detect anomalous events is based on low level characteristics such as movement, appearance and position. Usually, these types of characteristics belong to a specific environment and are correlated with the camera view. Nonetheless, depending on the context, anomalies could be detected using other information sources.

Anomalies depend on the scene context and the scenes could be similar to each other. Hence, in some cases, given the same context the anomalies could be the same in various scenes. In fact, this hypothesis comprises the fact that common patterns are similar or the same in various scenes that share the same context. To depict this idea, imagine this scenario: “a classroom in the building of the computer science department”. Despite the laboratories, the majority of classrooms in this building share the same context, and probably in most cases anomalies also, for instance, a crowd running to the exit, people walking after midnight and other suspicious situations. Based on this premise, information from a specific scene could be used in similar scenes. Using the past premise, some type of patterns can be employed in a different scene, therefore, our hypothesis is *the extraction of normal patterns in a specific scene could be used to detect anomalies in other scenes that share the same context*. Note, we refer to the

context to that information specific to the scene as are the elements and rules that make it up and give semantic meaning to a certain situation that happens in a certain place and time.

In Li et al. [2015], the information is extracted and used for the same scene. This is logical because the goal is to detect anomalies in a specific context. However, many of research are based on fixed regions on the frame. This type of information is highly correlated with camera position; thus, for instance, abrupt camera movement or zoom may change the scene and consequently the extracted information for this scene cannot be used in the new camera view. Then, information extracted from fixed regions, specially low level features, is not suitable to use in other scenes. Although it is an obvious affirmation, the idea is to highlight the importance of source information to determine what kind of patterns could be used to represent the events in different scenes.

Labeling scenes [Byeon et al., 2015; Johnson et al., 2016] is a very intuitive way to describe images. However, this type of description is limited to the knowledge of the learning model, because there is infinity number of possible situations. Following similar intuition, scene description is a good clue to describe events, specially objects that are part of the scene and people that perform the event. Therefore, our second hypothesis is *the interaction between humans and objects in the scene represents important information of the context*.

Before presenting the proposed approach, one important aspect must be discussed: the types of anomalies our approach intends to detect. Our model is oriented to detect anomalies in different environments, implying that typical information employed for anomaly detection (*e.g.*, velocity and orientation captured through optical flow), cannot be used due to the change of camera position. To overcome this problem, our approach describes the scene as a set of interactions between persons and objects, instead. Thus, our proposed model fits is oriented to detect anomalies in scenes where person’s pose can be recognized, it means, camera must be near enough to the people in the scene for a correct pose detection and object linking.

5.2 Proposed Approach

Our approach is divided into two main steps: *anomaly representation* and *anomalous pattern detection*. In the first step, a preprocessing stage allows our model to describe the sequence creating structures to represents interactions. The idea is to collect these interactions as patterns belonging to a given person. In the second step, the extracted

patterns are compared with learned pattern in the training phase to discover whether some of them may be considered as anomalous. The model collects a set of normal patterns based on interactions. The training step consists on vocabulary and probabilistic model generation. Then, in the testing, patterns found in training step are used to detect if a person is performing any anomaly event. Here, it is important to highlight that video sequences belonging to training set could be completely different from testing sequences. However, all of them have the particularity of having the same context. Figure 5.1 illustrates the proposed approach.

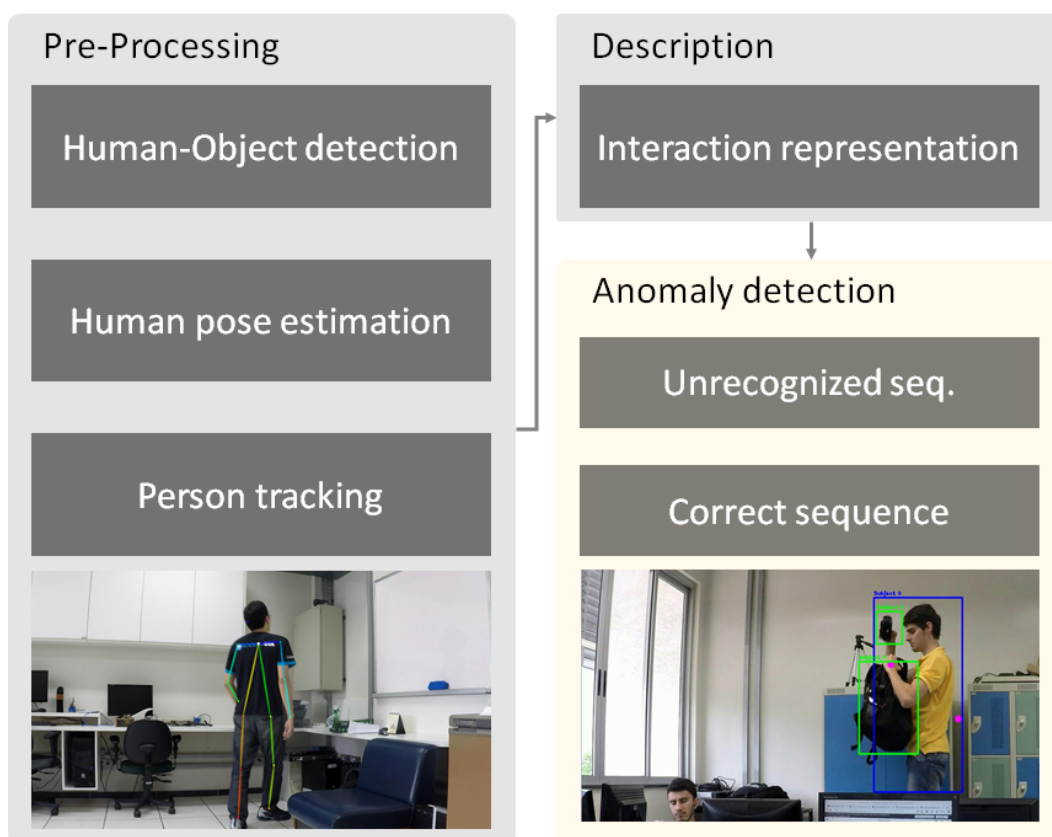


Figure 5.1: Diagram illustrating the proposed approach to perform anomalous event detection using human-object interaction. The box in gray color indicates the steps for anomaly representation (training and testing) and the box in yellow indicates step for anomalous pattern detection (testing only).

5.3 Anomaly Representation

To create human-object representation, actors (people and objects) must be located. Afterwards, our approach determines the interaction between a person and objects. We employ a human pose estimation approach [Eichner et al., 2012] to locate the

person’s hands. This is important because, we are looking for objects that are near to the person’s hands, and consequently, interacting with the individual. After that, we estimate people’s tracklets using a Kalman filter.

The tracking algorithm on this approach is simple. This heuristic creates a simple representation of a person tracklet. Our model creates a Kalman model for each bounding box in the scene, then using the Kalman model we predict the following position of the bounding box. We use as reference point the center of the bounding box. In the next frame, our model associates the new samples of bounding box with the last set of tracklets.

Once the tracklets have been defined, the next step consists in linking the objects with the people. However, recognizing when a human-object interaction happens is a difficult task, especially because the model does not have any depth information (single camera). Hence, we propose a heuristic model to link human and objects by employing the position of the hands based on the distance between objects and the person. If the distance is smaller than a threshold δ_2 , the model joins or relates this object with a person. This distance is computed from the hand to the perimeter of objects bounding box. When the pose estimation does not provide the hand position, our model determines a hand’s position using a straightforward heuristic. First, our model divides the bounding box into a grid of 3×3 . We assume, the middle row contains three blocks and the hands are the middle point in the exterior blocks of this row.

To create a structure that represents the object interactions with a person, we employ a graph representation, with connections between people and objects. Here, each interaction may be represented by a label meaning the set of objects that this specific interaction contains. For instance, if a person interacts with a chair and table, the structure is named as *chair-table*. Therefore, the objects which are in the interaction set give the name to the structure at *frame_i*. Thus, we can see the tracklet representation as a list of structures formed by square and circle nodes. Figure 5.2 illustrates the representation, where square node represents the person belonging to the tracklet and circle nodes represent the objects that have been linked with such person. Note that objects may appear many times, however the label interaction is the same. This is done by a unique interaction label composed by the sorted object identifiers. Further, Figure 5.3 presents another example of this process. At a given frame, the interaction for one person are described using a graph representation, we call to this representation, interaction structure. In the example, the interaction structure is the “word” [blank-chair-chair]. The token blank refers when there are no objects interacting with the person.

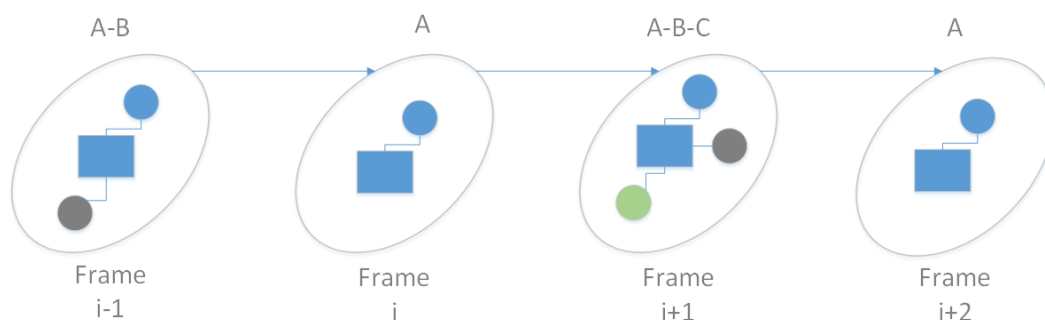


Figure 5.2: Interaction representation from a person tracklet. Squares represent the person at determinate frame and circles represent the linked objects, different colors indicate different objects. Letters on top of structures are the word representation for this interaction. For instance, in Frame $i - 1$, there are two objects: A and B.

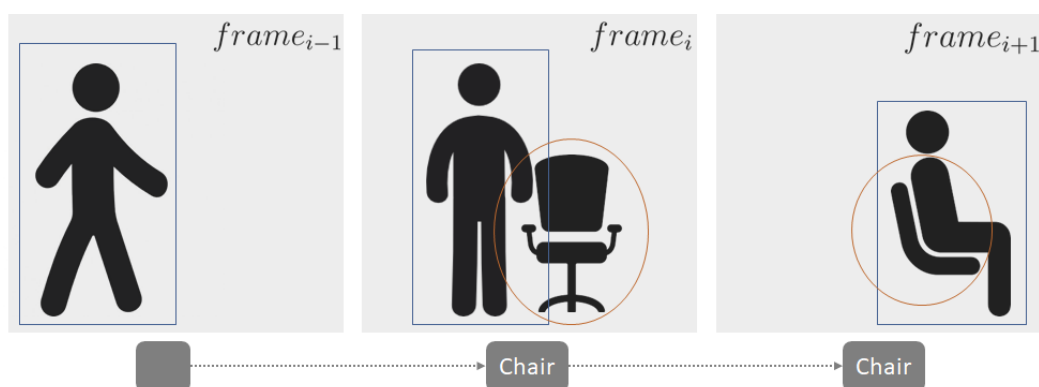


Figure 5.3: Graphical example of human-object interaction. It is a simple representation, where, there are only a single person with one object. The label for this interaction structure is [blank-chair-chair].

The aim of this part is to describe the scene, in such a way that objects and humans are detected. For instance, using the past classroom example, we can imagine the following scene: “*there are two students walking in the classroom through tables and chairs and one of them takes a book from the last table*”. In this example, there are two people, chairs, tables and book, and one person interacts with a book in the table.

5.4 Anomalous Pattern Detection

The idea of this part is to use the learned information in some context and detect the anomalies from this knowledge in another view. For instance, the model may collect the normal interactions in two classrooms in the university and attempt to detect anomalies in other classrooms. To accomplish this goal, the model looks for two type

of information, interactions that have not been observed in the training phase and from some sequence of interactions that have not been seen.

Given the interaction structures, the next step is to detect anomalous patterns on the testing phase. At this stage, our model is composed by two different strategies: (i) unrecognized interactions, and (ii) sequence of interactions. The goal of these strategies is to recognize anomalies by learning contextual information. The solution of our model is a combination of these strategies.

5.4.1 Unrecognized Interactions

In our model, this strategy represents the first level of anomaly detection, once model aims to recognize atomic structures of anomalies. The idea is simple, during the training phase a list of interactions is built, then, if some interaction is not present in this list, the structure is label as anomalous. This strategy helps to recognize when an object, or a set of objects, present in the scene has not been seen during the training phase. As an example, during training phase, in a computer laboratory, nobody interacts with the fire extinguisher, if such interaction happens in testing phase, this could represent a hazardous event and should be regarded as an anomalous event.

5.4.2 Sequence of Interactions

Based on [Crispim et al., 2016], this strategy explores statistical information regarding sequence interactions. Similar to probabilistic language model, the idea here is to detect some events given their occurrence probability. For this specific case, we are interested in events with low probabilities (anomalies).

An n-gram model [Jurafsky and Martin, 2016] is a type of probabilistic model that allows to make a statistical prediction of the next element of a certain sequence of elements that has happened up to now. An n-gram model can be defined by a Markov chain of order $n - 1$. More precisely, an n-gram model predicts x_i based on $x_{i-1}, x_{i-2}, \dots, x_{i-n}$. Due to computational limitations and the open nature of problems (there are usually infinite possible elements), it is usually assumed that each element depends only on the last n elements of the sequence. The two main advantages of this type of models are: relative simplicity and it is easy to expand the study context by increasing the size of n . In our specific case, a n-gram computes the probability of determinate occurrence using maximum likelihood

$$P(w_i|w_1, w_2 \dots w_{i-1}) \approx \frac{\text{count}(w_{i-1}, w_i)}{\text{count}(w_{i-1})},$$

where, w_i represents a label of structure in the graph representation of interactions which is built by square nodes and its related circle nodes (see Figure 5.2 and Figure 5.3) and *count* returns the number of times a determinate word occurs. During the training phase, we collect the words and word pairs to be used in this step. In our model, each word is represented by a structure label, *i.e.*, the object set that interacts with the person in a determinate frame. Finally, if the probability $P(w_i|w_{i-1})$ is smaller than a threshold η (0.6 in our experiments), such interaction and consequently the observed tracklet, is marked as anomalous.

5.5 Experiments

We divided the experiments into two parts: (i) tests based on unrecognized interactions; and (ii) test based on the sequence of interactions. We use the same set of videos for evaluating both approaches for detecting anomalous patterns. The hardware employed in these experiments has the following configuration: Intel(R) Core i7(R) 4960x @ 3.6GHz processor, 64 GB Kingston DDR3-1600MHz of ram memory; one hard disk Seagate 1.5TB; and one Geforce Titan X graphic card. All the framework were coding using python libraries.

Interaction Objects Dataset

This dataset was built from different views captured in distinct laboratories. We created a set of training videos to represent common events, including people interacting with chairs, laptops, backpacks and monitors. The dataset is divided in training and testing video sequences. The ground-truth is composed by every event that was not present in the training videos (*i.e.*, anomalous event). It is important to highlight the diversity not only on camera view but also in environments. Our dataset contains clips captured in different places and views for the same context which is a computer laboratory. For instance, various laboratory sequences and clips were recorded with different camera position in the room. In general, there are three or four people at most in the scenes.

Our proposed dataset is composed of everyday laboratory activities. For instance, students sitting, reading, coding, etc. In the camera view, people enter and pass in front of the camera. The videos do not exceed five minutes of recording. The dataset is composed by 11 clips for testing and 20 for training. The image frame in the videos has 1280×720 . In the ground-truth we consider as normal events where people interact with common computer laboratory objects including chair, table, notebooks, computers and books. As anomaly samples, we considered the interactions that a person performs with

the coffee machine or a pillow, which we chose to be the anomalous events. We also considered the sequence of events to determine if it is anomalous or not. Naturally, anomalous events are cases that do not happen continuously. Hence, we created a ground-truth and labeled some specific events for each set of tests.

5.5.1 Parameter setting.

In the first stage, our model detects actors and human poses. For this approach, we used the precomputed models provided by [Liu et al., 2015] and [Eichner et al., 2012]. These networks were trained by the authors, we just used the model provided by them, and run the algorithms with their default values; both solutions are coded in Keras [Chollet et al., 2015]. Our approach is limited by the objects that can be detected by the object detector. In the second stage, we have two variables: (i) tracklet building variable $\delta_1 = 50$; and (ii) Jaccard index $j = 0.7$. The Jaccard index is a complement to our tracking model since in some events the bounding boxes of certain person may vary by its pose, i.e., if this person stretches arms his bounding box will grow and only a distance threshold may give a wrong answer. Having on view of this, we used an occupation criterion based on overlapping areas. The configuration of Kalman filter is the default presented in the OpenCV framework [Bradski, 2000]. These values are chosen given the distance and the image view, in which almost the entire body appears in the camera view. Hence, the variables for this approach also depends on the camera view.

In the pattern description step, interaction representation for a particular tracklet is built. Here, we have a threshold $\delta_2 = 100$ to link objects with a person tracklet. The δ values are in pixel unity.

5.5.2 Metrics.

To evaluate the detection results, we use the metric proposed in [Cao et al., 2010]. Ground truth anomalies are denoted by $Q^g = \{Q_1^g, Q_2^g, \dots, Q_m^g\}$ and output results are denoted by $Q^d = \{Q_1^d, Q_2^d, \dots, Q_n^d\}$. The function $HG(Q_i^g)$ denotes whether a ground-truth interval Q_i^g is detected. Function $TD(Q_j^d)$ denotes whether a detected interval Q_j^d is relevant. $HG(Q_i^g)$ and $TD(Q_j^d)$ are judged by checking whether the Jaccard index is

above a threshold th (0.30 in our experiments). Using

$$\begin{aligned} HG(Q_i^g) &= \begin{cases} 1, & \text{if } \exists Q_k^d, \text{ s.t. } \frac{Q_k^d \cap Q_i^g}{\|Q_i^g\|} > th \text{ and} \\ 0, & \text{otherwise} \end{cases} \\ TD(Q_j^d) &= \begin{cases} 1, & \text{if } \exists Q_k^g, \text{ s.t. } \frac{Q_k^g \cap Q_j^d}{\|Q_j^d\|} > th, \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (5.1)$$

the precision and recall values are defined as

$$\begin{aligned} \text{Precision} &= \frac{\sum_{i=1}^m HG(Q_i^g)}{m}, \\ \text{Recall} &= \frac{\sum_{j=1}^n TD(Q_j^d)}{n} \end{aligned} \quad (5.2)$$

To better understand this metric, imagine two binary arrays, one for ground-truth and the other for the answer or prediction, the intervals are marked with ones. The intersection is measured bin by bin between these arrays using “and” operation.

Figure 5.4 shows simple examples of our model when an anomaly is detected. Subjects and objects are marked by blue and green bounding boxes, respectively. The hand positions are marked as pink circles. Figure 5.4(a) and 5.4(b) show examples of anomalous events. In these cases, the subject is removing the camera, which is considered as anomalous since the camera was placed in that position during the training phase, and nobody took it out of that place. Figures 5.4(c) and 5.4(d) show examples of unrecognized interactions. During the training these objects were not seen.

5.5.3 Unrecognized Interactions

These experiments are oriented to detect anomalies regarding unrecognized objects. As we mentioned in the dataset description, we introduced some objects in the scene that are not part of the laboratory. The idea of this experiment is to show that our model is able to recognize an anomalous event caused by a “suspicious” interaction. One can say that such binary problem is easy to solve when we talk about only objects that do not appear during the training. However, it is more than only unrecognized objects since we are also looking for unknown interactions. For instance, interactions with laptops, chairs, notebooks and backpacks are common in computer laboratories, nonetheless, some combinations such as computer and coffee cup may result in an unseen interaction that should be classified as an anomalous event. Another goal of this experiment is to determine interactions with unrecognized objects (e.g, people interacting with weapons may be considered an anomalous situation in a laboratory because such interactions are rare or previously unseen).

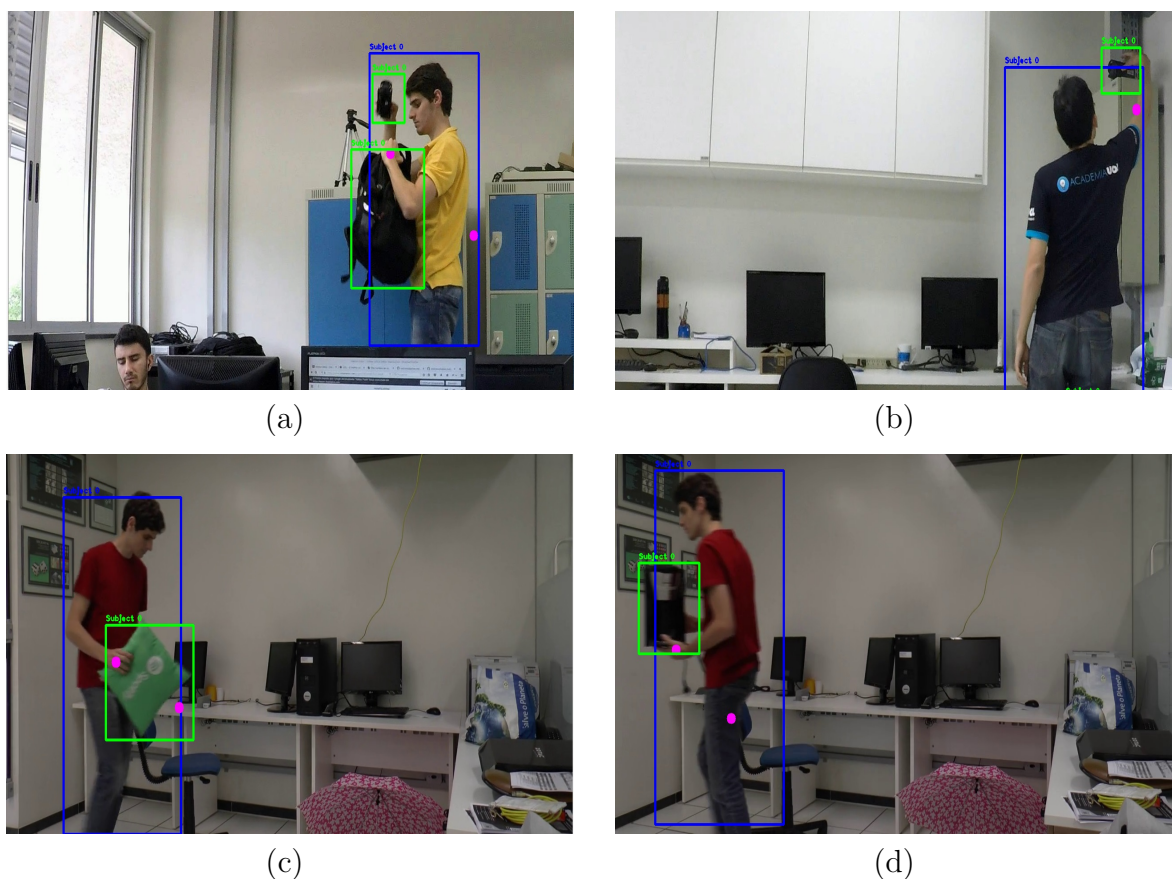


Figure 5.4: Examples analyzed through anomaly detection using human-object approach.

Table 5.1 presents the results of such experiments, which are composed by four test events. All tests focus on the object that does not belongs to the scene. As discussed in Section 5.4, we present the results following two different strategies: (i) unrecognized interactions; and (ii) sequence of interactions. Precision and recall values are presented as tuple (P, R) . According to the results, tests related to the detection of anomalies due to unknown interactions presented positive results in the first level of detection. Moreover, this strategy was satisfactory to detect anomalous sequences. However, test 3 and 4 presented a lower value of precision and recall than expected for both methods.

The reason for the low accuracy in test three is that we do not have the information regarding the depth of the image. Due to the perspective projection, a person with their hands far from an object in the depth can be considered as an interaction. The low recall in test 4 occurs because the pose estimator sometimes fails to detect the coordinates of the person hands, thus, although the person is near to the object the reference point of the hand appears far, this problem happens specially in occlu-

sions. Another mistake occurs when the distance between the hands of the actor and the objects exceeds the threshold distance between hands and objects that defining an interaction. This problem happens because the threshold was small to detect the interaction. This is a problem of our model because this parameter is variable in the context and the image view.

	Strategy 1	Strategy 2
Test 1	1/1	0/0
Test 2	1/0.5	1/0.5
Test 3	0.25/0	0.25/0.1
Test 4	1/0.11	1/0.07

Table 5.1: Precision and Recall results of human-object interaction dataset. Only unrecognized objects are present in these sequences.

5.5.4 Sequence of Interactions

Here, our main goal is the temporal information regarding the sequences of interactions that have not been seen before (e.g., to enter a bank office you first need to pass through a metal detector security door). The results are presented in Table 5.2. As it was expected, N-grams (Strategy 2) achieved the best results for detecting anomalies for sequences of events.

	Strategy 1	Strategy 2
Test 5	0/0	1/0.5
Test 6	0/0.5	0.5/0.5
Test 7	1/0.5	1/1
Test 8	0/0	1/0.33
Test 9	0/0	1/0.5
Test 10	0/0	1/1
Test 11	0/0	1/0.5

Table 5.2: Precision and Recall results of human-object interaction dataset. Only unknown sequences of interactions are presented in the clips.

The next paragraphs discuss important aspects regarding the proposed method to detect anomaly based on sequence of interaction human-objects.

Dependency on the low-level tasks: It is important to highlight that our goal is to introduce a new approach to recognize and detect anomalies, instead of using only a specific environment, our proposed approach attempts to learn context based on

human-object interactions. However, we deal with some artifacts and mistakes due to the object detector, tracking and pose recognition approaches that could be overcome by employing better low level approaches to generate tracklets.

Human-object interaction recognition: It is clear that only using a distance to link humans and objects is not the best strategy. However, we did not include depth information since surveillance commonly uses a single camera for certain environments. Our approach is quite simple in this stage, nonetheless, human-object interaction in video sequences is part of our future works.

Dataset and anomaly cases: Even though it is not our main goal, we also introduce a new sequence for anomaly detection. As future work, we intend to create a platform to progressively increasing the anomaly dataset.

On-line vs. off-line approaches: Other approaches such as [Javan Roshtkhari and Levine, 2013; Yuan et al., 2015] learn scene patterns to determine anomalies while the sequence is presented to them. However, an important disadvantage of these approaches is the training time. Moreover, some sequences start with the anomaly in the beginning making these approaches to learn it as a normal situation (e.g., UCSD dataset [Lab, 2014], specifically in Peds2 camera view, where many of test anomalies starts at the sequence beginning). Instead, off-line approaches attempt to learn as much as possible to avoid miss some anomalies.

Dataset annotation bias: The common underlying assumption behind anomaly detection is that anomalous events occur with low probability when data is collected. Then, when one annotates a ground-truth is natural to have a bias due to subjective observation of the scene, which makes the progress on this problem even harder.

Testing with other datasets: In literature, there are a few well-known datasets. However, these ones are pretty specific for crowds with low resolution. Nevertheless, our approach is oriented to recognize human-object interactions and in most cases, this type of relations are not clear given the clarity of the video sequences.

Comparisons with other approaches: As our approach does not have representative results in other datasets, other algorithms cannot fit in our proposed dataset. It is due to the type of representations which are based on characteristics like: speed, orientation, trajectories, appearance, textures. This type of features changes significantly scene from scene, leaving without representativeness that type of characteristics when the environment changes.

Chapter 6

Anomaly Detection based on Trajectories

Many studies focus on creating representations robust to most types of situations, i.e., models that fit in most of cases. Typically, the problem is formulated as a detection task where a model learns patterns from normal data to detect events that do not fit with them. Classical models usually choose to represent events based on spatiotemporal information, which is more prone to issues related to noise from complex backgrounds and illumination changes. This may be addressed using high level information (e.g., object detections), to model abnormal movements directly. In this chapter, we present an approach based on human trajectory description to detect anomalous events.

6.1 Overview

Abnormal event detection for video surveillance refers to the problem of finding patterns in sequences that do not conform to expected events [Du et al., 2013]. It is a challenging problem because the definition of anomaly is subjective to the particular scene context, giving origin to a large number of possibilities. For instance, someone running at a marathon is a normal event, while someone running during a regular working day might be due to an emergency, an anomalous event. Therefore, the difficulty of anomaly recognition is related to the semantics that are observed in the scene.

The complexity in defining algorithms that suit in any case is a hard problem. Indeed, most models focus on extracting features based on movement and appearance from spatiotemporal regions [Popoola and Kejun Wang, 2012]. Nevertheless, this type of information might be affected by noise due to complex backgrounds, illumination changes and poor lighting conditions. With new trends in computer vision, this issue

can be minimized by using higher semantic information, such as object detections and pose estimation to model anomalies directly from people movements.

In this approach, we exploit high level information to create a robust representation for anomaly recognition. Our approach model people’s movements by leveraging reference points from body skeletons obtained through a state-of-the-art pose estimator. The reference points are aggregated through time building a trajectory. Each trajectory is then represented using deep neural networks to better encode its morphology. Our hypothesis is that trajectories are able to encode the necessary information from movement to recognize certain anomalous events. Thus, our approach describes the trajectories to find a representation that encodes the people movement, this kind of feature is extracted from flow and morphology of the human trajectories in the scene. This approach aims to find a trajectory representation, then the anomalies that can be detected are strong related with people movement only. Thus, while trajectories could be extracted then our model may detect anomalies, it is important to determine the environment in which our model fits well to detect anomalies with success.

In addition to being more robust to the aforementioned issues, trajectories may also be used for people behavior analysis. To validate such application, we also evaluate the rarity of trajectories using clustering models. An advantage from using trajectories is that the localization of the particular individual performing an anomalous event is easily retrieved. Furthermore, trajectories may aid behavior understanding of pedestrians [Zhou et al., 2012]. It is important to highlight that the proposed model is oriented to scenes where people detector and tracking algorithms may offer a good representation; thus, high crowded scenes are not considered in the scope of this chapter. This approach intends to detect anomalies in static camera view.

6.2 Proposed Approach

In this section, we present the proposed approach for anomaly recognition comprising four main steps: (i) pose estimation, (ii) tracking building, (iii) feature extraction, and (iv) anomaly and rare trajectory recognition. In the first step (Section 6.2.1), the goal is to obtain a reference point of people within the scene to obtain a concise representation. In the second step (Section 6.2.2), the model creates for each reference point a tracklet and generates a set of trajectory points, which may be normalized depending on the number of points. Then, in the third step (Section 6.2.3), we propose two feature extraction models, whose input are normalized trajectories, generating a feature vector. At the end (Section 6.2.4), our approach outputs two predictions, the anoma-

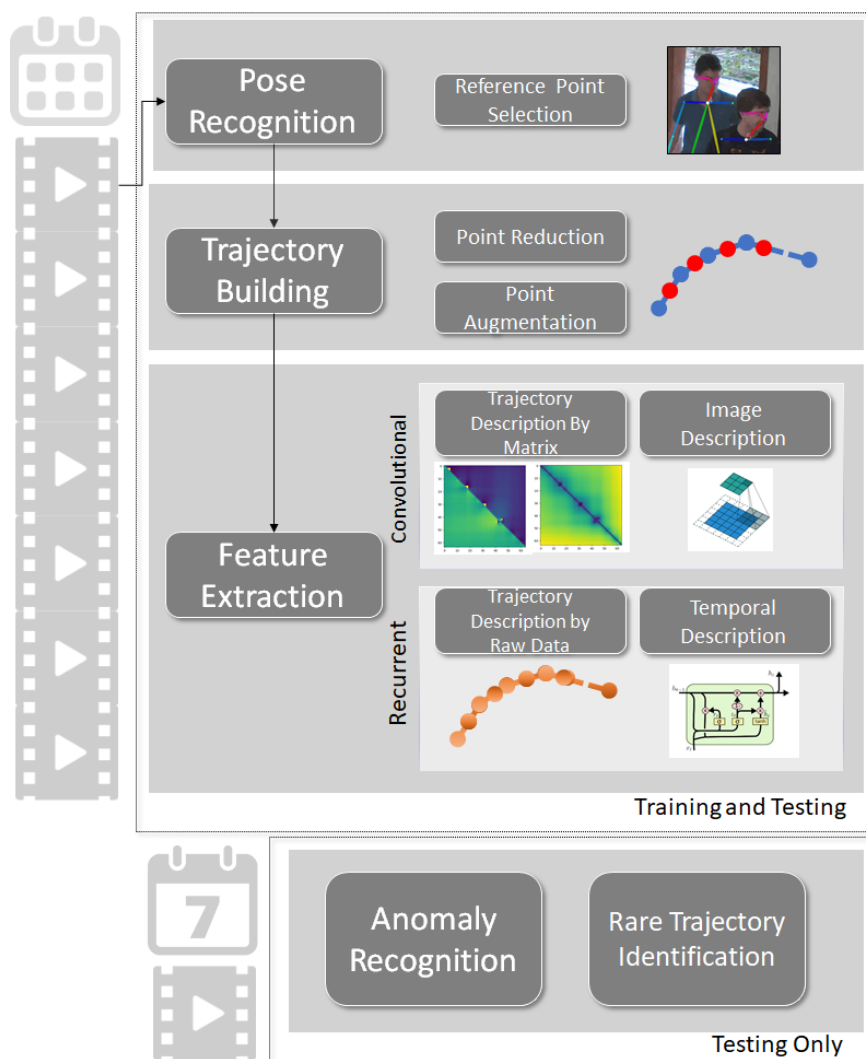


Figure 6.1: Overview of our approach. Given a body skeleton, we select reference points that are used to build trajectories. A sequence of such reference points consists of a trajectory. Then, we describe the normalized trajectories using two different techniques, a convolutional descriptor based on CNN or a recurrent descriptor modeled using a RNN. During the testing phase, we recognize anomalies and rare trajectories by comparing the descriptors extracted from each test sample regarding the trained model.

lous trajectories and the rare trajectories, each predicted by a corresponding method. The general workflow for training and testing are the same until the feature extraction step. Stage four addresses the anomaly recognition and trajectory discrimination and these are exclusive for testing. Figure 6.1 presents an overview of our approach.

6.2.1 Pose Estimation

Given a video sequence V as input, the goal of this step is to detect people in the scene and to find a reference point for each. The reference point represents a person and will be used to create a trajectory point. In the literature, we can find accurate object and person detectors [Redmon and Farhadi, 2017; Liu et al., 2016; Lin et al., 2017]. These detectors provide the bounding box of the detected person/object. Although, the person is inside the bounding box, the four points do not represent a reliable point to define as a reference point of a person due to the variation in size of the bounding boxes. For instance, a person with stretched arms will have a larger bounding box than a person with closed arms. Another case where bounding box coordinates are not reliable as reference points is when the detector detects a group of persons and their bounding box changes in size in every frame. The aim of finding a reference point is to define trajectories that represent people movement with more fidelity.

In our approach, we define the reference point of a person as the joint point between body and head. We use this point because it rarely presents independent movement from the whole body. To find this point, we use a multi-person pose estimator [Cao et al., 2017]. This model extracts the person skeleton and chooses the point that corresponds the joint between head and body parts. Before continue detailing the model, we present some definitions. A *reference point* $p_k = (x, y, t)$, where x and y are the position in the image and k is the ID for the reference point at frame t . Thus, at frame $F_t \in V$, the set P_t contains the reference points found at time t . Hence P is the set that contains all P_t . *Trajectory Point* $pt_i = (x, y, t)$, where x and y are the position in the image and i is the ID point that belongs to trajectory j at time t (the reference and trajectory point are the same). We use these two definitions explaining the following methods, where points may belong either to a set of reference points or to a trajectory.

6.2.2 Trajectory Building

The next step of our approach is to create the trajectories for each person. The goal is to connect reference points, relating them frame by frame and labeling the set with a person identifier. Literature presents many models for tracking objects [Watada et al., 2010; Čehovin et al., 2016], where most of them are based on data association approaches for single [Čehovin et al., 2016] and multi-tracking objects [Dehghan et al., 2015]. In the case of surveillance videos, the method must deal with multi-object tracking, which is a np-hard problem [Betke and Wu, 2016]. In this study, we introduce

an algorithm that aims at offering a straightforward alternative to complex multi-tracking models. This method is presented in Algorithm 4.

Algorithm 4 Trajectory Builder algorithm.

```

1: procedure TRAJECTORY( $P, |V|$ )
2:    $P = \{P_i | P_i = \{p_k^i\}_{k=1:np_i}, i = 1 : |V|\}$ 
3:    $P$  is the set point for whole video  $V$ ,
4:    $np_i$  is the number of points at frame  $i$ ,
5:    $|V|$  is the number of frames of video  $V$ ,
6:    $Trks = \{\}$ ,
7:   for  $i$  until  $|V|$  do
8:      $S \leftarrow \text{ComputeScoreMatrix}(P_i, Trks)$ ,
9:      $M \leftarrow \text{Munkres}(S)$ ,
10:     $Trks \leftarrow \text{Matching}(P_i, Trks, M)$ 
11:     $\text{Update}(Trks)$ 
12:  return  $Trks$ 

```

The procedure receives as input the set of points P computed in previous step and the number of frames $|V|$ of video V . $Trks$ is the trajectory set, which is initialized empty. In the first state of the main loop (line 8), the model computes the score between reference points that belong to P_i and current tracklet set $Trks$. An element of $Trks_j \in Trks$ is a tuple $(pt_i | i = 1 : n^{t-1}, pr_j^t, Km)$, where the first element is the trajectory points that belong to tracklet j at time $t - 1$ being pt_i^{t-1} the last point inserted in the set, the second is the predicted point using Kalman filter for time t , and finally the third is the Kalman model for this particular tracklet. To avoid excess nomenclature, point $p_k \in P_i$ will be just p , pt_i^{t-1} will be l (last) and pr_j^t will be pr (predicted). To compute the score point l is subtracted from p and pr in such a way that l is considered as coordinate origin. Let be the result of $r = (\vec{p} \cdot \vec{pr}) / \|\vec{p}\| \cdot \|\vec{pr}\|$, this value is truncated between $[-1, 1]$. Thus, the angle between p and pr is $\theta_{k,j} = \arccos(r)$ is in the range $[0, \pi]$. The final score is given by

$$scr(p_k, Trks_j) = \begin{cases} \theta_{k,j} & \text{if } \delta_1 < th + bf, \\ \theta_{k,j} \times \tau & \text{if } \delta_2 < \|\vec{pr}\|, \\ \pi \times \tau & \text{otherwise} \end{cases} \quad (6.1)$$

where $\delta_1 = \|p - pr\|$ is the distance between candidate point p and the predicted point pr , $bf = th / (2 \times |Trks_j|)$ is a value that is inversely proportional to the number of elements in the set $Trks_j$, $\delta_2 = \|p - pr/2\|$ is the distance from point p and the middle point of l and pr , as l is the origin of coordinates then, this point is half pr , the variable th is a threshold value that is set before the process and depends on the size of people

in the video, and $\tau \geq 2$ is a penalty value.

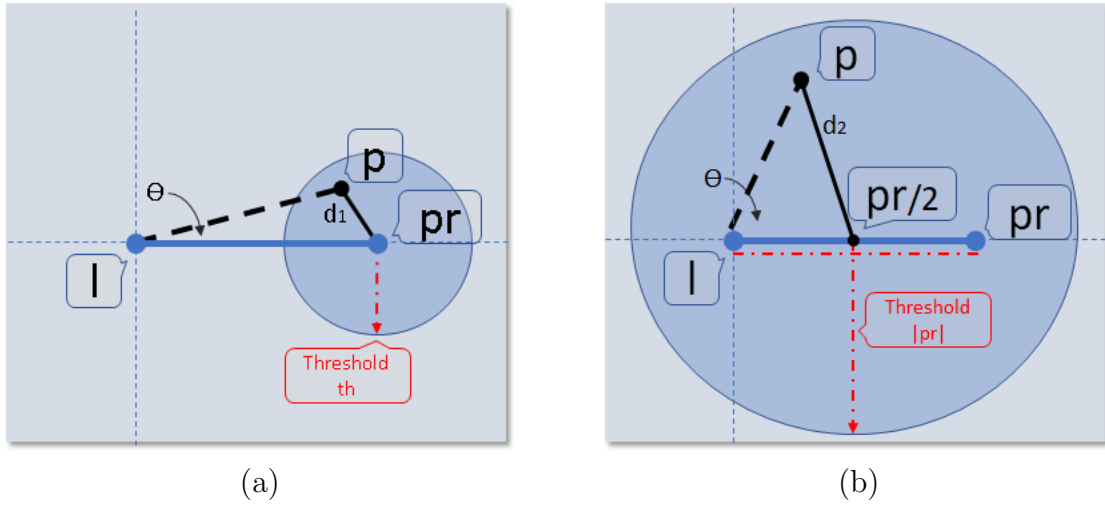


Figure 6.2: Score function examples. At left is the situation when the point p is near to point pr . At right is when point p is near enough to last point on tracklet and predicted point pr .

The idea of the score function is to assign a low value to the compared point that is closer to the predicted point, as we can see in Figure 6.2(a). Using the variable bf , we extend the initial threshold and balance the initial prediction of the Kalman model, specifically for the initial points, where the Kalman model is not stable. Variable bf decreases as more points are in the trajectory. The second case of the score function is when the point is not sufficiently near to the predicted point but it is close to the trajectory flow, as shown in Figure 6.2(b). The idea in this case is to cover a greater region where candidate point could move, including a little region behind the last trajectory point. In the last case, the candidate point is outside of the possible regions of movement.

Matrix $S_{n_{p_i} \times |Trks|}$ contains all the scores between points in P_i and tracklets $Trks$. In the next state, our approach computes the best distribution using Munkres' algorithm [Zhu et al., 2016] to solve the assignment problem [Shah et al., 2015]. After that, the points are assigned to a specific trajectory. Unassigned points create new tracklets. Finally, all trajectories update their information (Kalman model and predicted point), and trajectories that do not present changes within time lapse are closed and saved to avoid confusing with new trajectories.

After every trajectory is collected, the trajectories must have the same number of points to use this information as input for the models. Therefore, we trim or increase the number of points up to a certain value by employing a point reduction or point augmentation process. The problem in the first process lies in the choice of significant

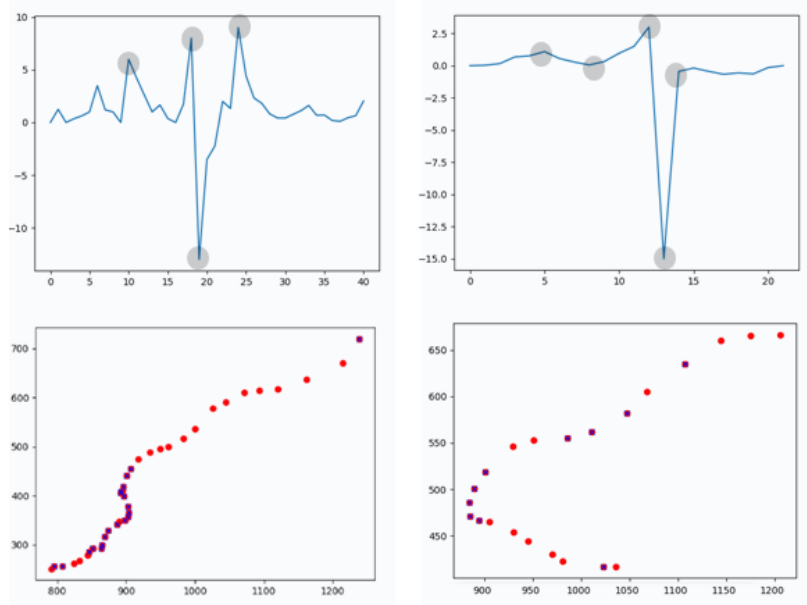


Figure 6.3: Examples for point selection when the number of points have to be reduced. First row corresponds to the first derivative (number of points vs derivative value), circles show some interest points. In second row are marked the selected points in the frame, which for this case has a 1270×720 dimension.

points. To solve this, we select the points that best represent the shape of the trajectory, giving preference to the points where there is more variation, such as curves. To select these points, we apply the second derivative to set of points [Escobedo and Camara, 2016]. Highest values represent the curves in the shape of trajectory. Thus, our approach chooses the interest points by sorting from largest to the smallest the values obtained by the second derivative. Figure 6.3 illustrates the idea of this process, where the first row depicts the images with the first derivatives of the trajectories, the highest values are the key points, and the second row shows the chosen points. Hence, with this heuristic, we reduce the number of points in a trajectory.

For the point augmentation, our approach performs a straightforward strategy. Depending on the number of required points, they are introduced in the middle of two consecutive points. This process is performed initially in the original set of points, if more points are needed, the process is repeated until the number of necessary points is reached. Finally, all the trajectories have the same number of points n . Thus, *Trajectory* is a tuple $T_j = (tid, \{pt_i | i = 1 : n\})$, where tid_i is the trajectory identifier, n is the number of points. A trajectory is formed by ordered set of points in time and is a part of a specific tracklet.

6.2.3 Feature Extraction

Unsupervised representation learning has become an important tool for anomaly recognition. An Autoencoder (AE) is a neural network trained with backpropagation algorithm that provides a dimensionality reduction by reducing the reconstruction error on the training set [Kiran et al., 2018]. Our approach presents two feature extraction models: *Descriptor A*, based on an image representation extracted from a Convolutional Auto Encoder (CAE), and *Descriptor B*, which directly utilizes the trajectory information in a recurrent AE.

Inspired by [Zhang and Lu, 2004], the idea of the first descriptor (Convolutional) is to find a representation that depicts the trajectory as a complete entity (i.e., without segmenting or dividing it). Therefore, the goal is to describe the morphology of the trajectory. To accomplish this goal, our approach saves the variation between each pair of points belonging to a given trajectory into two matrices: angular (AG) and radial (RD), which are square matrices of dimension $n \times n$. The position $AG_{a,b}$ is filled with the angle formed by points $p_a, p_b \in T_j$. Similarly, position $RD_{a,b}$ is filled with the magnitude of the vector formed by points $p_a, p_b \in T_j$. Thus, local information is saved in places that are near to diagonal, as soon as the global information appears closer to the edges of the matrix. The radial matrix is symmetric. In the angular matrix, the values are complements between superior and inferior triangular sections of the matrix. These images are $n \times n$ dimensions. Figure 6.4 presents examples of this matrices for a straight path trajectory.

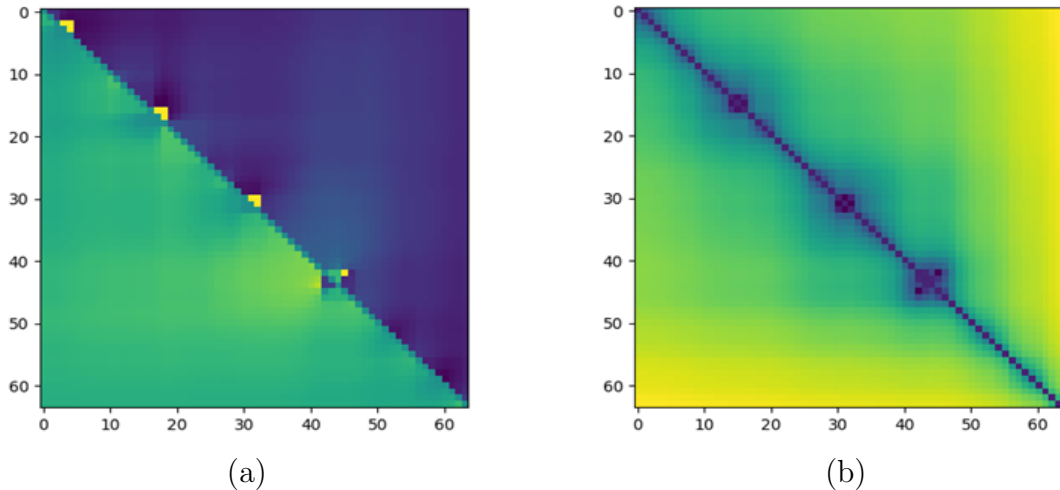


Figure 6.4: Trajectory matrix representation, angular and radial respectively.

The description process for these images is performed by a Convolutional Auto Encoder (CAE). Encoder and decoder are composed by two convolutional layers, each

layer has eight filters, the size of convolutional mask is 5×5 followed by max pooling and up sampling layers of size 2×2 . In the middle of this representation, the CAE architecture presents two fully connected layers, with 512 and 2048 neurons, respectively, the input and output for this segment is flattening and reshaping. The idea of this architecture is to find a semantic representation for angular and radial matrices. This network is trained with only normal trajectory images normalized between $[0, 1]$. After the model computes the weights, trajectory features are extracted from the first fully connected layer (512). The feature vector is the concatenation of outputs of angular and radial CAE. Hence, the final representation is a vector with 1024 dimensions. Figure 6.5 depicts the autoencoder architecture.

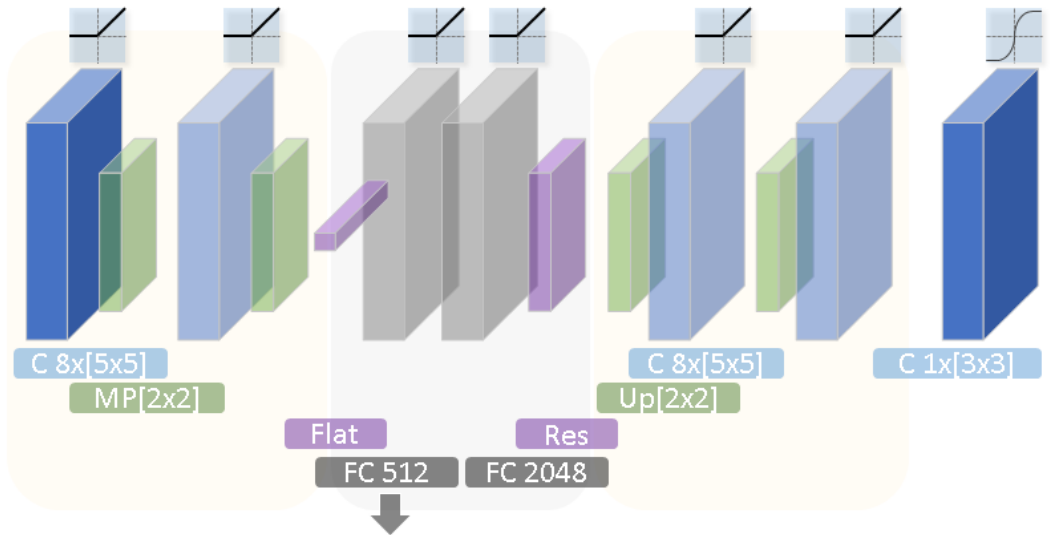


Figure 6.5: Architecture for convolutional autoencoder.

The second descriptor (Recurrent) builds the feature vectors using a recurrent Autoencoder (AE). Similar to the previous model, the idea in this approach is to find an entire representation for the trajectory by correlating the morphology and the temporal information. Thus, the proposed network learns the temporal patterns of trajectory. Composed by only three layers, it begins with a recurrent cell, which in our approach is a Gated Recurrent Unit (GRU) [Chung et al., 2014]. We opted to use GRU instead of Long Short Time Memory (LSTM) [Hochreiter and Schmidhuber, 1997] because they suit better with small training sets [Chung et al., 2014]. The input for this cell is the set of overlapping segments that compose the trajectory. The next element is a fully connected layer with 225 neurons. Both layers, recurrent cell and fully connected utilize sigmoid as activation function. At the end of the pipeline, the model reshapes the output to the same input size, thus the RAE can learn the trajectory patterns. This network is also trained with only normal trajectories. After

the weights are computed, the descriptor for a trajectory is the output of the fully connected layer, a vector with 225 dimensions. Figure 6.6 presents the architecture for recurrent autoencoder.

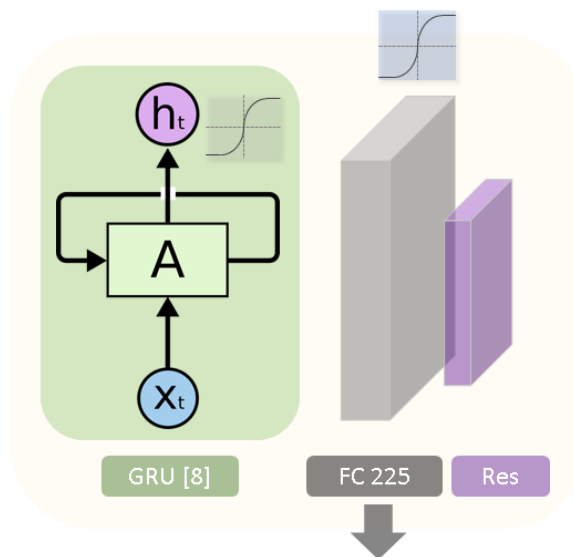


Figure 6.6: Architecture for recurrent autoencoder.

6.2.4 Anomaly Recognition and Rare Trajectory Identification

In the last step, our approach is divided into two approaches: anomaly trajectory recognition and rare trajectory identification, both oriented for video surveillance analysis. While the first addresses the problem of identifying anomalies, the second intends to provide a detection of rare trajectories.

The strategy for anomaly recognition is simple, our approach computes nearest neighbor for each point in testing. Following the Algorithm 3, the idea is the same as our approach based on low level features, where the anomalous patterns are located using the distance to the normal patterns.

In the rare trajectory identification, we suppose that points that represent common trajectories make clusters in the feature space and anomalies are isolated points or they are in groups with few elements. Thus, our approach groups the trajectories using a clustering model Affinity Propagation (AP) [Frey and Dueck, 2007]. An advantage of using AP model is that it does not need to set the number of clusters. Thus, the trajectories are segmented. The rareness of a trajectory is given number of elements of the cluster that it belongs.

6.3 Experiments

In this section, we present our experiments. First, we describe the results regarding anomaly recognition (Section 6.3.2) and then present the results achieved on the rare trajectory recognition task (Section 6.3.3). Experiments were performed on the following datasets: Subway [Adam et al., 2008], Avenue [Lu et al., 2013], Train sequence [Zaharescu and Wildes, 2010], and our proposed video dataset, named Laboratory. The complete framework was developed using Python and Keras [Chollet et al., 2015]. The hardware employed in these experiments has the following configuration: Intel(R) Core i7(R) 4960x @ 3.6GHz processor, 64 GB Kingston DDR3-1600MHz of ram memory; one hard disk Seagate 1.5TB; and one Geforce Titan X graphic card. We performed our experiments in Subway [Adam et al., 2008], Train sequence Zaharescu and Wildes [2010], Avenue [Lu et al., 2013] and our proposed dataset called Laboratory. In following paragraphs we present the datasets, not including Subway dataset which is detailed in Section 4.3.1.

Train Sequence Dataset

The train sequence is part of a set of videos for anomaly recognition proposed by Zaharescu *et al.* [Zaharescu and Wildes, 2010]. This video clip has a view from the interior of a train coach and is the only sequence that contains people in the scenes. It contains 19,218 frames which are very challenging due to drastic variation in lighting conditions and camera jitter. The anomalies in this sequence comprise people coming out and moving on the train.

Avenue Dataset

Introduced by Lu *et al.* [Lu et al., 2013], the avenue dataset contains videos from entrance avenue at the Chinese University of Hong Kong (CUHK) and is composed of 16 training videos and 21 test clips. Testing videos include both normal and anomalous events. It comprises three types of anomalies: running, wrong direction and abnormal object. The anomalous event in sequences abnormal object is a person that pulls up a backpack.

Laboratory Dataset

This dataset contains one month of recordings of the entrance of our laboratory at the Universidade Federal de Minas Gerais (UFMG). The image size is 1280×720 , and the frame rate is 30 FPS. The videos have a length between 30 seconds and four minutes.

The ground-truth is based on people behavior, for instance, a person staying for a long time at the door or going around suspiciously. For training, we selected ten days (1,100 normal trajectories), and rest of days for testing (2,946 normal and abnormal trajectories). Videos contain at least one person and might have up to 10 persons in the same scene.

6.3.1 General Settings

The setup for the tracking step depends on the video scene. Each dataset has a different variable setup, basically due to the people that appear on the scene being near or far from the camera. We fixed the number of points per trajectory to $n = 64$. For the recurrent autoencoder, the GRU input is the trajectory divided in segments, each segment composed of eight points with an overlap of four points between them for each segment. This autoencoder was trained after 200 epochs, we use sigmoid function for the activation and hard sigmoid the recurrent activation, we employed mean square error as loss function and AdaDelta algorithm for optimizer. For the convolutional autoencoder, the training phase was limited by 300 epochs, using mean square error as loss function and the Adam optimizer algorithm. The average loss value during training were 0.0001 and 0.01 for recurrent and convolutional networks respectively. These protocols were the same for all datasets. The Δ value for KNN distance algorithm is used to build the Receiver Operating Characteristic (ROC) curves, this value is set between $[0, 10]$ with intervals of 0.01. There are two evaluation criteria: Area Under Curve (AUC) and Equal Error Rate (EER) which is the ratio of misclassified frames at which $FPR = 1 - TPR$. We fixed number of points for trajectory is $n = 64$. For the recurrent autoencoder, the input for GRU is the trajectory divided in segments, each segment composed of eight points with an overlapping between them in four points for each segment. The Δ value for k-NN distance algorithm is used to built the curves, this value is set between $[0, 30]$. All the comparative values from other models were extracted directly from source papers using a tool that extracts points from plots [Rohatgi, 2018].

Our model employs a pre-processing step which consists in to find the reference point. This step is carried out by the model proposed by Cao et al. [2017], using the Keras implementation [Bouguet, 2018] (all values setting as default as the authors indicate in the website). This framework receives as input an image and returns the points corresponding to the body joints of the people in the scene. We modified the code so that it receives a video as input, and the output is only the point that represents the joint between head and the body. It is important to highlight that in our experiments

we also try to find the reference point using face or head detection models, however, in many cases specially when people have their backs to the camera, the detection is lost, in contrast, pose estimation framework present a better people detection. Additionally, pose estimation provides the reference point at the same time, instead of head or face detections which already have the problem of changing bounding box.

6.3.2 Anomaly Recognition

In figure 6.7 that presents the results (also in following figures that present the results), our model is called Temporal Autoencoder of Trajectories (TAoT), this label is accompanied by letter 'T' in case of recurrent autoencoder or 'M' in case of convolutional autoencoder. In figures, TAoT-T is colored in red and TAoT-M in green. To compare with ground-truth, any anomalous trajectory mark the entire frame as anomalous, this evaluation methodology is known as Frame Level Analysis.

The subway dataset. This dataset is composed of two sequences in a set of videos proposed by Adam [Adam et al., 2008]. The first video sequence, known as *Entrance Gate*, has a time length of one hour and 36 minutes and the second video, called *Exit Gate* has length of 43 minutes. These sequences correspond to a ticket gate in a subway entrance and exit. The original ground-truth provided by the authors, containing the initial frame of anomalous events, focuses on two specific anomaly types: walking in wrong direction and jumping the ticket gate. For both video sequences, we utilized the validation protocol presented by Saligrama and Chen [2012].

Entrance Gate is a sequence recorded from a subway entrance gate view. The training phase considers the initial 20 minutes (first 30,000 frames) and the remaining of the clip for test (approximately one hour and 16 minutes), where the ground-truth presented two types of anomalies: walking on wrong way and jumping the ticket gate. For this sequence, we compare our results with the state-of-the-art approaches proposed by Roshtkhari and Levine [2013] (Sparce), Cheng et al. [2015] (GPR), Li et al. [2014] (Bayes), Saligrama and Chen [2012] (Agr) and Colque et al. [2017]. Figure 6.7(a) shows our experimental results and the comparison with the state-of-the-art. Our recurrent descriptor achieved a promising result compared with recent methods in the literature. Unfortunately, our convolutional descriptor missed some anomalies, specifically the ticket jumping, because the convolutional autoencoder aims to describe the morphology of the trajectory and when the people jump the ticket gate the morphology of the trajectory is similar with other normal trajectories.

The *Exit Gate* clip contains data recorded from a subway exit. In this case, the ground-truth considers only people walking in wrong way. The training set considers

only the first five minutes (first 8,000 frames) and the rest of video is used to test. We compare our results with the methods proposed by Li et al. [2014] (Bayes) and our proposed descriptor Colque et al. [2017] (HOFME). Figure 6.7(b) presents the results for this clip. In this case our recurrent descriptor outperforms the other models. However in this case convolutional descriptor reports low AUC compared with the other methods.

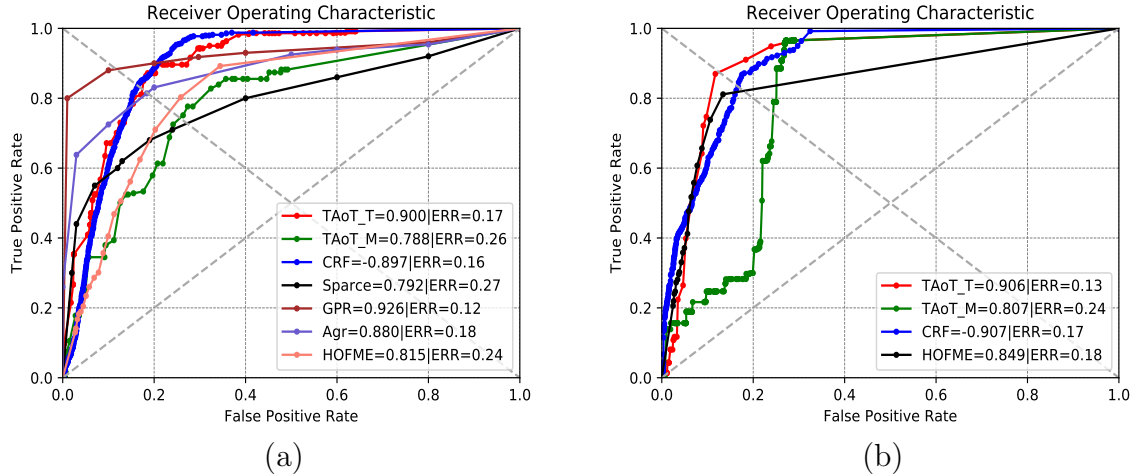


Figure 6.7: Experimental results and comparison with the state-of-the-art on the *Entrance*, *Exit*. (a) ROC results for Entrance clip; (b) results for the Exit clip.

Train sequence. The train sequence is part of a set of videos for anomaly recognition proposed by Zaharescu and Wildes [2010]. This video clip has a view from the interior of a train coach and is the only sequence in the dataset that contains people in the scenes. It has 19,218 frames which are very challenge due to drastic variation in lighting conditions and camera jitter. The anomalies in this sequence comprise people coming out and moving on the train.

For this video sequence, we present two results, where “TAoT-T-mod” (green colored curve) and “TAoT-T” (red colored curve), both using recurrent descriptor. The first experiment was performed training with 800 frames and testing with the rest of the video. The second experiment “TAoT-T”, which following the original ground-truth configuration file for validation where for training is used the first 800 frames and testing the last 5000. These experiments have two goals, first we want to evaluate our recurrent descriptor, which presents better results for anomaly detection in a difficult lighting condition sequence. Figure 6.8(c) shows our results and the results achieved by Cheng et al. [2016] (Bayes). In the experiments the second experiment obtained better results because the information for trajectories were clear in contrast

with the other experiment. Also, according to the results shown in Figure 6.8(c), our model outperforms the Cheng’s method because, to build the trajectories, our model utilizes a pose estimation/person detector, which is robust to problems of illumination changes, camera movement, shadows, etc. The “TAoT-T-mod” experiment looks for introducing more knowledge to the model, training with more frames than original 800. Unfortunately, the accuracy of the model reduced due to new trajectories being confused with anomalies.

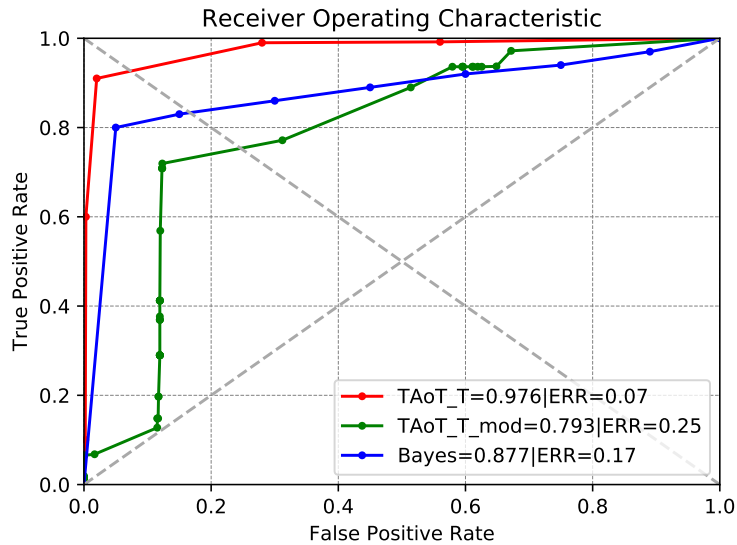


Figure 6.8: ROC curves for Train sequence.

Avenue dataset. Introduced by Lu et al. [2013], the avenue dataset contains videos from entrance avenue at the Chinese University of Hong Kong (CUHK) and is composed of 16 training videos and 21 test clips. Testing videos include both normal and anomalous events. It comprises three types of anomalies: running, wrong direction and abnormal object. Abnormal object sequences contains a person that pulls up a backpack. In this work, we did not perform test with sequences that contains this type of anomalies, because are out of study analysis. Thus, our experiments were performed without sequences 5, 10, 12, 13, 14, 16, 17 and 20. All videos for training were used to tune the network. In this experiment, we just tested the recurrent descriptor. According to Table 6.1, we achieve the best result in sequence 18 and the worst in sequence 19. In this last sequence, the missed anomaly is a person walking in a wrong direction towards the camera, but due to the projection, the generated trajectory was too small. In this case, depth information would be beneficial. We cannot compare this experiments with literature due to sequence reduction in our experiments. However,

Sequence	AUC	ERR %
1	0.69	45
2	0.80	24
3	0.44	53
4	0.94	22
6	0.84	21
7	0.88	21
8	0.82	29
9	0.80	29
11	0.74	34
15	0.50	47
18	0.95	9
19	0.22	51
21	0.61	35
Mean	.71	32.3

Table 6.1: AUC and ROC for Avenue sequences. Highlighted in bold, we present our best and worst result.

compared only the mean AUC with other studies [Hasan et al., 2016b; Kiran et al., 2018] our results are still competitive.

6.3.3 Rare Trajectory Identification

The experiments performed to identify rare trajectories intend to separate or identify trajectories that are not usual. The criterion is simple, clustering trajectories to segment common from uncommon. Rare trajectories are useful because they are not necessarily anomalies, but could be suspicious events that would trigger an alarm. For these experiments, we introduce a novel dataset called *Laboratory*.

The Laboratory dataset contains one month of recordings of the entrance of a laboratory. The video resolution is 1280×720 , recorded at a frame rate of 30 FPS. The videos have length between 30 seconds and four minutes. The ground-truth is based on people behavior, for instance, a person staying for a long time at the door or going around suspiciously. For training, we selected 10 days of recordings (1,100 normal trajectories), and the remaining for testing (2,946 normal and abnormal trajectories). Videos contain at least one person and might have up to 10 people in the same scene. We evaluate both the convolutional descriptor and recurrent descriptor. Table 6.2 shows the clustering results, which reports the number of clusters created, the cluster with the smallest number of trajectories and the one with largest number, respectively. Clusters with a small number of trajectories are trajectories that have unusual morphology, but are not necessarily anomalies. We see that TAoT-M creates

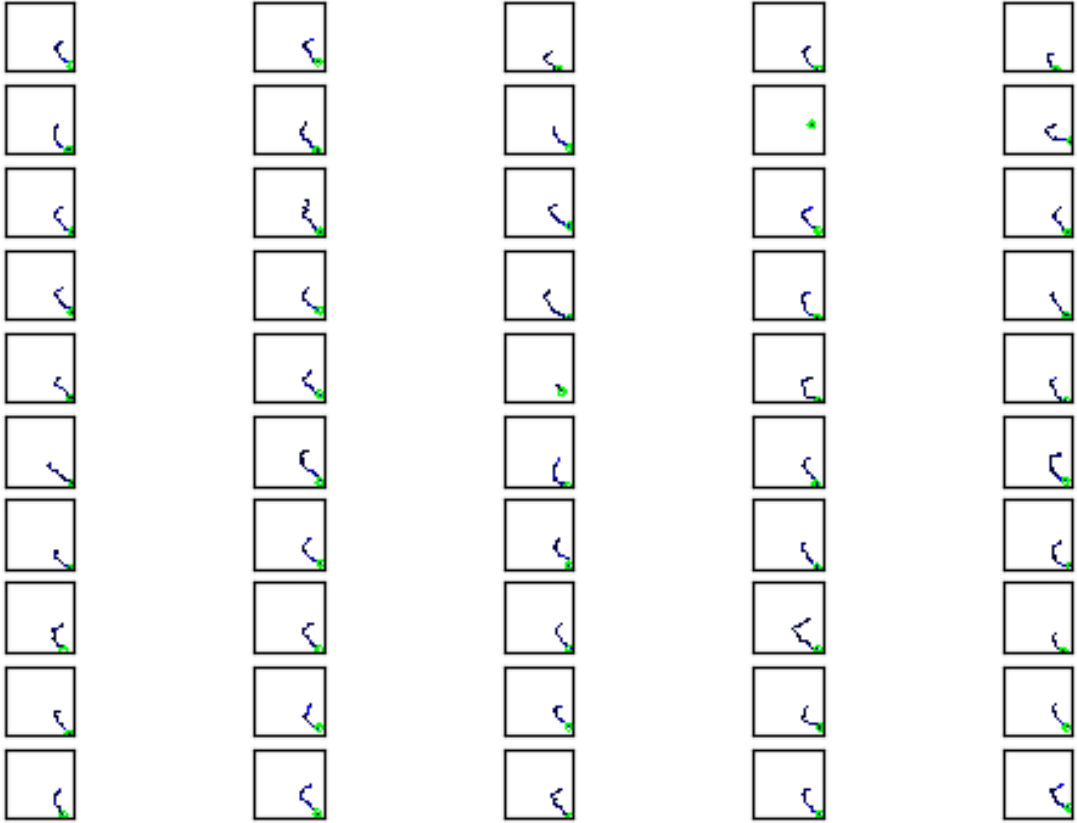


Figure 6.9: Example of normal cluster. which in this case contains 73 elements.

more clusters than TAO-T, which are in turn less compact, allowing to TAO-M to find more rare trajectories. This shows that TAO-M is better at finding more fine-grained differences between the trajectories. Despite this, TAO-T still yields better results on anomaly recognition, since it is able to group spatially similar trajectories, while anomalies have very dissimilar morphology.

Rare trajectories are trajectories that are morphologically distinct. In our case much of these events are not rare trajectories. Figure 6.9 presents a cluster with common trajectories. Figure 6.10 shows an example of rare trajectory identification, these images are thumbnails from original trajectories, the circle (green) represents the initial point of each trajectory. It is important to highlight that our descriptors preserve also the direction as well as the morphology of the trajectories.

Descriptor	N. Clus.	Min. Ele	Max. Ele
TAoT-T	58	5	212
TAoT-M	141	3	133

Table 6.2: Clustering chart for Rare Trajectory Identification.

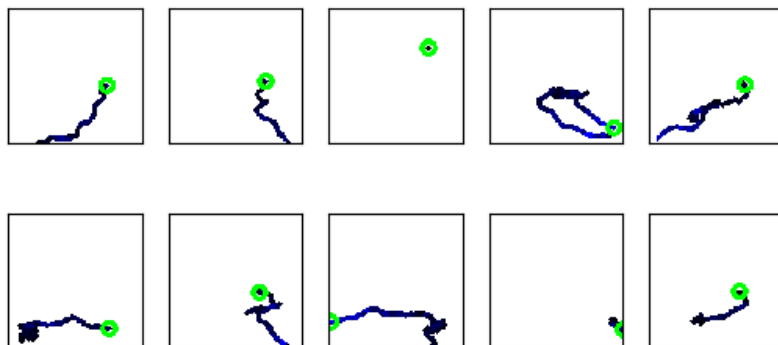


Figure 6.10: Example of anomalous cluster. which in this case contains 10 elements or trajectories.

We also performed tests over UMN Science and neering: Monitoring Human Activity [2018]. For trajectory based anomaly recognition. The UMN is a challenging dataset because of its low resolution, occlusion and very short time for training. After observing that, we just used two of eleven short sequences. For both sequences, one and four, clustering divides anomaly situations from normal into two groups.

6.3.4 Discussion

In this section we present some important remarks from the experiments performed considering all proposed approaches.

In the experiments carried out, specifically, in the approaches based on hand-crafted features and trajectories, we only use the k-NN method to detect the anomalous patterns. An alternative for this model could be the one-class SVM. However, we consider that the spatial distribution of the patterns in space is not uniform. Figure 6.11 presents a visualization of points using t-Distributed Stochastic Neighbor Embedding (t-SNE) [van der Maaten and Hinton, 2008] in a video sequence of UMN [Science and neering: Monitoring Human Activity, 2018] dataset. In this dataset, there are two situations where people are walking and running, being the last the anomalous event. Purple points correspond to normal patterns, colored with yellow, green and blue correspond to anomaly patterns, we colored these points using Afinity Propagation (AP) algorithm and the ground-truth. We can appreciated that the cloud points are not clearly segmented and anomalies are in the middle of the cloud. Furthermore, anomalies are not joint together. For this reason, we employed only k-NN for our experiments.

Another important point to take in consideration is the classification method, in our handcrafted descriptor and trajectory based approach, we employed the K-NN

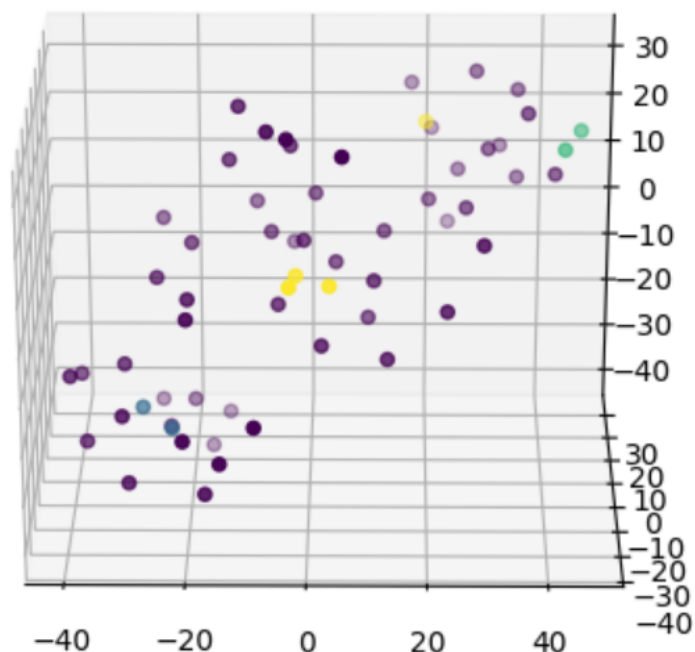


Figure 6.11: Image visualization of normal and anomalous points using t-SNE. Purple points correspond to normal patterns, colored with yellow, green and blue correspond to anomaly patterns.

method to label a pattern as normal or anomalous, it due to the unknown distribution of the points. It means, as we can see in Figure 6.11 (3D representation of space reduction), normal patterns could be mixed among the anomalous. For instance, taking as example our trajectory based approach experiments, specifically the Entrance view of Subway, when people jump the ticket gate normal trajectories are pretty similar with the trajectories of this anomalous event, the difference appears only when the person is jumping, this causes that feature points have a similar representation, differing only in some values that represent this deviation. Then K-NN model fits well in our model considering the points that are very near to the normal cloud and the outliers that are far from the point cloud.

Chapter 7

Conclusions

This dissertation has set out to explore the possibility of using environment context for activity understanding and unusual event detection in surveillance videos. In this study, we proposed three methods for anomalous event detection. In the first method, based on low level features, our goal is to detect anomalies in outdoors and when the environment may have crowds and usual detectors cannot detect many objects due to distance or video quality. In a second approach, we propose a model based on human-object interaction, with this model we pretend to use context information to define patterns, these patterns are described as set of objects that are related to a person in a determinate frame. In other words, we describe the scene using human-object relations. In the third approach, we present a model that describes the human trajectories in the scene. Similar to our second proposal, this approach employs uses higher level information to describe events on the scene.

Handcrafted descriptors for anomaly detection are very popular in literature. This type of features still make an important contribution to anomaly detection researches. In our first approach we proposed a low level descriptor called Histograms of Optical Flow Orientation, Magnitude and Entropy (HOFME). The main goal of the proposed feature is to improve anomalous event recognition tasks. It uses of orientation and magnitude from optical flow information in order to create a feature vector for a spatiotemporal region.

We evaluated the performance of our approach compared to other published results on the UCSD Anomaly Detection dataset and Subway dataset, two well known publicly available datasets for the evaluation of anomaly detection. On UCSD dataset we achieved state-of-the-art results on Peds2 scenario and our model presented comparable results on the Peds1 scenario.

On Subway dataset we achieved 100% of accuracy recognizing all anomalous situations on Exit Gate clip and 83% of accuracy on Entrance Gate clip. Although our model recognized most of the anomalous events, it also presented many false alarms.

The two main reasons are because the original ground truth focused only on the ticket gate and, during the training phase, people move in few directions. In view of that, we can state that our knowledge regarding anomalous direction is not only for the ticket gate, but it is for all the scene. To cope with these situations, we also proposed a new ground truth addressing such anomalies considering all the scene. Moreover, we introduced a new anomaly detection dataset, known as Badminton, composed by a labeled video sequence recorded from a badminton game.

The parameters of this model must be set according to the scenario. For instance, depending of the camera position and the human size spatiotemporal regions would be changed in every scene. This would be taken as a disadvantage, because the collected information only can be used for this specific situation. If camera moves the scene may vary enough to present a totally different scenario.

During the investigation, many models have been testing to improve the results of the complete pipeline, for instance, including new axis with appearance information (quantized Gabor information), or even modifying the distance when histograms are compared (cosine, mahalanobis, manhattan). Nevertheless, none of them achieves an important results that are worth to present.

An important contribution of this approach is that our descriptor is quite simple compared with other complex model in literature. However, as we presented in experiments chapter, our results can be compared with other descriptors.

As second approach we proposed a model for anomaly detection and localization based on context information. Instead of modeling using common information such texture, magnitude or orientation, we proposed a model based on human-object interactions. An important contribution of this study is the different perspective about the information collecting and the anomaly representation. Our model is capable to detect anomalies and to determine which individual in a certain frame makes a suspicious action.

Our model can use information from many scenes (same context) and use this information to detect anomalous patterns in other scene belonging to the same context. Also, our approach tries to introduce semantic information leaning relation patterns, thus, it does not need that camera is in a determinate position, as long as the context does not vary much. Therefore, our approach can detect and also determine the subject that caused the anomalous event.

In a third approach we proposed a method based on human trajectories. It consist in a spatial and temporal trajectory descriptor for anomalous event detection based on deep neural networks, aiming at describing trajectories by their morphology. This study presents a novel approach for anomaly recognition extracted from higher

level information.

In our approach we also proposed a heuristic for multi-object tracking for data association based on Kalman filter. The goal of our heuristic is to propose an easy alternative to a complex problem, based only on the results of the anomalies and our subjective analysis. However, the main objective of the model does not focus on the algorithms but on the general idea.

In our study we presented an experimental evaluation regarding trajectories and the relation between anomalies and rarity. Our experiments show that our proposed model achieves comparable results regarding the state of the art in terms of recognition of anomalies. As expected, the clustering of the trajectories also showed that the rare events were in the clusters with less number of elements.

7.0.1 Additional discuss

Anomaly event detection is a challenging task even for humans, actually even to create the annotations or to define the anomalies in a determinate scene. Indeed, many of the dataset ground-truth have the inherit subjectivity of the annotator. In our experience in the creation of own ground-truth for the proposed datasets and also the subway dataset, the events that could be anomalous depending of the point of view of the annotator about the scene context. For instance, for Cosar et al. [2017] in the ground-truth employed in their experiments over subway entrance clip, they considered people loitering, where in our opinion the people just waiting at the door of the entrance is not much suspicious as the child running which appears minutes later of this event in the video sequence. Following the same argument, we can mention the anomaly event based on the wheelchair in UCSD dataset, because that person goes in same velocity as the other people, here, the annotator clearly focuses on the appearance of the wheelchair, however it is neither suspicious nor anomalous. Then, in many cases the algorithms fit well in the certain datasets and are not fine for others. In our experiments we chose different datasets in which our approaches can fit or solve the anomaly detection problem. In the case of our proposed datasets, the ground-truth was annotated by different people in our laboratory.

The camera view is important for our approaches. The handcrafted and trajectory based approaches must use videos with fixed camera view. It means the camera cannot present movement. It because they collect information about the scene in that specific view. In contrast, our approach based on human-object interactions may utilize the information collected in one camera view and use this information in another scene that has the same context of the collected information. In addition, this model allows small

movements of the camera since the scene context does not change. For instance, in the classroom example, if the camera rotates, however the view is still the classroom, then our approach works fine. In other case, if the camera rotates and the view change to the window or to another view where the idea of classroom changes, then our approach cannot be employed.

Following with the scene context discussion, how similar can a context be? The context is also a subjective concept, for instance in the classroom example, there are many types of classrooms, then the context must be limited by the semantic where the model could be used. Thus, for example, the classroom in the computer science building composed by chairs, tables, square, students, professors, among others typically objects, however, could be a special classroom that only contains couch and other objects regarding presentations. Although this example is obvious, it is important to highlight because, in some cases the camera view could contain different objects.

In the literature, there is another type of analysis for anomalous event validation, this methodology is called pixel level. In this methodology, the certain localization at pixel level is provided by the ground-truth, thus the researchers could test their model trying to find the exact localization of the anomaly in a determinate frame. However, this methodology is not common in the literature, because it depends on the correct delimitation of the ground-truth, and the definition of anomaly in that context. This could be very subjective. In our trajectory based approach, the goal is to identify the person whose performs the anomalous events, it is similar to pixel level, however, not the same. Pixel level refers to position on the image frame, instead of that our model refers to the person which has a position in the image frame. This difference is important because in pixel level only the event is highlighted, in our approach the tracklet shows the complete event from the beginning to the end.

The complexity of our approaches mainly depends on the tools that have been employed for achieving the goal anomalous event detection. Hence, in handcrafted approach the complexity is in worst case when all the frame presents movement, then the final complexity for feature extraction is $O(f(nm))$, where n is the number of pixels in the spatiotemporal region, m is the $3D$ region for entropy extraction and f is the optical flow computing. The complexity for testing phase is given by the number of cuboids in the entire test video, for each cuboid the complexity is the distance for all trained patterns with each test pattern. In the human-object interaction approach, the complexity is computed by the structure building, this depends on the number of people in the scene. To determine the anomalous event the complexity of each first strategy is the binary search into the dictionary building with the structure label. In the second strategy, the complexity is computed regarding the two-gram complexity, in this case

we built a sparse probability matrix in form of two hash tables, where is stored the probability each tuple of labels found in the training phase. The search in this matrix is $O(\log(d))$, where d is the number of words in the dictionary label structure. In the third approach, the complexity is the sum of various models, pose estimation, tracking building, and neural network feature extraction in training phase. Pose estimation has high computational cost, which is not detailed in the original paper. Tracking building depends on the number of people in the scene, for instance in some frame with three people the complexity is $O(m(N^2))$, where $m(N^2)$ is the Munkres algorithm and N the number of people in this specific frame. For feature extraction, the complexity depends on the neural framework. In testing phase, k-NN complexity is given by the number of learned patterns same as affinity propagation clustering.

All the experiments in our research were done offline. Undoubtedly, one of the characteristics of the recognition of anomalies in surveillance videos is prevention, in this way, some studies focus on real time models [Bera et al., 2016; Sultani et al., 2018]. However, in our opinion, our first approach could be easily adapted to an approach of this type models. In the case of our future proposals, we do not doubt that with the advance of the technology, the speed in the pose recognition, which is the method that takes the longest, allows to realize the detection of anomalous events in real time.

7.1 General Conclusions

In this section we present the general study conclusions:

- The definition anomaly can have some degree of ambiguity within a domain of application.
- Anomalies are infinite, in this study proposed models with the objective of recognize at most as possible type of anomalies that happening in surveillance videos. In our approaches we focus on movement patterns, where other characteristics like appearance or more semantic information are not employed.
- In a complete framework all the approaches complement each other.
- Approach based on low-level features suits better for crowded scenes, however, the influence of noisy data, the choice and representation of low-level features, significantly influences the discriminative power of the detection.
- Human-object based approach tries to cover the drawbacks presented in first model, using patterns with more semantic information.

- Trajectory based approach attempts to offer an alternative to anomalous event detection where the target is recognized at the same time as the anomalous event.
- Two datasets and a video sequence have been incorporated to the literature. The goal of these datasets is to introduce an alternative for anomaly detection experiments.
- Some of our models are available in the page of the laboratory, and the rest will be published soon.

7.2 Future Directions

In this section, directions that can be extended from the contributions of this study, are presented. This study proposes different modeling techniques and feature representation techniques to the problem of anomaly detection.

Deep Neural Networks are a trending research topic in both machine learning and computer vision in recent times. They have been outperforming most of the state of the art performances in the fields of object classification and recognition. In this work we propose two descriptors based on autoencoders, however, new trends based on generative models [Ravanbakhsh et al., 2017; Lawson et al., 2017] present a very interesting field to continue with our proposed feature descriptors.

In our second approach, human-object interactions, the main idea is to describe the scene. Hence, a path that would be taken is to describe also the activities that people perform. These activities initially could be a simple action like running, walking, sitting, waiting. The aim is to introduce more information into the structures.

Multi-tracking humans in crowds is a hard problem to solve [Čehovin et al., 2016]. However, this is a point that must be improved for the continuation of our study. A good characteristic that would help to improve the tracking is the appearance. Since the reference point is located, we may assume that the region around that point presents a fixed appearance. This information could be employed into the score function.

Another possible approach is to start by densely sampling spatiotemporal features and subsequently add structure to the sequence representation. This information also can be modeled or described by AEs.

One type of information that was not explore in our study is the depth. We believe that depth of the actors (people and objects) contains important information. In this perspective, we found two alternatives, one based on auto camera calibration [Vasconcelos et al., 2018] for scene homography and the second using depth map [Eigen

et al., 2014] extracted using a DNN. We consider that both options would provide promising results.

Bibliography

- Adam, A., Rivlin, E., Shimshoni, I., and Reinitz, D. (2008). Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 555–560.
- Aggarwal, C. C. (2013). *Outlier Analysis*. Springer Publishing Company, Incorporated. ISBN 1461463955, 9781461463955.
- Agrawal, S. and Agrawal, J. (2015). Survey on anomaly detection using data mining techniques. *Procedia Computer Science*, pages 708 – 713.
- Ahmed, M., Mahmood, A. N., and Hu, J. (2016a). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, pages 19 – 31.
- Ahmed, M., Mahmood, A. N., and Islam, M. R. (2016b). A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems*, pages 278 – 288.
- Akoglu, L., Tong, H., and Koutra, D. (2015). Graph based anomaly detection and description: a survey. *Data Mining and Knowledge Discovery*, pages 626–688.
- Andersson, M., Gustafsson, F., St-Laurent, L., and Prevost, D. (2013). Recognition of Anomalous Motion Patterns in Urban Surveillance. *Selected Topics in Signal Processing, IEEE Journal of*, pages 102–110.
- Andrade, E., Blunsden, S., and Fisher, R. (2006). Modelling crowd scenes for event detection. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 175–178.
- Antić, B. and Ommer, B. (2011). Video parsing for abnormality detection. *Proceedings of the IEEE International Conference on Computer Vision*, pages 2415–2422.
- Antić, B. and Ommer, B. (2011). Video parsing for abnormality detection. In *ICCV*.

- Bera, A., Kim, S., and Manocha, D. (2016). Realtime anomaly detection using trajectory-level crowd behavior learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.
- Betke, M. and Wu, Z. (2016). *Data Association for Multi-Object Visual Tracking*, volume 6. Synthesis Lectures on Computer Vision.
- Boiman, O. and Irani, M. (2005). Detecting Irregularity in Images and in Video. *Proceedings of ICCV*, pages 1--7.
- Bouguet, J. Y. (2018). Keras real time multi person pose estimation. <https://github.com/michalfabern>.
- Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, page 1.
- Byeon, W., Breuel, T. M., Raue, F., and Liwicki, M. (2015). Scene labeling with lstm recurrent neural networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3547--3555.
- Cao, L., Tian, Y., Liu, Z., Yao, B., Zhang, Z., and Huang, T. S. (2010). Action detection using multiple spatial-temporal interest point features. pages 340--345.
- Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Caudill, M. (1987). Neural networks primer, part i. *AI Expert*, pages 46--52.
- Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, pages 1--58.
- Chaudhry, R., Ravichandran, A., Hager, G., and Vidal, R. (2009). Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*.
- Chebiyyam, M., Reddy, R. D., Dogra, D. P., Bhaskar, H., and Mihaylova, L. (2017). Motion anomaly detection and trajectory analysis in visual surveillance. *Multimedia Tools and Applications*, pages 1--26.

- Cheng, K.-W., Chen, Y.-T., and Fang, W.-H. (2015). Video anomaly detection and localization using hierarchical feature representation and gaussian process regression. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Cheng, K. W., Chen, Y. T., and Fang, W. H. (2016). An efficient subsequence search for video anomaly detection and localization. *Multimedia Tools and Applications*, pages 15101--15122.
- Choi, W. and Savarese, S. (2012). A unified framework for multi-target tracking and collective activity recognition. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part IV*.
- Chollet, F. et al. (2015). Keras. <https://keras.io>.
- Chong, Y. S. and Tay, Y. H. (2017). Abnormal event detection in videos using spatiotemporal autoencoder. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 189--196.
- Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, pages 1--9.
- Colque, R. M., Caetano, C., de Melo, V. H. C., Chávez, G. C., and Schwartz, W. R. (2018). Novel anomalous event detection based on human-object interactions. In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018) - Volume 5: VISAPP, Funchal, Madeira, Portugal, January 27-29, 2018.*, pages 293--300.
- Colque, R. V. H. M., Caetano, C., de Andrade, M. T. L., and Schwartz, W. R. (2017). Histograms of optical flow orientation and magnitude and entropy to detect anomalous events in videos. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 673--682.
- Colque, R. V. H. M., Caetano, C., and Schwartz, W. R. (2015). Histograms of optical flow orientation and magnitude to detect anomalous events in videos. In *28th SIBGRAPI Conference on Graphics, Patterns and Images, SIBGRAPI 2015, Salvador, Bahia, Brazil, August 26-29, 2015*.
- Cosar, S., Donatiello, G., Bogorny, V., Garate, C., Alvares, L., and Bremond, F. (2017). Toward Abnormal Trajectory and Event Detection in Video Surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 683--695.

- Crispim, C. F., Koperski, M., Cosar, S., and Bremond, F. (2016). Semi-supervised understanding of complex activities from temporal concepts. In *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 80–87.
- Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, pages 886–893.
- Dautov, R., Distefano, S., Bruneo, D., Longo, F., Merlino, G., Puliafito, A., and Buyya, R. (2018). Metropolitan intelligent surveillance systems for urban areas by harnessing iot and edge computing paradigms. *Software: Practice and Experience*, pages 1475–1492.
- De Almeida, I. R., Cassol, V. J., Badler, N. I., Musse, S. R., and Jung, C. R. (2017). Detection of Global and Local Motion Changes in Human Crowds. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 603–612.
- Dee, H. M. and Caplier, A. (2010). Crowd behaviour analysis using histograms of motion direction. In *2010 IEEE International Conference on Image Processing*, pages 1545–1548.
- Dehghan, A., Assari, S. M., and Shah, M. (2015). Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4091–4099.
- Del Giorno, A., Bagnell, J. A., and Hebert, M. (2016). *A Discriminative Framework for Anomaly Detection in Large Videos*, pages 334–349. Springer International Publishing.
- Du, D., Qi, H., Huang, Q., Zeng, W., and Zhang, C. (2013). Abnormal event detection in crowded scenes based on Structural Multi-scale Motion Interrelated Patterns. *Proceedings - IEEE International Conference on Multimedia and Expo*, pages 1–8.
- Duan, S., Wang, X., and Yu, X. (2014). Crowded abnormal detection based on mixture of kernel dynamic texture. In *2014 International Conference on Audio, Language and Image Processing*, pages 931–936. IEEE.
- Eichner, M., Marin-Jimenez, M., Zisserman, A., and Ferrari, V. (2012). 2D articulated human pose estimation and retrieval in (almost) unconstrained still images. *International Journal of Computer Vision*.

- Eigen, D., Puhrsch, C., and Fergus, R. (2014). Depth map prediction from a single image using a multi-scale deep network. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2366--2374.
- Escobedo, E. and Camara, G. (2016). A new approach for dynamic gesture recognition using skeleton trajectory representation and histograms of cumulative magnitudes. In *Graphics, Patterns and Images (SIBGRAPI), 2016 29th SIBGRAPI Conference on*, pages 209--216. IEEE.
- Fang, Z., Fei, F., Fang, Y., Lee, C., Xiong, N., Shu, L., and Chen, S. (2016). Abnormal event detection in crowded scenes based on deep learning. *Multimedia Tools and Applications*, pages 14617--14639.
- Feng, Y., Yuan, Y., and Lu, X. (2016). Deep Representation for Abnormal Event Detection in Crowded Scenes. *Proceedings of the 2016 ACM on Multimedia Conference - MM '16*, pages 591--595.
- Frey, B. J. and Dueck, D. (2007). Clustering by passing messages between data points. *Science*, 315 5814:972--6.
- Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A. K., and Davis, L. S. (2016a). Learning Temporal Regularity in Video Sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1--31.
- Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A. K., and Davis, L. S. (2016b). Learning Temporal Regularity in Video Sequences. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1--31.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*, pages 770--778. IEEE Computer Society.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, pages 1735--1780.
- Horn, B. K. P. and Schunck, B. G. (1981). Determining optical flow. *Artif. Intell.*, 17:185--203.
- Hu, Y., Zhang, Y., and Davis, L. S. (2013). Unsupervised abnormal crowd activity detection using semiparametric scan statistic. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 1:767--774.

- Javan Roshtkhari, M. and Levine, M. D. (2013). An on-line, real-time learning method for detecting anomalies in videos using spatio-temporal compositions. *Comput. Vis. Image Underst.* ISSN 1077-3142.
- Jiang, F., Wu, Y., and Katsaggelos, A. (2009). Detecting contextual anomalies of crowd motion in surveillance video. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*.
- Jiang, F., Yuan, J., Tsafaris, S. A., and Katsaggelos, A. K. (2011). Anomalous video event detection using spatiotemporal context. *Computer Vision and Image Understanding*, pages 323--333.
- Johnson, J., Karpathy, A., and Fei-Fei, L. (2016). Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.
- Jurafsky, D. and Martin, J. H. (2016). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 4th edition.
- Keval, H. (2006). Cctv control room collaboration and communication: Does it work? In *Human Centred Technology Workshop*.
- Kim, J. and Grauman, K. (2009). Observe locally, infer globally: A space-time mrf for detecting abnormal activities with incremental updates. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2921–2928.
- Kiran, B. R., Thomas, D. M., and Parakkal, R. (2018). An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. *J. Imaging*, 4:36.
- Kläser, A., Marszałek, M., and Schmid, C. (2008). A spatio-temporal descriptor based on 3d-gradients. In *British Machine Vision Conference*.
- Kratz, L. and Nishino, K. (2009). Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1446--1453.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS'12*, pages 1097-1105.

- Lab, S. V. C. (2014). UCSD anomaly data set. <http://www.svcl.ucsd.edu/projects/anomaly/>.
- Laptev and Lindeberg (2003). Space-time interest points. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 432--439 vol.1.
- Laptev, I., Marszalek, M., Schmid, C., and Rozenfeld, B. (2008). Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*.
- Larose, D. T. (2004). *Discovering Knowledge in Data: An Introduction to Data Mining*. Wiley-Interscience. ISBN 0471666572.
- Lavee, G., Rivlin, E., and Rudzsky, M. (2009). Understanding video events: A survey of methods for automatic interpretation of semantic occurrences in video. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, pages 489--504.
- Lawson, W., Bekele, E., and Sullivan, K. (2017). Finding Anomalies with Generative Adversarial Networks for a Patrolbot. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 484--485.
- Le Cun, Y., Jackel, L. D., Boser, B., Denker, J. S., Graf, H. P., Guyon, I., Henderson, D., Howard, R. E., and Hubbard, W. (1990). Handwritten digit recognition: Applications of neural net chips and automatic learning. In Soulié, F. F. and Héroult, J., editors, *Neurocomputing*, pages 303--318. "Springer Berlin Heidelberg".
- Leyva, R., Sanchez, V., and Li, C. T. (2017). Video Anomaly Detection With Compact Feature Sets for Online Performance. *IEEE Transactions on Image Processing*, pages 3463--3478.
- Li, C., Han, Z., Ye, Q., and Jiao, J. (2013). Visual abnormal behavior detection based on trajectory sparse reconstruction analysis. *Neurocomputing*, pages 94--100.
- Li, T., Chang, H., Wang, M., Ni, B., Hong, R., and Yan, S. (2015). Crowded Scene Analysis: A Survey. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 367--386.
- Li, W., Mahadevan, V., and Vasconcelos, N. (2014). Anomaly detection and localization in crowded scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36:18--32.

- Li Yandong, HAO Zongbo, L. H. (2016). Survey of convolutional neural network. *Journal of Computer Applications*, pages 2508–2508.
- Lin, T., Dollár, P., Girshick, R. B., He, K., Hariharan, B., and Belongie, S. J. (2017). Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 936–944.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2015). SSD: Single shot multibox detector. *arXiv preprint arXiv:1512.02325*.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *Computer Vision – ECCV 2016*, pages 21–37. Springer International Publishing.
- Liu, Y., Li, Y., and Ji, X. (2014). Abnormal Event Detection in Nature Settings. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, pages 115–126.
- Lu, C., Shi, J., and Jia, J. (2013). Abnormal event detection at 150 fps in matlab. In *2013 IEEE International Conference on Computer Vision*, pages 2720–2727.
- Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, pages 674–679.
- Mahadevan, V., Li, W., Bhalodia, V., and Vasconcelos, N. (2010). Anomaly detection in crowded scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*.
- Mehran, R., Oyama, A., and Shah, M. (2009). Abnormal crowd behavior detection using social force model. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 935–942.
- Nallaivarothayan, H., Ryan, D., Denman, S., Sridharan, S., and Fookes, C. (2012). Anomalous event detection using a semi-two dimensional hidden Markov model. In *2012 International Conference on Digital Image Computing Techniques and Applications, DICTA 2012*, pages 1–7.
- Nallaivarothayan, H., Ryan, D., Denman, S., Sridharan, S., and Fookes, C. (2013). An evaluation of different features and learning models for anomalous event detec-

- tion. In *2013 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8.
- Olah, C. (2018). <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. In *Christopher Olah blog*, page 1.
- Pathan, S. S., Al-Hamadi, A., and Michaelis, B. (2010). Crowd behavior detection by statistical modeling of motion patterns. *Proceedings of the 2010 International Conference of Soft Computing and Pattern Recognition, SoCPaR 2010*, pages 81--86.
- Popoola, O. P. and Kejun Wang (2012). Video-Based Abnormal Human Behavior Recognition; A Review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42:865--878.
- Qiao, M., Wang, T., Li, J., Li, C., Lin, Z., and Snoussi, H. (2017). Abnormal event detection based on deep autoencoder fusing optical flow. *Chinese Control Conference, CCC*, pages 11098--11103.
- Ravanbakhsh, M., Nabi, M., Sangineto, E., Marcenaro, L., Regazzoni, C., and Sebe, N. (2017). Abnormal event detection in videos using generative adversarial nets. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1577--1581.
- Redmon, J. and Farhadi, A. (2017). Yolo9000: Better, faster, stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517--6525.
- Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1137--1149.
- Ribeiro, M., Lazzaretti, A. E., and Lopes, H. S. (2017). A study of deep convolutional auto-encoders for anomaly detection in videos. *Pattern Recognition Letters*, pages 1--10.
- Rohatgi, A. (2018). Webplotdigitizer. <https://automeris.io/WebPlotDigitizer/>.
- Roshtkhari, M. J. and Levine, M. D. (2013). An on-line, real-time learning method for detecting anomalies in videos using spatio-temporal compositions. *Computer Vision and Image Understanding*, pages 1436--1452.
- Ryan, D., Denman, S., Fookes, C., and Sridharan, S. (2011). Textures of optical flow for real-time anomaly detection in crowds. In *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*.

- Sabokrou, M., Fayyaz, M., Fathy, M., Moayed, Z., and Klette, R. (2018). Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes. *Computer Vision and Image Understanding*.
- Saini, R., Pratim Roy, P., and Prosad Dogra, D. (2018). A segmental HMM based trajectory classification using genetic algorithm. *Expert Systems with Applications*, pages 169--181.
- Saligrama, V. and Chen, Z. (2012). Video anomaly detection based on local statistical aggregates. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2112--2119.
- Science, D. C. and neering: Monitoring Human Activity, E. (2018). Artificial intelligence, robotics and vision laboratory university of minnesota. http://mha.cs.umn.edu/proj_events.shtml#crowd.
- Shah, K., Reddy, P., and Vairamuthu, S. (2015). Improvement in hungarian algorithm for assignment problem. In Suresh, L. P., Dash, S. S., and Panigrahi, B. K., editors, *Artificial Intelligence and Evolutionary Algorithms in Engineering Systems*, pages 1--8. Springer India.
- Shao, J., Loy, C. C., Kang, K., and Wang, X. (2016). Slicing Convolutional Neural Network for Crowd Video Understanding. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5620--5628.
- Shao, J., Loy, C. C., and Wang, X. (2014). Scene-independent group profiling in crowd. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*.
- Shi, Y., Liu, Y., Zhang, Q., Yi, Y., and Li, W. (2016). Saliency-based abnormal event detection in crowded scenes. *Journal of Electronic Imaging*, 25:061608.
- Simonyan, K. and Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *Science (New York, N.Y.)*, 313:504--7.
- Sodemann, A. A., Ross, M. P., and Borghetti, B. J. (2012). A review of anomaly detection in automated surveillance. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, pages 1257--1272.
- Sultani, W., Chen, C., and Shah, M. (2018). Real-world anomaly detection in surveillance videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1--8.

- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9.
- Tripathi, G., Singh, K., and Vishwakarma, D. K. (2018). Convolutional neural networks for crowd behaviour analysis: a survey. *The Visual Computer*, pages 1–24.
- Vallejo, D., Villanueva, F. J., Albusac, J. A., Glez-Morcillo, C., and Castro-Sanchez, J. J. (2014). Intelligent surveillance for understanding events in urban traffic environments. *International Journal of Distributed Sensor Networks*, 10:723819.
- van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, pages 2579–2605.
- Vasconcelos, F., Barreto, J. P., and Boyer, E. (2018). Automatic camera calibration using multiple sets of pairwise correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 791–803.
- Čehovin, L., Leonardis, A., and Kristan, M. (2016). Visual tracking using anchor templates. In *WACV 2016: IEEE Winter Conference on Applications of Computer Vision*, pages 1–8.
- Vishwakarma, S. and Agrawal, A. (2013). A survey on activity recognition and behavior understanding in video surveillance. *The Visual Computer*, 29:983–1009.
- Wang, H., Klaser, A., Schmid, C., and Liu, C.-L. (2013). Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*.
- Wang, J. and Xu, Z. (2016). Spatio-temporal texture modelling for real-time crowd anomaly detection. *Computer Vision and Image Understanding*, pages 177–187.
- Wang, X., Ma, K. T., Ng, G.-W., and Grimson, W. (2008). Trajectory analysis and semantic region modeling using a nonparametric Bayesian model. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.
- Watada, J., Musa, Z., Jain, L. C., and Fulcher, J. (2010). Human tracking: A state-of-art survey. In Setchi, R., Jordanov, I., Howlett, R. J., and Jain, L. C., editors, *Knowledge-Based and Intelligent Information and Engineering Systems*, pages 454–463.

- Xiang, T. and Gong, S. (2005). Video behaviour profiling and abnormality detection without manual labelling. *Proceedings of the IEEE International Conference on Computer Vision*, pages 1238--1245.
- Xiang, T. and Gong, S. (2008). Video behavior profiling for anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 893--908.
- Xu, D., Ricci, E., Yan, Y., Song, J., and Sebe, N. (2015). Learning deep representations of appearance and motion for anomalous event detection. In *British Machine Vision Conference - BMVC*.
- Xu, D., Wu, X., Song, D., Li, N., and Chen, Y.-L. (2013). Hierarchical activity discovery within spatio-temporal context for video anomaly detection. In *International Conference on Image Processing (ICIP)*, pages 3597--3601.
- Ye, R. and Li, X. (2017). Collective Representation for Abnormal Event Detection. *Journal of Computer Science and Technology*, 32:470--479.
- Yuan, Y., Mou, L., and Lu, X. (2015). Scene recognition by manifold regularized deep learning architecture. *IEEE Transactions on Neural Networks and Learning Systems*, pages 2222--2233.
- yves Bouguet, J. (2000). Pyramidal implementation of the lucas kanade feature tracker. *Intel Corporation, Microprocessor Research Labs*.
- Zaharescu, A. and Wildes, R. (2010). Anomalous behaviour detection using spatiotemporal oriented energies, subset inclusion histogram comparison and event-driven processing. In *Proceedings of the 11th European Conference on Computer Vision: Part I*, pages 563--576.
- Zhang, D., Gatica-Perez, D., Bengio, S., and McCowan, L. (2005). Semi-supervised adapted HMMs for unusual event detection. *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, pages 611--618.
- Zhang, D. and Lu, G. (2004). Review of shape representation and description techniques. *Pattern Recognition*, pages 1--19.
- Zhang, Y., Lu, H., Zhang, L., Ruan, X., and Sakai, S. (2015). Video anomaly detection based on locality sensitive hashing filters. *Pattern Recognition*, pages 302--311.

- Zhao, B., Fei-Fei, L., and Xing, E. P. (2011). Online detection of unusual events in videos via dynamic sparse coding. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3313–3320.
- Zhou, B., Wang, X., and Tang, X. (2012). Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2871–2878.
- Zhou, S., Shen, W., Zeng, D., Fang, M., Wei, Y., and Zhang, Z. (2016). Spatial-temporal convolutional neural networks for anomaly detection and localization in crowded scenes. *Signal Processing: Image Communication*, pages 358–368.
- Zhou, S., Shen, W., Zeng, D., and Zhang, Z. (2015). Unusual event detection in crowded scenes by trajectory analysis. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 1300–1304.
- Zhu, H., Liu, D., Zhang, S., Zhu, Y., Teng, L., and Teng, S. (2016). Solving the Many to Many assignment problem by improving the Kuhn-Munkres algorithm with backtracking. *Theoretical Computer Science*, pages 30–41.