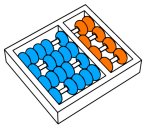
Universidade Estadual de Campinas
Instituto de Computação



Erick Luis Moraes de Sousa

CENTRIST3D: A Spatio-Temporal Descriptor for Abnormality Detection in Crowd Videos

CENTRIST3D: Um Descritor Espaço-Temporal para Detecção de Anomalias em Vídeos de Multidões

CAMPINAS
2017

Erick Luis Moraes de Sousa

CENTRIST3D: A Spatio-Temporal Descriptor for Abnormality Detection in Crowd Videos

CENTRIST3D: Um Descritor Espaço-Temporal para Detecção de Anomalias em Vídeos de Multidões

Thesis presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Computer Science.

Dissertação apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Supervisor/Orientador: Prof. Dr. Hélio Pedrini

Este exemplar corresponde à versão final da Dissertação defendida por Erick Luis Moraes de Sousa e orientada pelo Prof. Dr. Hélio Pedrini.

CAMPINAS
2017

Agência(s) de fomento e nº(s) de processo(s): CAPES, 1406874; CNPq, 159166/2015-2

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

So85c

Sousa, Erick Luis Moraes de, 1990-
CENTRIST3D : a spatio-temporal descriptor for abnormality detection in crowd videos / Erick Luis Moraes de Sousa. – Campinas, SP : [s.n.], 2017.

Orientador: Hélio Pedrini.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Anomalias. 2. Visão por computador. 3. Aprendizado de máquina. 4. Multidões. I. Pedrini, Hélio,1963-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: CENTRIST3D : um descritor espaço-temporal para detecção de anomalias em vídeos de multidões
Palavras-chave em inglês:
Anomalies
Computer vision
Machine learning
Crowds
Área de concentração: Ciência da Computação
Titulação: Mestre em Ciência da Computação
Banca examinadora:
Hélio Pedrini [Orientador]
Tiago José de Carvalho
Ricardo da Silva Torres
Data de defesa: 11-04-2017
Programa de Pós-Graduação: Ciência da Computação

Universidade Estadual de Campinas
Instituto de Computação

Erick Luis Moraes de Sousa

CENTRIST3D: A Spatio-Temporal Descriptor for Abnormality Detection in Crowd Videos

CENTRIST3D: Um Descritor Espaço-Temporal para Detecção de Anomalias em Vídeos de Multidões

Banca Examinadora:

- Prof. Dr. Hélio Pedrini
  IC/UNICAMP
- Prof. Dr. Tiago José de Carvalho
  IFSP/Campinas
- Prof. Dr. Ricardo da Silva Torres
  IC/UNICAMP

A ata da defesa com as respectivas assinaturas dos membros da banca encontra-se no processo de vida acadêmica do aluno.

Campinas, 11 de abril de 2017

Dedication

I dedicate this work to my family.

Acknowledgements

I would like to thank

- my family, girlfriend and friends, for giving me the support and strength to overcome all challenges over the years.

- the Institute of Computing and the University of Campinas, for providing an outstanding environment and for all the happy memories that I will forever cherish.

- and, most of all, Professor Helio Pedrini for his exceptional guidance and unending patience.

Resumo

O campo de estudo da detecção de anomalias em multidões possui uma vasta gama de aplicações, podendo-se destacar o monitoramento e vigilância de áreas de interesse, tais como aeroportos, bancos, parques, estádios e estações de trens, como uma das mais importantes. Em geral, sistemas de vigilância requerem profissionais qualificados assistir a longas gravações à procura de alguma anomalia, o que demanda alta concentração e dedicação. Essa abordagem tende a ser ineficiente, pois os seres humanos estão sujeitos a falhas sob condições de fadiga e repetição devido aos seus próprios limites quanto à capacidade de observação e seu desempenho está diretamente ligado a fatores físicos e psicológicos, os quais podem impactar negativamente na qualidade de reconhecimento. Multidões tendem a se comportar de maneira complexa, possivelmente mudando de orientação e velocidade rapidamente, bem como devido à oclusão parcial ou total. Consequentemente, técnicas baseadas em rastreamento de pedestres ou que dependam de segmentação de fundo geralmente apresentam maiores taxas de erros. O conceito de anomalia é subjetivo e está sujeito a diferentes interpretações, dependendo do contexto da aplicação. Neste trabalho, duas contribuições são apresentadas. Inicialmente, avaliamos a eficácia do descritor *CENsus TRansform hISTogram* (CENTRIST), originalmente utilizado para categorização de cenas, no contexto de detecção de anomalias em multidões. Em seguida, propusemos o CENTRIST3D, uma versão modificada do CENTRIST que se utiliza de informações espaço-temporais para melhorar a discriminação dos eventos anômalos. Nosso método cria histogramas de características espaço-temporais de quadros de vídeos sucessivos, os quais foram divididos hierarquicamente utilizando um algoritmo modificado da correspondência em pirâmide espacial. Os resultados foram validados em três bases de dados públicas: *University of California San Diego* (UCSD) *Anomaly Detection Dataset*, *Violent Flows Dataset* e *University of Minesota* (UMN) *Dataset*. Comparado com outros trabalhos da literatura, CENTRIST3D obteve resultados satisfatórios nas bases *Violent Flows* e UMN, mas um desempenho abaixo do esperado na base UCSD, indicando que nosso método é mais adequado para cenas com mudanças abruptas em movimento e textura. Por fim, mostramos que há evidências de que o CENTRIST3D é um descritor eficiente de ser computado, sendo facilmente paralelizável e obtendo uma taxa de quadros por segundo suficiente para ser utilizado em aplicações de tempo real.

Abstract

Crowd abnormality detection is a field of study with a wide range of applications, where surveillance of interest areas, such as airports, banks, parks, stadiums and subways, is one of the most important purposes. In general, surveillance systems require well-trained personnel to watch video footages in order to search for abnormal events. Moreover, they usually are dependent on human operators, who are susceptible to failure under stressful and repetitive conditions. This tends to be an ineffective approach since humans have their own natural limits of observation and their performance is tightly related to their physical and mental state, which might affect the quality of surveillance. Crowds tend to be complex, subject to subtle changes in motion and to partial or total occlusion. Consequently, approaches based on individual pedestrian tracking and background segmentation may suffer in quality due to the aforementioned problems. Anomaly itself is a subjective concept, since it depends on the context of the application. Two main contributions are presented in this work. We first evaluate the effectiveness of the CENsus TRansform hISTogram (CENTRIST) descriptor, initially designed for scene categorization, in crowd abnormality detection. Then, we propose the CENTRIST3D descriptor, a spatio-temporal variation of CENTRIST. Our method creates a histogram of spatio-temporal features from successive frames by extracting histograms of Volumetric Census Transform from a spatial representation using a modified Spatial Pyramid Matching algorithm. Additionally, we test both descriptors in three public data collections: UCSD Anomaly Detection Dataset, Violent Flows Dataset, and UMN Datasets. Compared to other works of the literature, CENTRIST3D achieved satisfactory accuracy rates on both Violent Flows and UMN Datasets, but poor performance on the UCSD Dataset, indicating that our method is more suitable to scenes with fast changes in motion and texture. Finally, we provide evidence that CENTRIST3D is an efficient descriptor to be computed, since it requires little computational time, is easily parallelizable and achieves suitable frame-per-second rates to be used in real-time applications.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

In this chapter, we discuss in details the problem under investigation in this work, its context and applications in security and non-security environments. Next, we present the motivation that inspired the development of our methodology, its main objectives, and our contributions. We then describe some of the encountered challenges. Finally, the text organization is presented.

1.1 Problem Description

With the growth of society and the increasing number of inhabitants, the need for improved security has given birth to a variety of fields of study. Monitoring and analyzing gatherings of people has become a matter of public security, as an abnormal behavior of a crowd might indicate imminent danger. Surveillance equipments are easy to integrate and allow the monitoring of almost all environments, although such systems are often ineffective due to some human-related factors: the natural limits of human attention capabilities, the increasing amount and length of video footages that require a large number of operators, as well as the subjective nature of what might be considered an abnormal or erroneous situation. To make the problem more complex, available datasets have very distinct characteristics.

The task of analyzing and detecting abnormal patterns in crowds heavily relies on the amount of people present in the scene, occurrence of occlusion, difficulty in removing the background, and resolution of the video sequences. Moreover, it is also dependent on the domain context, as the definition of normality and abnormality is particular to each application. Such task is very challenging and demanding, even for human operators.

Crowd abnormality detection [2, 4] can be applied in many critical areas of surveillance, such as in airports, banks, shopping centers, parks, train stations, stadiums, among others. Many applications usually require real-time monitoring of groups of people. Furthermore, the algorithms can also be employed in post-analysis tasks, where one might have to process massive volumes of data.

Besides its inherent application in security, crowd analysis is also employed in non-security related problems. Designing public spaces requires following some guidelines to optimize the organization of the available space in order to avoid overcrowding specific

places and control the flow of pedestrians [4]. Analysis of crowd can also be used to model virtual environments, as in generating artificial crowds for movies or games [4].

1.2 Motivation and Objectives

Crowd abnormality detection is a very challenging problem in the field of computer vision. No definitive general solution has been proposed yet. Due to the arising issues related to public security, advanced techniques in multidisciplinary areas (such as image processing, machine learning, data mining) can be employed to improve security of the population.

In this work, we focused our efforts to design a crowd abnormality detection framework to be employed in real-time analysis (extraction of features with at least 24 frames per second). We first evaluate the classification performance of the CENsus TRansform hISTogram (CENTRIST) descriptor [5] and then we extend it to incorporate the capability to extract meaningful spatio-temporal information. Anomaly is then detected in a frame-by-frame fashion. Several experiments are conducted on public datasets and we compare our results to other methods of the literature.

1.3 Challenges

As aforementioned, crowd abnormality detection presents some interesting challenges that must be dealt with. First, it is very difficult to define the concept of anomaly, even across the same application. In the UCSD Dataset [1], for instance, anomalous events correspond to any non-pedestrian entities, such as cyclists and skateboarders, in the walkways or any discrepant patterns of motion from a pedestrian. In a more general environment, as in a shopping center, the definition of normalcy and anomaly concepts is an even harder task: one might consider a child running as an anomalous event, even if a harmless one. Moreover, some anomalous events are specific to some jurisdictions. The act of standing in a public place with no apparent reason during a prolonged time, also called loitering, is totally or partially forbidden in some countries.

Second, crowd scenes often contain a high number of occlusions, where two or more pedestrians might partially or totally obstruct one another, making the detection and tracking of individuals less reliable. Additional factors impact negatively on the task solution: quality of the video sequences, camera position, presence of distortions and noise.

1.4 Contributions

The main contributions of this work are:

- proposition of the CENTRIST3D descriptor.
- evaluation of CENTRIST and CENTRIST3D descriptors on three crowd anomaly datasets (UCSD [1], Violent Flows [2] and UMN [3]).

- performance analysis to demonstrate that the CENTRIST3D descriptor is efficient to be computed in real-time applications and benefits associated with parallelism mechanisms.

1.5 Text Structure

This text is organized as follows. In Chapter 2, we briefly review some relevant concepts and works related to the problem under investigation. Our methodology is described in Chapter 3, along with some metrics for evaluation and other validation techniques. Chapter 4 presents the datasets used in our experiments to validate the proposed CENTRIST3D descriptor. Chapter 5 presents and discusses the results obtained with our methodology, conducts a comparison with other approaches of the literature, and describes a benchmark to measure the effectiveness of our descriptor. Finally, Chapter 6 presents our final considerations and possible improvements for future research.

Chapter 2

Literature Review

In this chapter, we first present some fundamental concepts associated to the crowd abnormality detection problem. We then review and discuss relevant approaches available in the literature.

2.1 Fundamentals

Relevant concepts related to the problem of crowd abnormality detection are briefly described in the following subsections. First, we define the concept of anomaly, presenting a coherent definition across various applications, in other words, a context independent definition. Next, we introduce the concept of crowds and their many patterns of self-organization and behavior. Afterwards, we introduce the concepts related to the chosen classifier: the Bayes theorem and the mathematical formulation of the Naïve Bayes classifier.

2.1.1 Abnormality and Abnormal Event

Anomaly is defined as a non-recurrent pattern in a given set of observations. In general, anomaly and outlier are terms used interchangeably [6]. Figure 2.1 depicts an example of two discrepant points in an artificial dataset.

There is no simple definition of anomaly in scenes of crowds since it depends on the application context. In a music concert, for instance, an abnormal situation might consist of people running, which could represent a panic situation. This same behavior might be considered normal in a marathon, where the majority of the participants would be running.

Even in simple situations, it is complex to automatically decide whether an event is considered as an abnormality or not, since crowds may contain distinct patterns of motion and different sizes. Thus, we decided to adopt the aforementioned definition, given the generic nature and requirements of our problem.

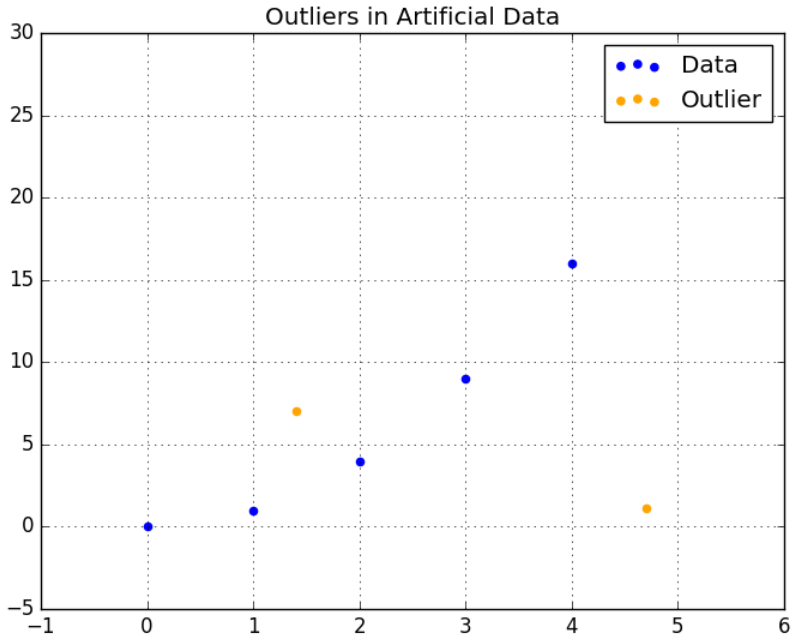


Figure 2.1: Visualization of two discrepant points (outliers) - (1.4, 7.0) and (4.8, 1.7) - in a dataset.

2.1.2 Crowds

A crowd is a clustering of people who share common characteristics or objectives. Due to population growth, the analysis of crowds has become an active research area in several knowledge domains, such as social sciences, space design, and disaster prevention [7].

The analysis of crowds can provide useful information about the occurring events of a specific place. According to Blumer [8], it is possible to categorize crowds into four different classes:

- *casual:* a crowd containing agents with little or no interaction / common objective, for instance, people walking inside a shopping center.
- *conventional:* a planned crowd, such as the participants of a protest or strike.
- *expressive:* the expressive crowd components share an emotional bond due to an occurring event. This specific crowd may arise in soccer matches or music concerts.
- *acting:* a group of people who share a common objective, such as people fleeing from a building in fire.

Similarly, the *Contagion Theory*, proposed by Le Bon [9], defines a crowd as a group of individuals who willingly accept the behavior of the collective in detriment of their own. In this manner, the crowd develops its own behavior and emotion, which can lead to an irrational state of its participants. Moreover, this unpredictable state may produce outbreaks of violence.

Following a different approach, the *Convergence Theory* [10], developed by AllPort, states that people with a similar behavior or which are willing to behave similarly tend to group into crowds. In other words, a crowd is just the direct manifestation of the tendencies of its participants. Ralph and Killian [11] proposed the *Emergent-Norm Theory*, which, analogously to the *Convergence Theory*, defines crowds as agents with similar behavior. Additionally, they state that even a small subset of participants of a crowd can generate a massive response of the whole.

Consequently, we observe that the various shapes, behaviors and interactions of crowds do not follow a clear pattern and may even change in unpredictable ways.

2.1.3 Bayes' Theorem

The Bayes' theorem [12] expresses the conditional probability of an event based on prior knowledge related to the event. Equation 2.1 expresses the mathematical definition of the Bayes' theorem, where A and B are two independent events

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} \tag{2.1}$$

$P(A)$ and $P(B)$ are the prior probabilities of occurring A and B, respectively, with $P(B) \neq 0$, $P(A|B)$ is the conditional probability of event A given B, and $P(B|A$ is the conditional probability of observing event B given A.

2.1.4 Naïve Bayes Classifier

Naïve Bayes classifier [13, 14, 15] is a simple probabilistic classifier based on applying the Bayes' theorem with strong independence assumptions between the features. Despite the term *naïve*, this classifier can achieve good results in several practical problems, such as text categorization [16] and automatic medical diagnosis [15].

As an advantage, Naïve Bayes classifier requires a number of parameters linear with respect to the number of variables, which makes it highly scalable. It demands a small number of training data to estimate the parameters necessary for the classification.

Since a Naïve Bayes is a conditional probability model, it assigns probabilities

$$p(C_k|x_1, ..., x_n) \tag{2.2}$$

for each of K possible classes C_k, where a set of n features are represented by a vector $\mathbf{x} = (x_1, ..., x_n)$.

The formulation in Equation 2.2 becomes intractable if the number of features is large. By using the Bayes' theorem, the conditional probability can be decomposed as

$$p(C_k|\mathbf{x}) = \frac{p(C_k)\, p(\mathbf{x}|C_k)}{p(\mathbf{x})} \tag{2.3}$$

Since the denominator does not depend on C and the values of the features are given, the denominator is constant. The numerator is then equivalent to the joint probability

model

$$p(C_k, x_1, ..., x_n) \tag{2.4}$$

which can be rewritten as

$$\begin{aligned} p(C_k, x_1, ..., x_n) &= p(x_1, ..., x_n, C_k) \\ &= p(x_1|x_2, ..., x_n, C_k)\, p(x_2, ..., x_n, C_k) \\ &= p(x_1, ..., x_n, C_k)\, p(x_2|x_3, ..., x_n, C_k)\, p(x_3, ..., x_n, C_k) \\ &= ... \\ &= p(x_1|x_2, ..., x_n, C_k)\, p(x_2|x_3, ..., x_n, C_k) \, ... \, p(x_n|C_k)\, p(C_k) \end{aligned} \tag{2.5}$$

It is assumed that each feature is conditionally independent of every other feature, which means that

$$p(x_i|x_{i+1}, ..., x_n, C_k) = p(x_i|C_k) \tag{2.6}$$

Therefore, the joint model can be expressed as

$$\begin{aligned} p(C_k|x_1, ..., x_n) &\approx p(C_k, x_1, ..., x_n) \\ &\approx p(C_k)\, p(x_1|C_k)\, p(x_2|C_k)\, p(x_3|C_k)... \\ &\approx p(C_k) \prod_{i=1}^{n} p(x_i|C_k) \end{aligned} \tag{2.7}$$

The Naïve Bayes classifier combines this probability model with a decision rule. A possible rule is select the hypothesis that is most probable, known as the maximum a posteriori (MAP) decision rule. In this case, the corresponding Bayes classifier is the function that assigns a class label $\hat{y} = C_k$ for some k:

$$\hat{y} = \arg \max_{k \in \{1,...,K\}} p(C_k) \prod_{i=1}^{n} p(x_i|C_k) \tag{2.8}$$

2.2 Related Work

In this work, previous approaches of the literature are classified into three main categories: tracking-based methods [17, 18, 19, 20, 21, 22, 23], pattern detection using local and global spatio-temporal features [2, 24, 25, 26, 27, 28, 29, 30] and, more recently, deep learning methods [31, 32, 33].

Tracking-based methods rely on accurate algorithms and often on background segmentation techniques in order to effectively extract trajectories. Andersson et al. [17] proposed an anomaly detection framework which dynamically detects whether a crowd is sparse or dense. A sparse crowd is defined as a one which contains enough space between individuals, so that each person can be detected and tracked. In the opposite way, a dense crowd contains lots of people - and consequently - a high number of occlusions.

Foreground-background segmentation is initially applied. Individuals and crowd candidates are then detected by analyzing human-like shaped groups and subsequently clustered into crowds using K-Means algorithm [34]. Crowd behavior is modelled using Hidden

Markov Models [35]. Basharat et al. [18] presented a novel method with scene model feedback for learning patterns of motion and sizes of objects. Tracking information (position, timestamp and object dimensions) is measured over a pre-determined temporal window, generating a set of 5-dimensional random variables. A multivariate Gaussian Mixture Model (GMM) is used to model the probability density function for each pixel in the tracked data. Finally, abnormality is detected by conducting local and global analysis of the trends in data.

In local analysis, the most recent event is compared only to the previous observation. On the other hand, global analysis captures information by analyzing a series of previous observations. Piciarelli et al. [19] presented an anomaly detection approach based on trajectory clustering using single-class Support Vector Machines (SVM) [36], exploring the fact that single-class SVMs attempt to model a set of data by enclosing the normal occurrences and leaving out possible anomalies.

Li et al. [20] proposed an approach using trajectory sparse reconstruction [37]. A set of normal trajectories is initially extracted using a track algorithm or a motion detection algorithm and manually categorized into subsets based on their appearance. Least-squares Cubic Spline Curves Approximation [38] features are extracted to create a dictionary. Sparse Reconstruction Analysis is applied on the normal dictionary and on new uncategorized samples to reconstruct the trajectories with as few as possible dictionary samples. L1-norm minimization is used and the results are classified into normal and abnormal behavior.

Chongjing et al. [21] developed a technique for motion pattern analysis using clustering of tracklets. Tracklets are obtained by tracking dense points from videos of crowds and a hierarchical clustering algorithm is used to learn motion patterns. Jodoin et al. [22] proposed a method for extracting dominant motion patterns. A motion histogram is computed for each pixel and then converted to a probability density function of the motion pattern, called the Orientation Distribution Function (ODF).

Diverging from other methods based on individual tracking, this method employs a meta-tracking algorithm for tracking the dominant flow of motion. Similarly, Mousavi et al. [23] presented a method using tracklets, known as the Histogram of Oriented Tracklets. Scale-Invariant Feature Transform (SIFT) [39] is used to detect points of interest and Lucas-Kanade Tracker (KLT) [40] is employed to keep track of them. The video sequences are then split in temporal-overlapping cuboids. The magnitude and orientation of all tracklets that intersect a given cuboid are calculated and results are binned into histograms according to the occurrence of pairs of magnitude-orientation. Abnormality is classified using Latent Dirichlet Allocation (LDA) [41] and SVM [36].

All of the aforementioned methods heavily rely on accurately tracking individuals or groups of individuals. Thus, their performance might be affected in more complex crowds and environments. The following methods adopt different strategies, avoiding the need to track or remove the background by extracting meaningful data from spatial and temporal features.

Hassner et al. [2] presented a new dataset and a new descriptor for detecting outbreaks of violence in crowded scenes, such as fights among supporters of a soccer team during a match. First, optical flow [42] is estimated between pairs of consecutive frames, resulting

in a set of flow vectors. The magnitude of each flow vector is compared to its temporal successor up to a constant factor, yielding a binary vector for each frame. Ultimately, the binary vectors are normalized and used for classifying violent and non-violent crowds using Linear SVM [36].

Mehran et al. [24] introduced a spatio-temporal Social Force [43] model based on optical flow [42] to represent interaction forces in a grid of particles. These interaction pixels are thus mapped into the images to create a Force Flow. A random amount of Force Flow spatio-temporal cuboids are extracted to model the normal behavior of the scene using Latent Dirichlet Allocation [41]. New clips are then classified as abnormal based on their dissimilarity to the model.

An algorithm based on fluid mechanics is presented by Wang et al. [25]. This approach creates a representation of crowd behavior by combining optical flow with a streakline model, the flow vector. Similarly to Hassner et al. [2], this method creates a binary feature vector for each frame by comparing the magnitude of two temporarily consecutive flow vectors. An SVM [36] is applied to the resulting vectors to classify whether a given flow is an anomaly or not.

Xu et al. [26] described a framework using the local binary patterns on three orthogonal planes (LBP-TOP) [44] to model the crowd behavior. Hierarchical Bayesian Models are then used to detect unusual events. Kratz et al [27] presented a method for modeling dense crowds using Spatio-Temporal Motion Patterns. Frames are divided into cuboids and a spatio-temporal gradient is calculated for each cuboid. The distribution of gradients is modeled using a 3D Gaussian Distribution and the underlying motion structure of the scene is captured using the symmetric Kullback-Leibler divergence [45].

Cong et al. [28] proposed the Multi-Scale Histogram of Optical Flow (MHOF), a modified Optical Flow [42] calculated in two different scales. A Sparse Coding Model is utilized to reconstruct the smallest possible dictionary of features. Saligrama et al. [29] described an anomaly detection approach based on spatio-temporal signatures. These signatures are modeled over co-occurrence statistics of abnormal events on cuboids using Markov Random Fields (MRF) [46]. Li et al. [30] presented an approach based on joint detection of spatio-temporal anomalies using a model of Mixture of Dynamic Textures (MDT) [47].

Recently, Deep Learning approaches have achieved good results in many applications, including ones related to scene analysis, such as image classification [48] and pedestrian tracking [49]. Deep Learning approaches do not require a priori knowledge of the problem or explicit handcrafting of features. Nevertheless, additional effort is necessary to setup a proper network topology.

Xu et al. [31] proposed an approach to learning motion and appearance patterns in an unsupervised framework, requiring no previous knowledge of the abnormalities themselves, using stacked denoising autoencoders (SDAE) [50]. Appearance representation and motion representation are learned from low-level features, respectively, multi-scale image patches and optical-flow patches. Early fusion is applied on both low-level features, resulting in a third model, the joint representation. A one-class SVM [36] classifier is trained for each model and their results are then combined into late fusion.

Shao et al. [32] presented a method for using crowd motion and appearance channels

as the input of a deep model. In a complementary scheme, motion-channel is divided into three independent features [51]: collectiveness (degree of union-behavior of individuals in a crowd), stability (preserving the internal order) and conflict (interaction between approaching groups).

Sabokrou et al. [33] developed a method based on an adapted Convolutional Neural Network (CNN) into a fully convolutional neural network. A two-component architecture is defined for feature representation and a cascade outlier detector.

Chapter 3

Anomaly Detection Methodology

In this chapter, we describe the main characteristics of the CENTRIST descriptor [5] and analyze its limitations. We then establish the properties suitable for a spatio-temporal descriptor and propose the CENTRIST3D. Finally, we present some possible metrics to evaluate the results obtained with our descriptor.

3.1 CENTRIST Descriptor

The CENsus TRansform hISTogram (CENTRIST) [5] is a histogram vector descriptor, initially proposed for scene classification. We first present its advantages and limitations, followed by an overview of the algorithm.

3.1.1 Properties

The CENTRIST descriptor creates a holistic representation of the scene. For instance, it is possible to infer high-level semantic features without tracking objects or removing the video background. This is specially beneficial for videos containing large amounts of people and occlusion, for instance, spectators fighting during a soccer match, where segmentation would not be practical.

CENTRIST offers good intra-class generalization capabilities: scenes with visual variations are correlated. Furthermore, structural properties are extracted, such as shapes and flat surfaces, to the detriment of texture information.

Additionally, the huge diversity of texture details on crowded environments provides little insight about the current behavior. On the other hand, it is only able to model global shape structure of small image patches. In order to improve its efficiency, a spatial pyramid-based representation [52] was proposed and constructed in this work.

CENTRIST is a very efficient descriptor, requiring only 16 operations to compute the Census Transform (CT) value for a given pixel. Due to the parallel nature of the Census Transform operation (all CT calculations are independent of each other), it is possible to speed up even further the algorithm. In the next chapter, we compare the performance between the original CENTRIST and the CENTRIST3D descriptors.

Despite of the preceding qualities and due to its innate design, CENTRIST computation does not take into account color information. Furthermore, it is not scale and

rotation invariant. Consequently, some applications might not benefit from the descriptor characteristics, such as object recognition and detection. Additional limitations include the inability to encode, by itself, temporal information as well as capturing only the global structure of small image patches. As a consequence of the latter constraint, the algorithm tries to mitigate it by creating a spatial pyramid representation.

3.1.2 Construction of the Pyramidal CENTRIST Descriptor

The proposed algorithm for constructing the CENTRIST descriptor can be divided into four steps, as shown in Figure 3.1: construction of a spatial pyramid representation, computation of the Census Transform for all image patches, calculation of the Census Transform Histogram and, lastly, concatenation of the histograms into a unified feature vector.

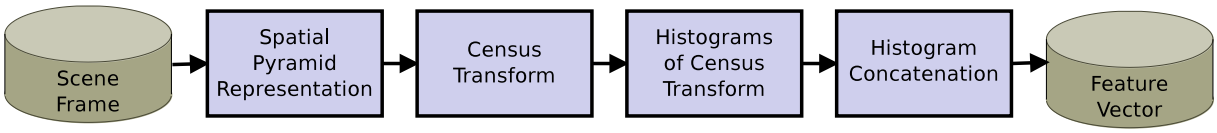


Figure 3.1: Pipeline of the CENTRIST descriptor construction.

Initially, we create a spatial pyramid representation from an image. As shown in Figure 3.2, at Level 2, we subdivide the image into 16 non-overlapping equal sized patches and, additionally, shift the partition (horizontally and vertically) in order to create other 9 patches to avoid artifacts. Likewise, the image at Level 1 is created by scaling down the original image by a factor of 2 in each dimension, resulting a total of 5 patches. Level 0 representation is also created in the same way. Next, we calculate the Census Transform for each image patch.

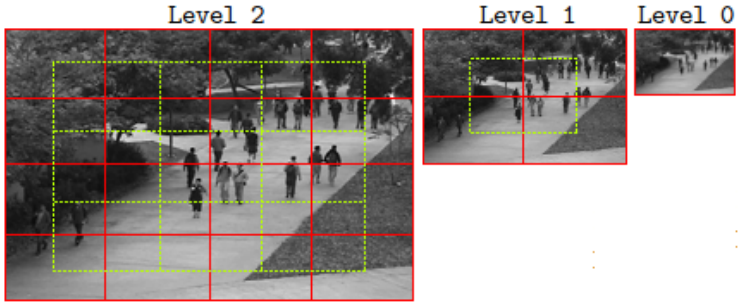


Figure 3.2: Visualization of the subdivided images and their respective patches.

Census Transform compares the value of a center pixel to its eight surrounding neighbors. If the intensity of a neighbor pixel is greater than, or equal to, the intensity of the center pixel, value 1 is assigned into its cell. Otherwise, value 0 is assigned. Afterwards, we concatenate all CT values, converting them into unsigned 8-bit integers. The process

of value assignment based on neighborhood intensities is expressed as

100	50	100
50	75	100
50	50	100

$$10111000 = (184)_{10} \tag{3.1}$$

For visualization purpose, a CT image is shown in Figure 3.3. Afterwards, we create a histogram of CT values for each and every image patch, yielding a collection of 256-dimensional histograms. Finally, we concatenate the resulting histograms into a unique feature vector.

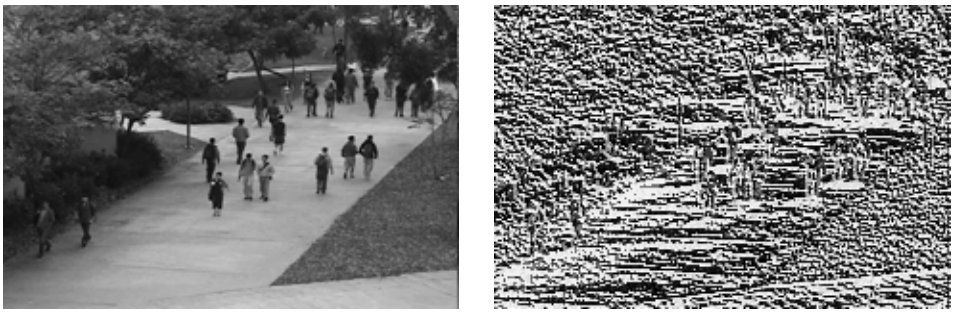


Figure 3.3: Visualization of Census Transform applied to a frame extracted from UCSD Dataset [1].

3.2 CENTRIST3D Descriptor

In this section, we discuss some desirable characteristics for our descriptor. The CENTRIST3D is then presented and analyzed.

3.2.1 Properties

CENTRIST3D algorithm is based on the CENTRIST descriptor, whereas the latter is based on a variation of the LBP descriptor [53]. By combining three orthogonal components (two temporal and one spatial) CENTRIST3D possesses similar characteristics to the LBP-TOP [54]. Both descriptors have similar calculation steps. Nevertheless, CENTRIST3D adds two additional stages: first, it creates a multiscale representation with two grids, both overlapping with each other, in order to extract spatio-temporal information across many resolutions. The second addition is a threshold C to tune the amount of discarded or retained shape information.

Our task requires handling crowded scenes and high amount of occlusion, therefore, our descriptor must be iterable to acquire information without segmentation, tracking or background removal. Anomalies may also occur in the temporal domain, for instance, in the form of a divergent motion, so it is indispensable for our algorithm to be capable of extracting meaningful temporal features.

3.2.2 Construction of the Pyramidal CENTRIST3D Descriptor

Computation steps for the construction of the CENTRIST3D descriptor [55] are similar to the creation of the original CENTRIST descriptor. Figure 3.4 shows the 5-step pipeline of our feature extraction method. For each video frame, we create a spatial pyramid representation. Frames are arranged into overlapping groups of size 3, as shown in Figure 3.5. Frame 1, 2 and 3 are stored into group 1, whereas frames 2, 3 and 4 are stored into group 2. Likewise, group 3 contains frames 3, 4 and 5. This is repeated until the very last frame of the video and we end up with $k-2$ groups. Next, we create the Spatial Pyramid Representation of the grouped frames, resulting in three sets of representations of depth k. Analogously to the CENTRIST algorithm, the Volumetric Census Transform (VCT) is applied to all groups. Furthermore, we create histograms from the resulting VCT values. Finally, we concatenate all histograms.

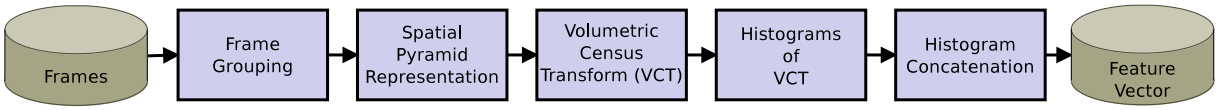


Figure 3.4: Pipeline of the CENTRIST3D descriptor construction.

In the same manner as the CENTRIST construction, a spatial pyramid representation is created for each video frame. Adjacent frames are arranged into groups of 3, as shown in Figure 3.5. By exploring the fact that motion can be detected through the change of pixel intensity values [44], we define the VCT.

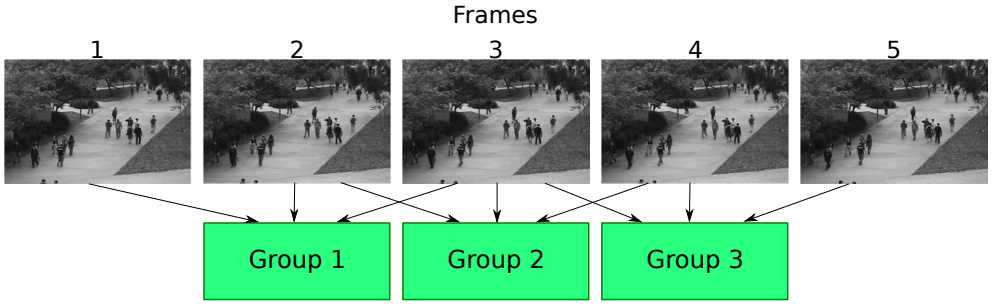
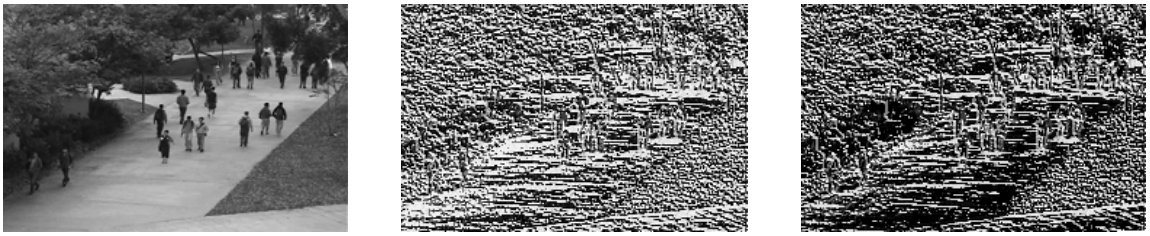


Figure 3.5: Grouping step applied to frames of the UCSD Dataset [1].

In the first step of VCT, we calculate a modified version of Census Transform. When comparing the center pixel to one of its neighbors, we take into account a new parameter C, a threshold which controls the correlation between the center pixel and the neighborhood. It controls the amount of captured geometry, as shown in Figure 3.6.

To encode motion information, the process is repeated on the XT and YT planes (Figures 3.7 and 3.8), resulting in 3 CT values for each 3×3×3 volume. Calculation of patch histograms is done independently for every plane and taking into account the previous and next frames, yielding a 768-dimensionality histogram per patch. Finally, VCT histograms are concatenated into a single feature vector.



Original frame $C = 0$ $C = 3$

$C = 7$ $C = 21$

Figure 3.6: Comparison of the modified Census Transform with thresholds set to 0, 3, 7 and 21.

$T-1$

50	100	100
100	75	50
100	50	100

T

75	100	100
80	50	10
120	75	100

$T+1$

90	100	50
15	30	45
70	10	150

Figure 3.7: Three consecutive frames at time $T - 1$, T and $T + 1$.

$CT-XY$

75	100	100
80	50	10
120	75	100

XY Census Transform

$CT-XT$

100	75	50
80	50	10
15	30	45

XT Census Transform

$CT-YT$

100	100	100
75	50	30
50	75	10

YT Census Transform

Figure 3.8: Volumetric Census Transform for all planes.

3.3 Model Learning

The depth of the pyramid-representation significantly influences the number of dimensions of the feature vector. For a pyramid with two levels, the CENTRIST3D generates a 7936-dimensionality feature vector for each plane, culminating in a 23808-dimensional feature vector and impacting negatively not only on the predictive power, but also requiring more computational resources and a larger number of features to compensate for the redundancy. In the next chapter, we present our analysis on selecting the most suitable pyramid depth.

Furthermore, an adequate prediction function needs to be learned from our dataset.

However, learning the parameters of a prediction function and executing the test on the same dataset might create a biased model. Ideally, our model must have good generalization capabilities to correctly predict new data.

To address this issue, we employ a K-fold cross validation technique. First, we split our dataset in three subsets: training set (70%), cross-validation set (20%) and test set (10%). We fit our model parameters using the training dataset. They will be in turn tuned using the cross-validation set, in an attempt to minimize the error. Finally, after rounds optimizing the parameters and evaluating against the cross-validation set, we test the resulting model against the test set. At this point, we no longer tune any parameters, as the test set is a small sample of real-world examples and we do not want to create any bias towards this data. Scores from the test set are compared to other models in order to choose the most suitable model.

The K-fold cross validation technique works as follows: the data is shuffled and divided into k groups (referred to as folds). The training set contains $k-1$ folds and the cross-validation set consists of the remaining fold. This procedure is repeated k times using different splits to account for any biases introduced during shuffle. At each round, we test the training set against a different fold. Figure 3.9 shows the fold splitting of K-Folding Cross Validation with $k = 4$.

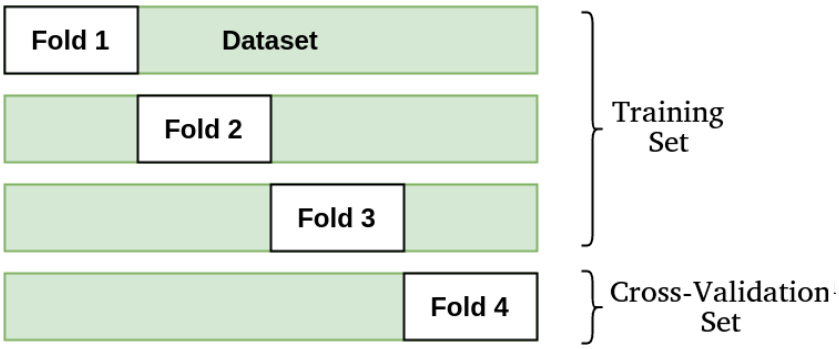


Figure 3.9: Illustration of K-fold cross validation with 4 folds.

Figure 3.10 illustrates the main components of the methodology for abnormality detection in crowd video sequences.

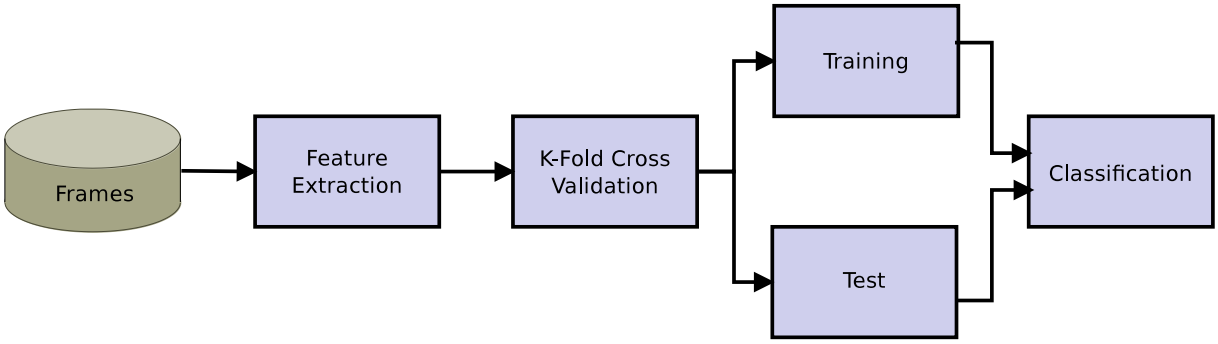


Figure 3.10: Pipeline of the proposed abnormality detection methodology in crowd videos.

3.4 Evaluation Metrics

In order to validate the effectiveness of proposed algorithms, some evaluation metrics are available: precision, recall, F1-score, accuracy, and equal error rate. Additionally, we evaluated the behavior of our method against the variation of training-set size using learning curves.

3.4.1 Precision

The precision metric evaluates the ratio between true positive and all results classified as positive (true positive and false positive), defined as:

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{3.2}$$

where TP and FP correspond to the true positive and false positive rates, respectively.

3.4.2 Recall

Recall measures the ratio between correctly classified samples and all samples belonging to a class, expressed as

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{3.3}$$

where FN corresponds to the false negative rate.

3.4.3 F1-Score

F1-score combines precision and recall in a unified measure, defined as:

$$\text{F1-Score} = 2 * \frac{\mathrm{P} * \mathrm{R}}{\mathrm{P} + \mathrm{R}} \tag{3.4}$$

3.4.4 Accuracy

The accuracy metric is defined as

$$= \text{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{3.5}$$

where TN corresponds to the true negative rate.

3.4.5 Equal Error Rate

The Equal Error Rate (EER) is defined as the intersection point of the false acceptance rate (FAR) and false rejection rate (FRR) curves. Consequently, the lower the EER, the more accurate is the prediction. Figure 3.11 illustrates the EER.

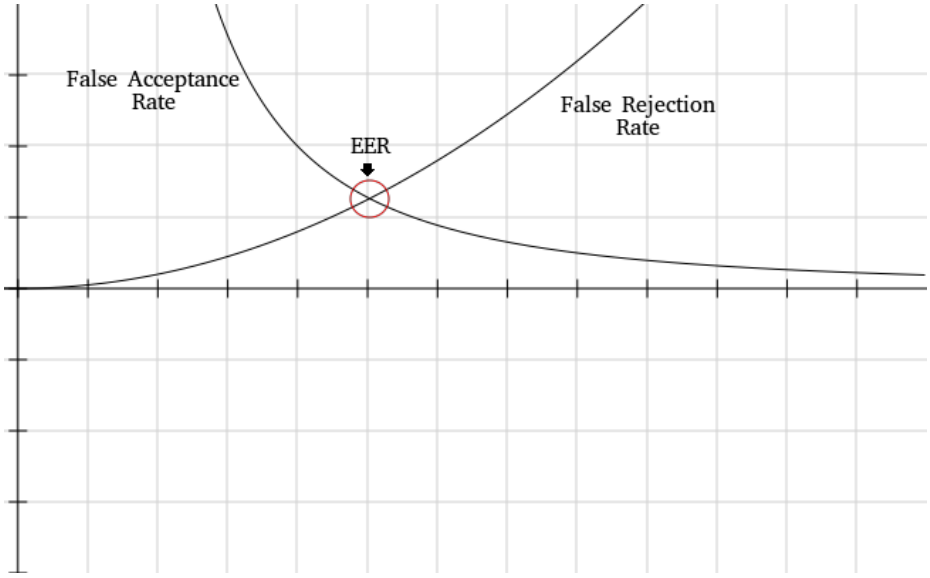


Figure 3.11: Illustration of Equal Error Rate.

3.4.6 Learning Curve

The learning curve is a technique to analyze the benefits of adding more training examples to a given model [56]. Using this technique, we can train on increasingly larger subsets, allowing us to analyze whether our model is suffering from overfitting or underfitting.

A high variance (overfitting) model will display a high score training curve, whereas the respective cross-validation curve possesses a significantly lower score. If both training and cross-validation scores converge to a low value, regardless of the number of samples, we can conclude that the model is underfitting. Thus, a model with a low variance and bias with a high score can be considered a good model. Figures 3.12, 3.13 and 3.14 illustrates, respectively, an overfitting, an underfitting and a good model using artificially generated data.

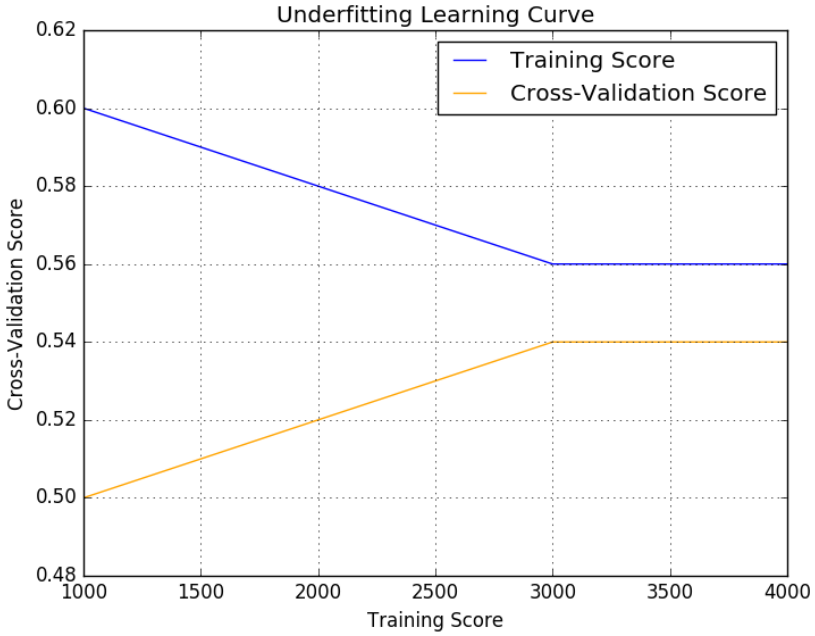


Figure 3.12: Underfitting model representation using artificial data.

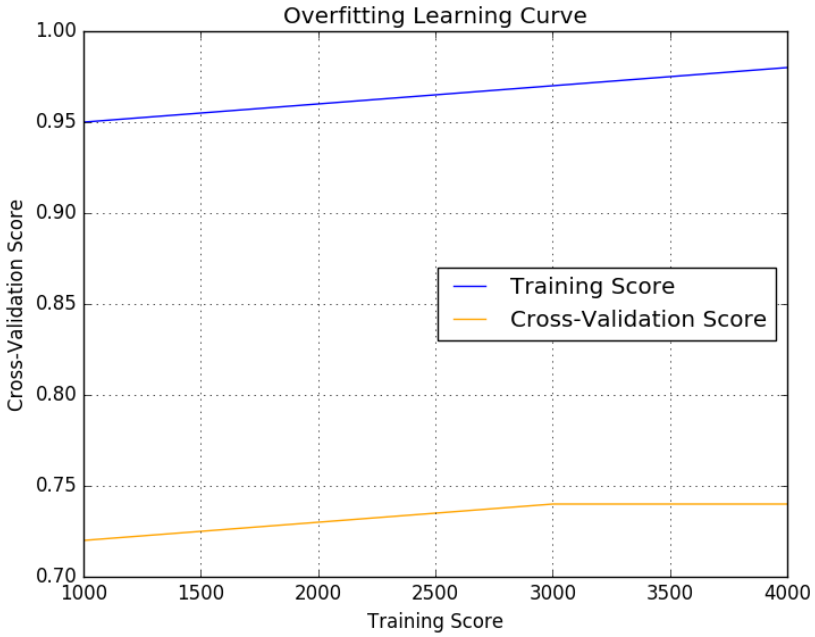


Figure 3.13: Overfitting model representation using artificial data.

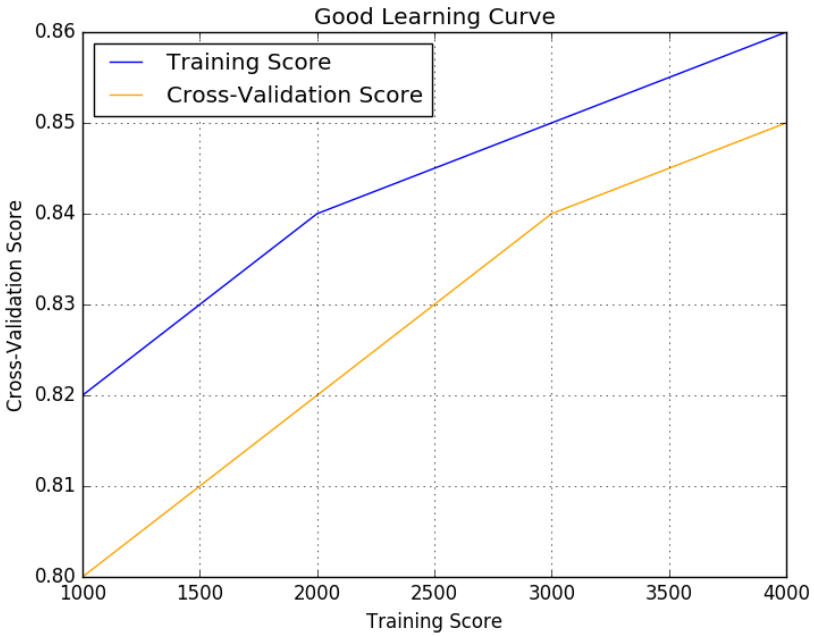


Figure 3.14: Representation of a good model using artificial data.

Chapter 4

Datasets

Three public crowd abnormality datasets are used to validate the proposed methodology: Violent Flows Dataset [2], UCSD Anomaly Detection Dataset [1] and UMN Dataset [3].

4.1 Violent Flows Dataset

This dataset is composed of 246 videos of crowds in stadiums, where half of the videos is labeled as non-violent and the other half is labeled violent. The Violent Flows Dataset is divided into 5 smaller groups, where each group is also subdivided into violent and non-violent samples.

All videos where extracted from YouTube[1] and their lengths average 3.60 seconds. The shortest video has a duration of 1.04 seconds, whereas the longest has a duration of 6.52 seconds.

Figures 4.1 and 4.2 show examples of videos labeled as non-violent and violence, respectively. We define normal frames as those that do not contain violent contents. Abnormal frames are defined as those with violent contents.

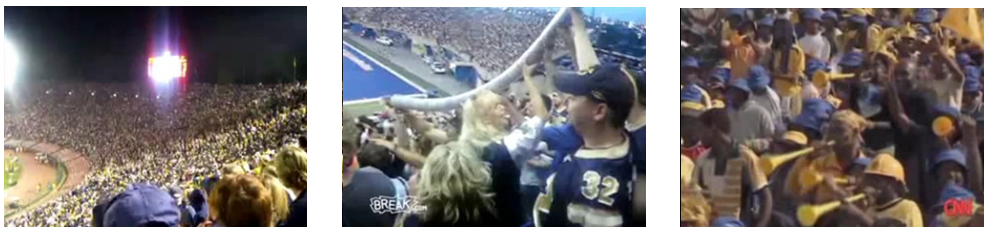


Figure 4.1: Frames from the Violent Flows Dataset [2] labeled as non-violent.

Figure 4.2: Frames from the Violent Flows Dataset [2] labeled as violent.

[1] www.youtube.com

4.2 UCSD Anomaly Detection Dataset

The UCSD Anomaly Detection Dataset [1] comprises clips of street and pedestrian monitoring, recorded by stationary cameras. Many events are defined as anomalous, such as, but not limited to, cyclists, skateboarders and pedestrians walking on the grass.

The samples are divided into two parts, namely "peds1" and "peds2". The former has 70 clips, 34 for training and 36 for testing, of groups of people walking towards and away from the camera and some perspective distortion. The latter has a total of 28 videos, where 16 are for training and 12 for testing, with videos of pedestrians walking parallel to the camera plane. Figures 4.3 and 4.4 show examples of normal and abnormal situations (marked with a red rectangle).

Figure 4.3: Normal frames from UCSD Anomaly Detection Dataset [1].

Figure 4.4: Abnormal frames from UCSD Anomaly Detection Dataset [1].

4.3 UMN Dataset

The UMN Dataset [3] is composed of 11 surveillance videos, recorded in 3 different environments, both outdoors and indoors. The footages of the first environment, an outdoor region with grass, contain a total of 1453 frames, divided into 2 videos. The second environment, an indoor footage, contains a total of 4143 frames, split into 6 videos. The third environment contains 2145 frames filmed in a square.

Each video depicts scenes of normal crowds followed by a subtle change in motion. Figure 4.5 shows examples of normal frames, one of each environment. Figure 4.6 shows examples of anomalous behavior, in each environment.

Figure 4.5: Normal frames from UMN Dataset [3].

Figure 4.6: Abnormal frames from UMN Dataset [3].

Chapter 5

Experimental Results

This chapter presents and analyzes the experimental results obtained by applying the proposed methodology to the evaluated datasets. The first section specifies the hardware and software environment of development and testing. The following two sections summarizes the results for CENTRIST and CENTRIST3D descriptor using the Naïve Bayes classifier. We compare our results to other approaches available in the literature. Finally, we show that CENTRIST3D can achieve real-time rates in all evaluated datasets.

5.1 Hardware and Software Platform

Experiments were conducted on two steps. Initially, we developed an extensible tool set in Python programming language using scikit [57] and numpy [58] libraries, where both CENTRIST and CENTRIST3D descriptors were implemented and their classification performance was evaluated. Then, we demonstrated that CENTRIST3D is also efficient to be computed by implementing both algorithms in C++ programming language using OpenMP, a Multi-Threading library [59].

We executed the performance analysis tasks on a low-end hardware: an Intel Core i5-4210U 1.70 GHz CPU and 4GB of RAM.

5.2 Results for CENTRIST

First, the influence of the pyramid depth is evaluated on the chosen datasets. Experiments were conducted using depths of 0, 1 and 2, since any depth greater than 2 would result in an extremely large feature vector (over 100,000 features), which would not only be very memory consuming, but would also require more computational power to process, making it unsuitable for our needs.

Figures 5.1, 5.2 and 5.3 illustrate the best achieved results for each dataset. In order to select the best obtained configuration, we compare the recall measure on each depth. We decided to employ the recall metric in the comparison of the results for multiple iterations due to the inherent definition of recall and its applicability in our context: we intend to maximize the number of detected anomalous events. Given that we are evaluating CENTRIST3D in security-related tasks, it is expected that most anomalies are detected,

even if misclassifying some normal events as abnormalities. By contrast, improving recall might lead to reducing the precision metric, as a consequence of increasing the number of false positive results. Therefore, we focus our efforts in maximizing recall without compromising too much precision.

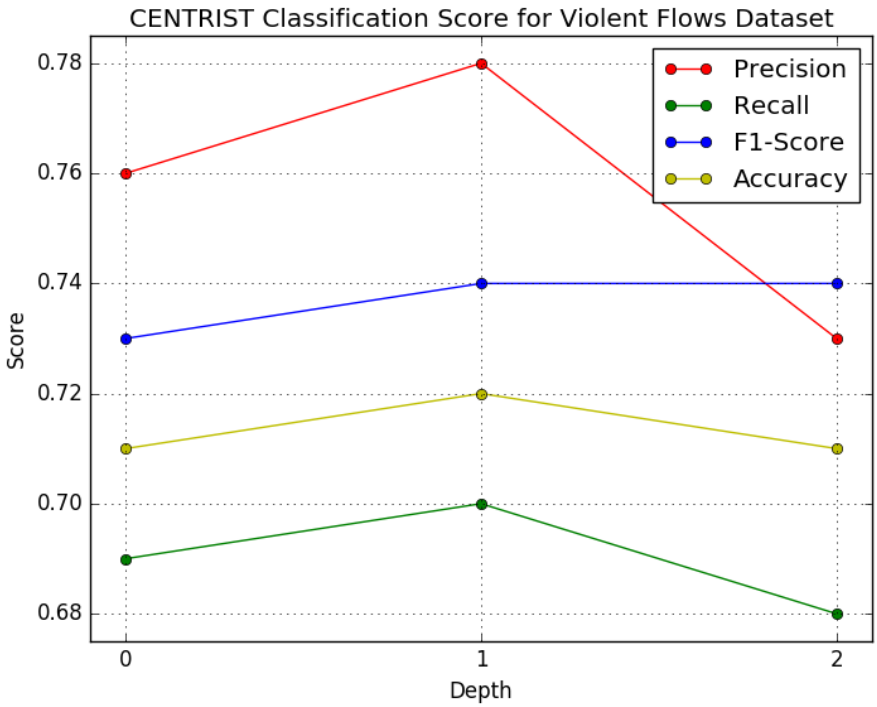


Figure 5.1: Evaluation metrics for the Violent Flows Dataset [2].

CENTRIST achieved its best score on Violent Flows and UMN Datasets by setting the pyramid depth to 1 and on UCSD Dataset by setting the depth to 0. Nevertheless, CENTRIST performed very poorly on UCSD dataset, barely reaching a recall score of 0.50 with a pyramid depth of 0, showing the inherent inability of CENTRIST to capture subtle changes in motion. Additionally, UCSD Dataset clips contains a small amount of actors, all of them with similar patterns of motion: walking upwards and downwards the path. Even the anomalies themselves follow similar patterns, slightly differing in speed or direction.

However, CENTRIST achieved slightly better results in scenes with a more chaotic behavior, such as the ones in Violent Flow Dataset, where sudden outbreaks of violence in crowds generated a more noticeable visual response. Results in UMN Dataset were also better, given the very distinctive nature of the normal events - pedestrians walking slowly in random directions - and of the abnormal events - immediate speed increase followed by crowd being dispersed from a point. Table 5.1 summarizes the best results achieved for each dataset.

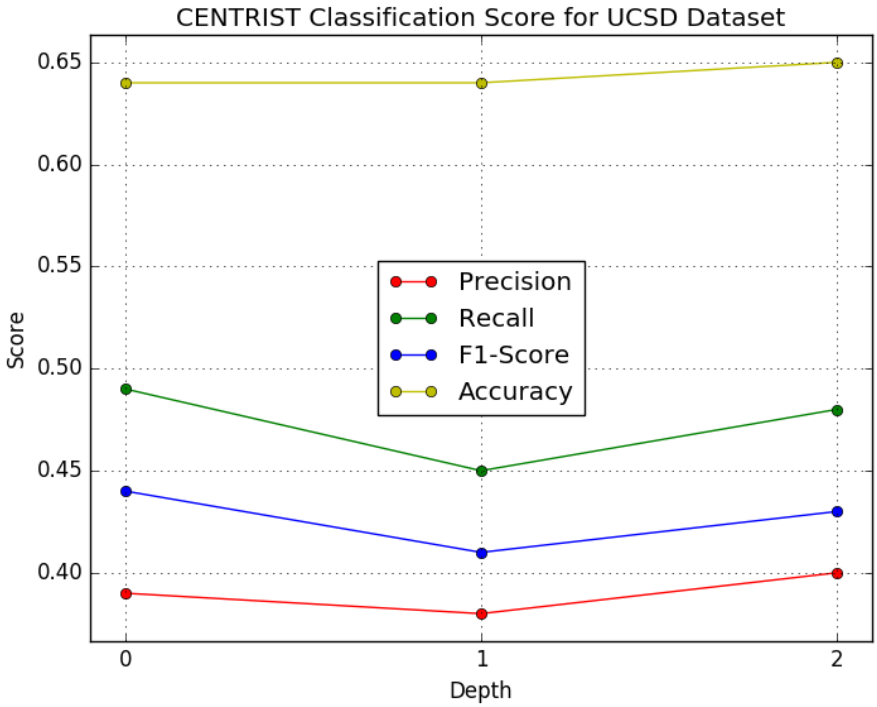


Figure 5.2: Evaluation metrics for the UCSD Dataset [1].

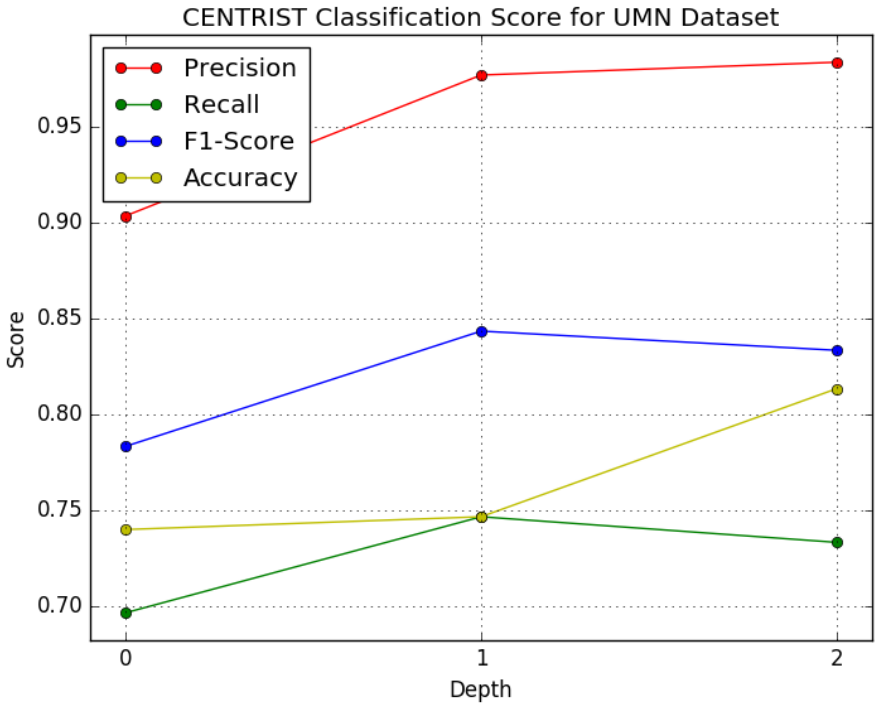


Figure 5.3: Evaluation metrics for the UMN Dataset [3].

5.3 Results for CENTRIST3D

In order to evaluate the classification performance of the CENTRIST3D descriptor, we conducted experiments to determine the influence of the pyramid depth and the VCT

Table 5.1: Best results achieved with CENTRIST for each dataset.

Dataset	Depth	Precision	Recall	F1-Score	Accuracy
Violent Flows	D=1	0.78	0.70	0.74	0.72
UCSD	D=0	0.39	0.49	0.44	0.64
UMN	D=1	0.97	0.74	0.84	0.74

Threshold over the chosen datasets. Similarly to the CENTRIST experiments, we tested our method using the same 3 depths, as shown in Figures 5.4, 5.5 and 5.6.

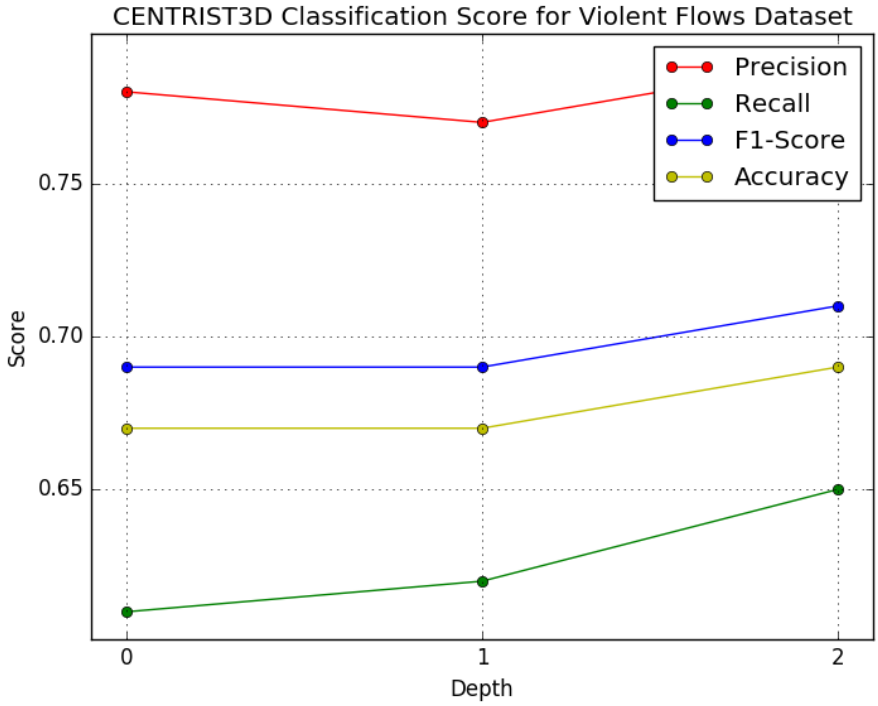


Figure 5.4: Pyramid depth influence of CENTRIST3D on Violent Flows Dataset [2].

Differently from CENTRIST, CENTRIST3D performs slightly better on UCSD Dataset using a pyramid representation of depth 1. Likewise, our method performed slightly better using a pyramid of depth 2. Besides tuning the depth parameter, we also took into account the effects of the threshold on the Volumetric Census Transform. The VCT parameter was analyzed over a specific range of values, from $C = 0$, which is equivalent to using the original Census Transform, to $C = 14$. This value was selected empirically.

We analyzed the variation of the classification metrics after increasing the VCT value and selected the VCT range which contains a global maximum. We noticed that, for all datasets, the aforementioned range always lies below $C = 14$ and the tendency of stabilizing - in the case of UCSD Dataset - or reducing - in the case of Violent Flows and UMN Datasets - the classification score after reaching the threshold value. As shown in Figure 3.6, increasing the threshold C too much leads to a smaller amount of retained features. This fact impacts negatively the classification score and can be directly observed after $C = 11$, $C = 7$ and $C = 13$, respectively, in the Violent Flows, UCSD and UMN

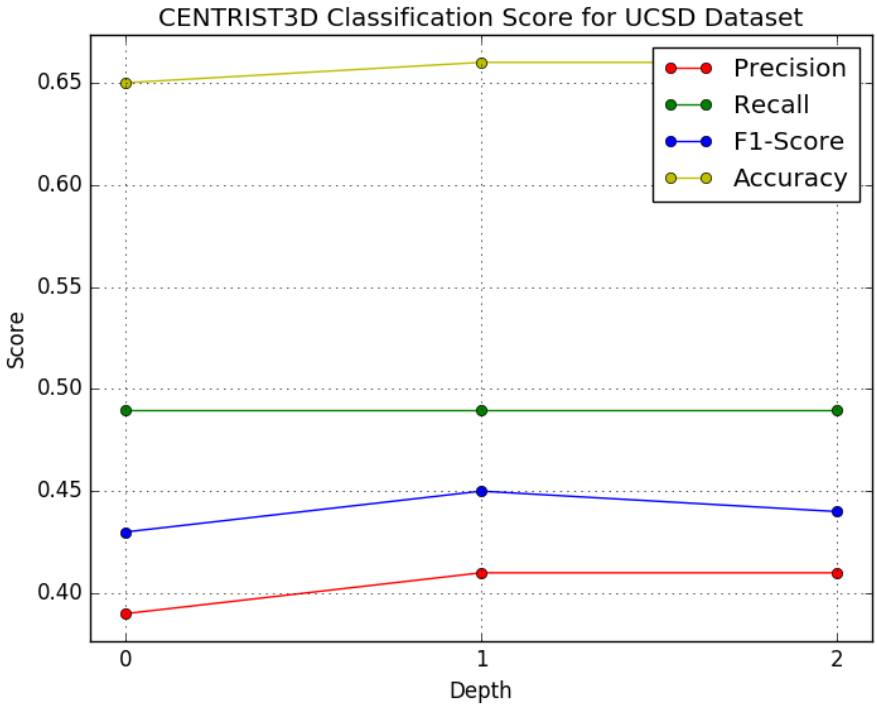


Figure 5.5: Pyramid depth influence of CENTRIST3D on on UCSD Dataset [1].

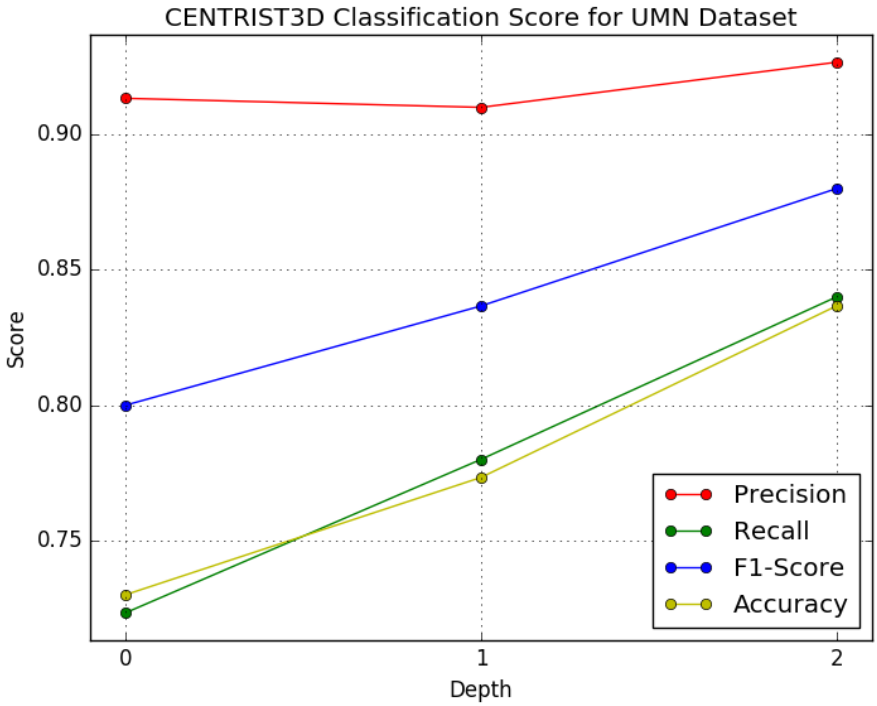


Figure 5.6: Pyramid depth influence of CENTRIST3D on UMN Dataset [3].

Datasets. Figures 5.7, 5.8 and 5.9 summarize the obtained scores for each threshold increment.

By comparing the results of our model to the original CENTRIST, we observe that our method is able to better extract spatio-temporal information, specially in scenes with

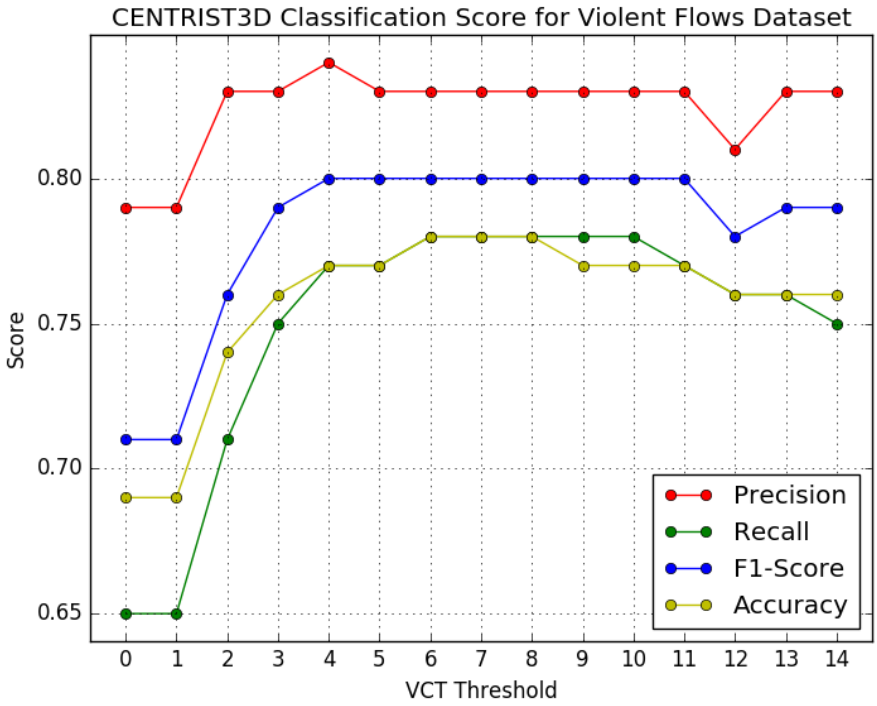


Figure 5.7: Analysis of influence of the threshold C on evaluation metrics for the Violent Flows Dataset [2].

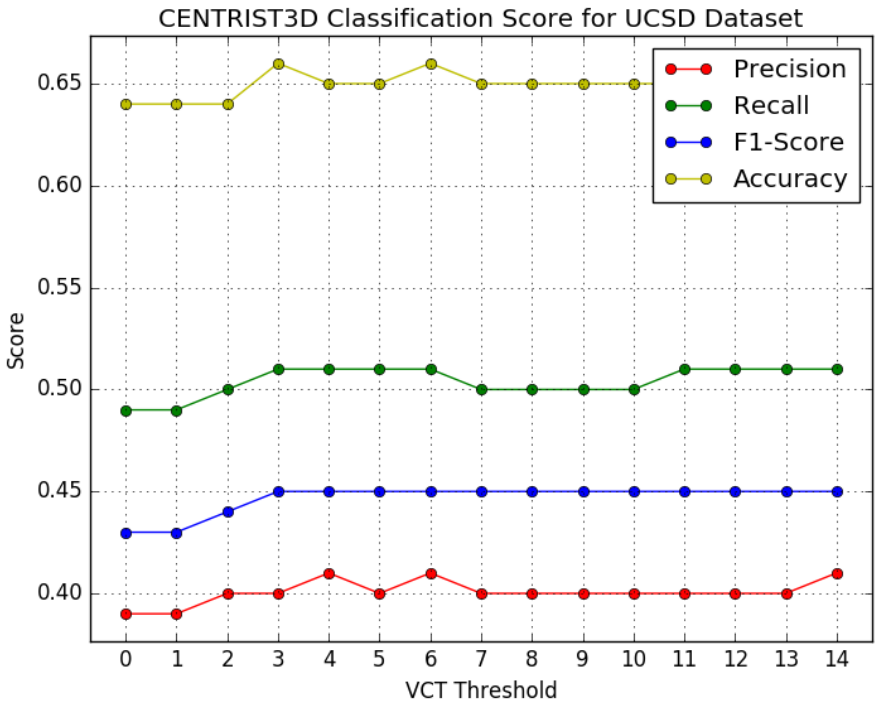


Figure 5.8: Analysis of influence of the threshold C on evaluation metrics for the UCSD Dataset [1].

subtle variation of motion. Even though the initial score for $C = 0$ on CENTRIST3D is worse than on CENTRIST for the respective result, after adjusting the VCT threshold,

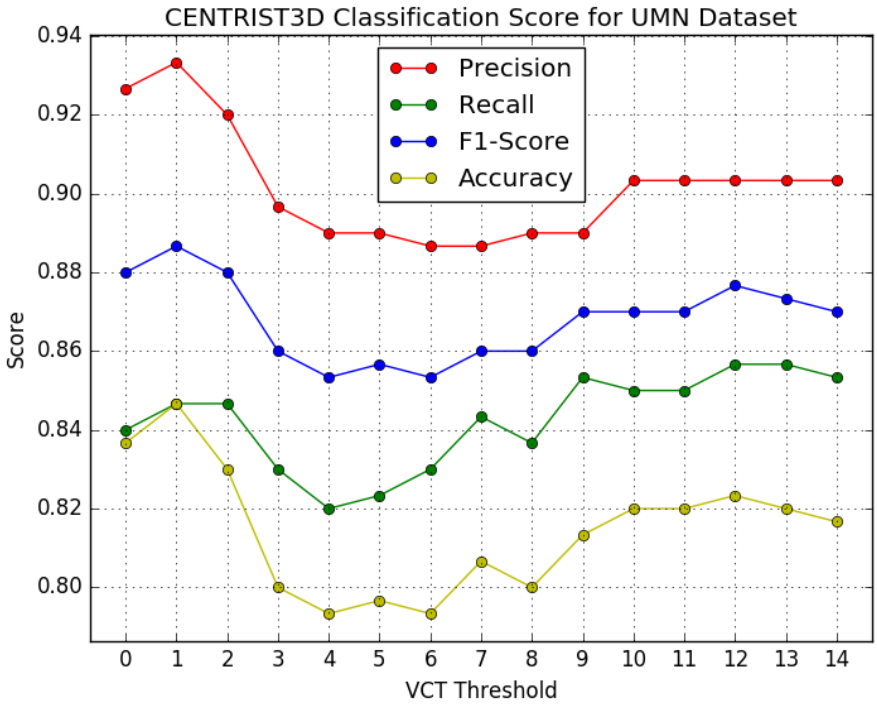


Figure 5.9: Analysis of influence of the threshold C on evaluation metrics for the UMN Dataset [3].

we were able to improve results further.

This behavior occurs in all datasets: on UCSD Dataset, CENTRIST roughly achieves a recall score of 0.49 when using a pyramid of depth 0, while CENTRIST3D slightly outperforms CENTRIST when setting the VCT threshold to 6. In the same manner, on Violent Flows Dataset, CENTRIST shows its best overall score with a pyramid of depth 1, achieving a recall score of 0.70. CENTRIST3D outperforms CENTRIST when VCT threshold is 6. Our method achieved the best overall results on UMN Dataset, reaching a recall score of 0.85. This can be explained by the very nature of the dataset: less crowded environments, less occlusions and a more consistent anomaly definition (subtle change in motion, causing the crowd to steer away from a location). Table 5.2 shows the best results achieved for each dataset.

Table 5.2: Best results achieved with CENTRIST3D on each dataset.

Dataset	Depth	Threshold	Precision	Recall	F1-Score	Accuracy
Violent Flows	D=2	C=6	0.83	0.78	0.80	0.78
UCSD	D=1	C=6	0.40	0.51	0.45	0.65
UMN	D=2	C=12	0.90	0.85	0.87	0.82

Then, we validated that our model was not overfitting by analyzing the learning curve of our best achieved results. The learning curve of UCSD Dataset is depicted in Figure 5.11. Inspecting how variance changes over the increase of examples provides strong evidence that our model is not overfitting to our data. Initially, the variance between the

training and cross-validation curves is less than 0.015 and as data is inserted, this difference is only decreased, as both curves slowly converge, reaching a minimum difference of 0.05.

Figure 5.10 shows the learning curve of Violent Flows Dataset. In a similar manner, both training and cross-validation curves also slowly converge and, thus, the small variance between the curves implies that our model did not overfit on the Violent Flows dataset.

The analysis of the learning curve for the UMN Dataset, depicted in Figure 5.12, shows the same recurring pattern we previously observed in the other datasets. Both training and cross-validation start differing by less than 0.05 and they slowly converge to a difference of less than 0.01 as the number of training examples increases. This provides strong evidence that we did not overfit our model during the experiments.

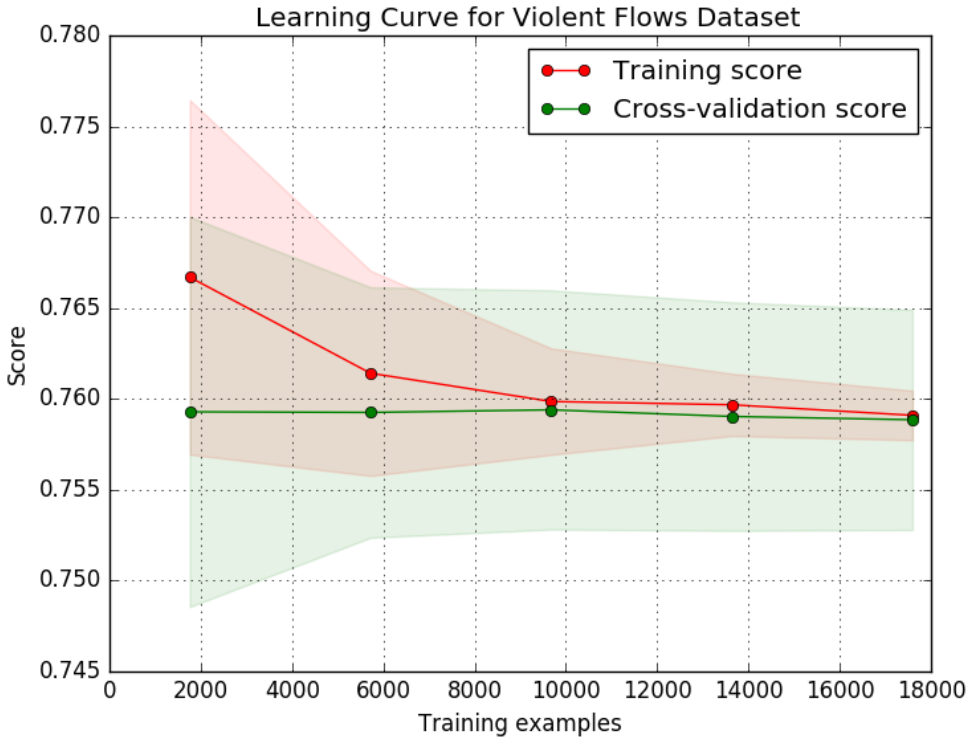


Figure 5.10: Learning Curve for Violent Flows Dataset [2].

Finally, we were able to visualize some of the correctly classified frames, depicted in Figures 5.13, 5.14 and 5.15. Our method correctly classified such UCSD frame as abnormal, which contains a cyclist, differing from the overall scene, which only contains pedestrians. The normal behavior is expressed on the right frame of Figure 5.14. Our method correctly classified it as a normal situation, as the frame contains only pedestrians.

Furthermore, our results show that we were also able to predict the abnormal event of the extract frame from Violent Flows Dataset, a fight during a match, as well as correctly predicting a calm crowd playing instruments as a normal event. Finally, the method achieved good results on UMN Dataset, reaching an accuracy of 82%.

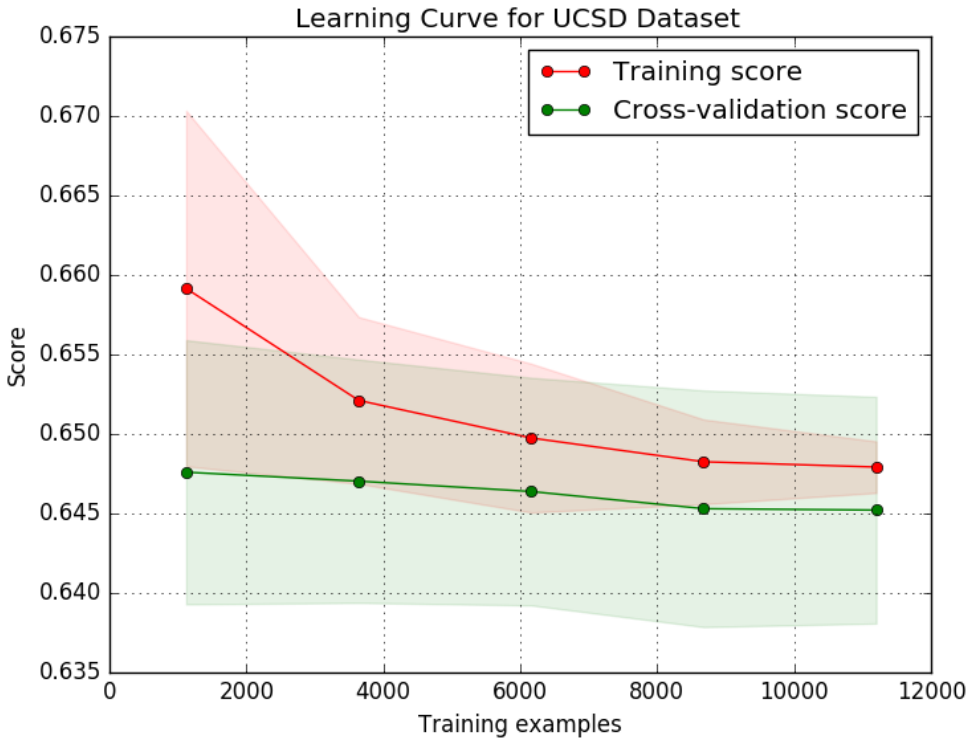


Figure 5.11: Learning Curve for UCSD Dataset [1].

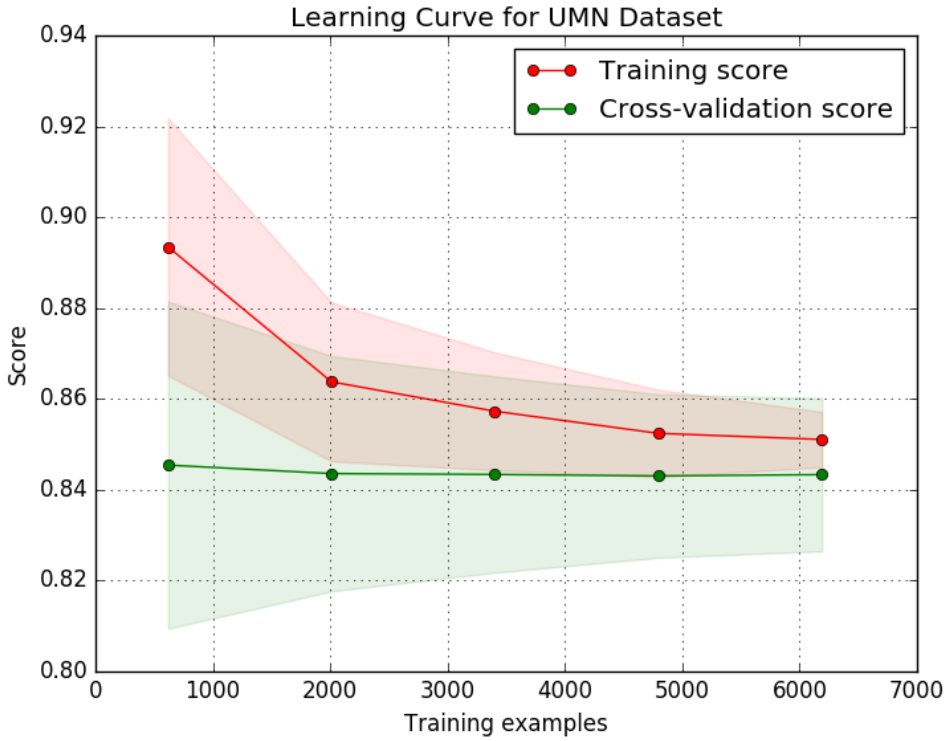


Figure 5.12: Learning Curve for UMN Dataset [3].

Figure 5.13: Correctly classified frames of Violent Flows Dataset [2], where crowd violence is an anomaly.

Figure 5.14: Correctly classified frames of UCSD Anomaly Detection Dataset [1], where bicycles are anomalies.

Figure 5.15: Correctly classified frames of UMN Dataset [3], where crowd dispersion is an anomaly.

5.4 Result Comparison

To support our decision of using the Naïve Bayes classifier [60], we provide evidences in Table 5.3 of the best achieved results by comparing the SVM [36] and the Random Forest [61] classifiers. From the table, we can observe that Naïve Bayes obtained a lower Equal Error Rate on UCSD Dataset and greater accuracy rates and AUC value on the other datasets, outperforming both SVM and Random Forest classifier, which made us select Naïve Bayes over the other options.

Next, we summarized in the following tables some results obtained with approaches available in the literature. Table 5.4 shows accuracies from methods applied to the Violent Flows Dataset. Similarly, Tables 5.5 and 5.6 summarize results achieved on UCSD and UMN Datasets.

Although our method did not outperform the state-of-the-art results, we were able to reach a reasonable score on the Violent Flows Dataset and on the UMN Dataset. This

Table 5.3: Comparison of results achieved by different classifiers using the CENTRIST3D descriptor.

Classifier	UCSD (EER)	Violent Flows (ACC)	UMN (AUC)
SVM (Linear Kernel)	47%	56%	0.67
Random Forest	44%	69%	0.81
Naïve Bayes	41%	78%	0.92

poses a good evidence that the CENTRIST3D descriptor might be better suited for tasks that contain fast changes in motion and texture, such as action recognition. Nevertheless, our method performed poorly on the UCSD Dataset, barely surpassing a coin toss.

Table 5.4: Result comparison of the CENTRIST3D descriptor to other methods available in the literature on Violent Flows Dataset [2].

Algorithm	Accuracy
HOG [62]	57.43%
HOF [62]	58.53%
LTP [63]	71.53%
CENTRIST3D	78.00%
Violent Flows [2]	81.30%
Fluid Mechanics [25]	92.30%

Table 5.5: Result comparison of the CENTRIST3D descriptor to other methods available in the literature on UCSD Dataset [1].

Algorithm	Equal Error Rate (EER)
CENTRIST3D	41%
MPPCA [64]	40%
Social Force + MPPCA [1]	32%
Sparse Reconstruction [28]	19%
Local Statistical Aggregates [29]	16%
AMDN [31]	16%
Detection at 150 FPS [65]	15%

5.5 Performance Analysis

In order to evaluate how efficient our descriptor is in terms of speed, we benchmarked the computational time for the feature extraction step in the same datasets that we evaluated our last experiments. To minimize the side effects of third-party processes, we executed each test 10 times. Computational time not directly related to the feature extraction stage, such as reading the datasets from disk, library initialization and resource cleanup, was not included in the measurements.

Table 5.6: Result comparison of the CENTRIST3D descriptor to other methods available in the literature on UMN Dataset [3].

Algorithm	Area Under the Curve (AUC)
Pure Optical Flow [24]	0.8400
CENTRIST3D	0.9230
Social Force [24]	0.9600
Cong et al. [28]	0.9780
Chaotic Invariant [66]	0.9940
Li et al. [30]	0.9950
OADC-SA [67]	0.9967

Additionally, we assessed the feature extraction using a single-threaded implementation, as well as a multi-threaded implementation - with 2 and 4 threads - using the OpenMP library. This setup is repeated for both CENTRIST and CENTRIST3D descriptors and their total execution time is shown, respectively, in Figures 5.16 and 5.17.

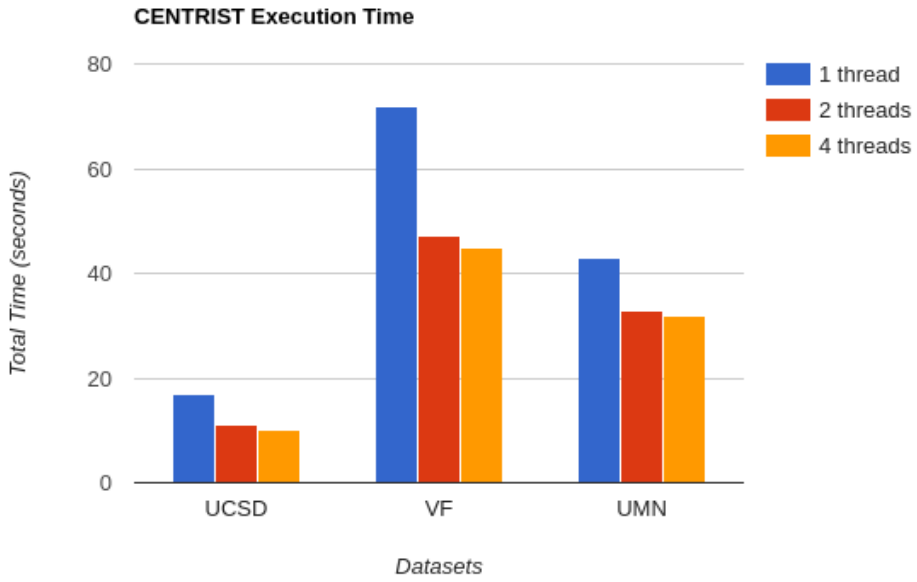


Figure 5.16: Total execution time comparison of the CENTRIST descriptor in all evaluated datasets.

It is possible to infer from the graphs that both algorithms benefit from the workload division, across all datasets, almost doubling speed when switching from a single-threaded implementation to a double-threaded one. Furthermore, when switching from 2 to 4 threads, we see little gain, mostly because of the frame size and workload division.

Our main strategy for improving speed consists in dividing the Census Transform calculation over the threads, since they are all independent from each other. Still, frame size plays a decisive role and even though switching to a multi-threaded environment may yield an additional performance gain of approximately 35%, splitting Census Transform

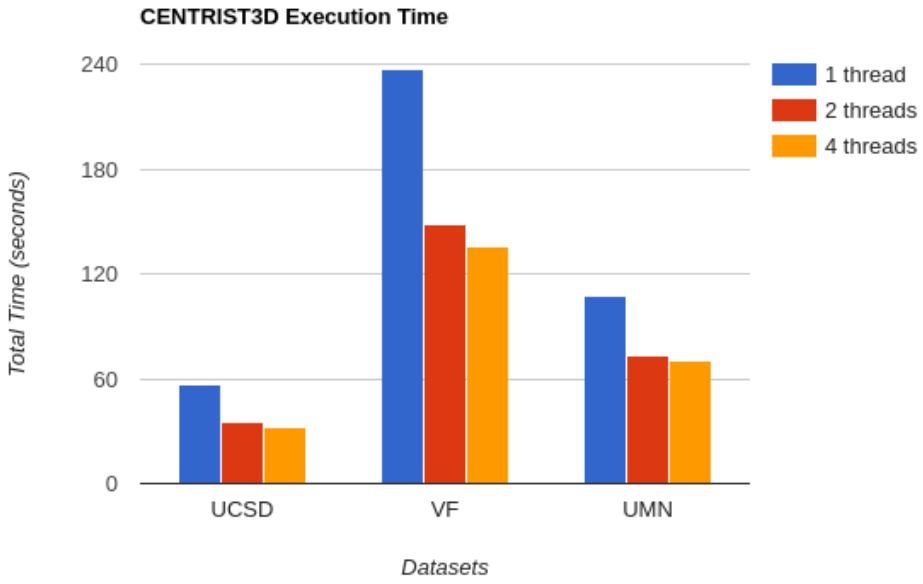


Figure 5.17: Total execution time comparison of the CENTRIST3D descriptor in all evaluated datasets.

calculations over 4 threads achieves a maximum of 10% of improvement on the 2-thread version.

Therefore, we can conclude from Tables 5.7 and 5.8 that the CENTRIST3D descriptor is at least 58% slower than CENTRIST in all datasets, while still being able to reach enough frames per second to be executed in real time, even when utilizing the single-threaded implementation, allowing us to conclude that CENTRIST3D can be used in real-time applications.

Table 5.7: CENTRIST worst and best frame-per-second (FPS) rate for each dataset.

Dataset	Worst FPS (1 Thread)	Best FPS (4 Threads)
Violent Flows	305	488
UCSD	823	1400
UMN	179	241

Table 5.8: CENTRIST3D worst and best frame-per-second (FPS) rate for each dataset.

Dataset	Worst FPS (1 Thread)	Best FPS (4 Threads)
Violent Flows	92	161
UCSD	245	437
UMN	72	110

Chapter 6

Conclusions and Future Work

Crowd anomaly detection is a complex subject due to many constraints, such as crowd density, increasing length of video sequences, subjectivity of normalcy and abnormality concepts. Nevertheless, it is an active area of research with applications in several different domains, particularly involving surveillance.

In this work, we analyzed crowd anomaly detection applied to the field of security. As main contributions, we evaluated the CENTRIST descriptor in the context of crowd abnormality detection and then proposed the CENTRIST3D descriptor, a modified algorithm to model spatio-temporal features. Features based on texture and temporal changes are extracted without the need for removing background or tracking subjects. Local and global representations are encoded in the features by combining information across multiple scales. Finally, we employed this data to train a Naïve Bayes classifier.

Our method was evaluated on three open datasets: Violent Flows, UCSD and UMN. Our method achieved an accuracy of 78% in the Violent Flows Dataset, 51% in the UCSD Dataset, and 83% in the UMN Dataset. This provides strong evidence that our method is more suitable in tasks that contain scenes with larger visual variations and less subtle motion. Furthermore, we implemented both CENTRIST and CENTRIST3D descriptors in C++ programming language to evaluate the performance both of a single-threaded implementation and a multi-threaded implementation (with 2 and 4 threads). Results show that our descriptor is efficient to be computed and achieved over 24 FPS in all datasets, making it suitable for real-time applications.

As directions for future work, we intend to investigate how the CENTRIST3D descriptor performs in other computer vision tasks, such as recognition of action and activities [68, 69, 70, 71, 72, 73]. Additionally, we intend to incorporate pixel-level anomaly detection information, evaluate the descriptor by calculating the Volumetric Census Transform (VCT) over non-adjacent frames of variable steps, and test the proposed methodology on other datasets. Finally, given the notable achievements of deep learning and neural networks applied in other computer vision problems, we also aim to investigate such approaches to crowd anomaly detection.

Bibliography

[1] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 249, p. 250, 2010.

[2] T. Hassner, Y. Itcher, and O. Kliper-Gross, "Violent flows: Real-time detection of violent crowd behavior," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–6, IEEE, 2012.

[3] UMN, "UMN Dataset," 2017. `http://mha.cs.umn.edu/Movies/Crowd-Activity-All.avi`.

[4] T. Li, H. Chang, M. Wang, B. Ni, R. Hong, and S. Yan, "Crowded scene analysis: A survey," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 367–386, 2015.

[5] J. M. Jianxin, Wu; Rehg, "CENTRIST: A visual descriptor for scene categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 1489–1501, Dec. 2010.

[6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, p. 15, 2009.

[7] B. Zhan, D. N. Monekosso, P. Remagnino, S. A. Velastin, and L.-Q. Xu, "Crowd analysis: A survey," *Machine Vision and Applications*, vol. 19, no. 5, pp. 345–357, 2008.

[8] H. Blumer, "Collective Behavior," in *Principles of Sociology*, pp. 67–121, 1951.

[9] G. Lebon, *Psychologie des Foules*. Alcan, 1895.

[10] F. H. Allport, "The group fallacy in relation to social science," in *The Journal of Abnormal Psychology and Social Psychology*, vol. 19, p. 60, 1924.

[11] T. H. Ralph and L. M. Killian, *Collective Behavior*. Prentice Hall College Div, Jan. 1987.

[12] P. M. Lee, *Bayesian Statistics: An Introduction*. John Wiley & Sons, 2012.

[13] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and Naive Bayes," *Advances in Neural Information Processing Systems*, vol. 2, pp. 841–848, 2002.

[14] J. D. Rennie, L. Shih, J. Teevan, and D. R. Karger, "Tackling the poor assumptions of Naive Bayes text classifiers," in *International Conference on Machine Learning*, vol. 3, pp. 616–623, Washington DC), 2003.

[15] I. Rish, "An empirical study of the Naive Bayes classifier," in *Workshop on Empirical Methods in Artificial Intelligence*, vol. 3, pp. 41–46, IBM New York, 2001.

[16] A. McCallum and K. Nigam, "A comparison of event models for Naive Bayes text classification," in *Workshop on Learning for Text Categorization*, vol. 752, pp. 41–48, Citeseer, 1998.

[17] M. Andersson, J. Rydell, L. St-Laurent, D. Prévost, and F. Gustafsson, "Crowd analysis with target tracking, K-means clustering and hidden Markov models," in *15th International Conference on Information Fusion*, pp. 1903–1910, IEEE, 2012.

[18] A. Basharat, A. Gritai, and M. Shah, "Learning object motion patterns for anomaly detection and improved object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2008.

[19] C. Piciarelli, C. Micheloni, and G. L. Foresti, "Trajectory-based anomalous event detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1544–1554, 2008.

[20] C. Li, Z. Han, Q. Ye, and J. Jiao, "Visual abnormal behavior detection based on trajectory sparse reconstruction analysis," *Neurocomputing*, vol. 119, pp. 94–100, 2013.

[21] W. Chongjing, Z. Xu, Z. Yi, and L. Yuncai, "Analyzing motion patterns in crowded scenes via automatic tracklets clustering," *China Communications*, vol. 10, no. 4, pp. 144–154, 2013.

[22] P.-M. Jodoin, Y. Benezeth, and Y. Wang, "Meta-tracking for video scene understanding," in *10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 1–6, IEEE, 2013.

[23] H. Mousavi, S. Mohammadi, A. Perina, R. Chellali, and V. Murino, "Analyzing tracklets for the detection of abnormal crowd behavior," in *IEEE Winter Conference on Applications of Computer Vision*, pp. 148–155, IEEE, 2015.

[24] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 935–942, IEEE, 2009.

[25] X. Wang, M. Gao, X. He, X. Wu, and Y. Li, "An abnormal crowd behavior detection algorithm based on fluid mechanics," *Journal of Computers*, vol. 9, no. 5, pp. 1144–1149, 2014.

[26] J. Xu, S. Denman, C. Fookes, and S. Sridharan, "Unusual event detection in crowded scenes using bag of LBPs in spatio-temporal patches," in *International Conference on Digital Image Computing Techniques and Applications*, pp. 549–554, IEEE, 2011.

[27] L. Kratz and K. Nishino, "Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1446–1453, IEEE, 2009.

[28] Y. Cong, J. Yuan, and J. Liu, "Sparse reconstruction cost for abnormal event detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3449–3456, IEEE, 2011.

[29] V. Saligrama and Z. Chen, "Video anomaly detection based on local statistical aggregates," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2112–2119, IEEE, 2012.

[30] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 18–32, 2014.

[31] D. Xu, Y. Yan, J. Song, and N. Sebe, "Learning deep representations of appearance and motion for anomalous event detection," *CoRR*, vol. abs/1510.01553, Oct. 2015.

[32] J. Shao, K. Kang, C. C. Loy, and X. Wang, "Deeply learned attributes for crowded scene understanding," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4657–4666, IEEE, 2015.

[33] M. Sabokrou, M. Fayyaz, M. Fathy, and R. Klette, "Fully convolutional neural network for fast anomaly detection in crowded scenes," *arXiv preprint 1609.00866*, 2016.

[34] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[35] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[36] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[37] S. J. Wright, R. D. Nowak, and M. A. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2479–2493, 2009.

[38] C. De Boor and J. R. Rice, "Least squares cubic spline approximation I-fixed knots," Tech. Rep. CSD TR 20, Computer Science Department, Division of Mathematical Sciences, Purdue University, 1968.

[39] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[40] C. Tomasi and T. Kanade, *Detection and tracking of point features*. Carnegie Mellon University, Pittsburgh, USA: School of Computer Science, 1991.

[41] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, Jan. 2003.

[42] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[43] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, no. 5, p. 4282, 1995.

[44] R. Mattivi and L. Shao, "Human action recognition using LBP-TOP as sparse spatio-temporal feature descriptor," in *International Conference on Computer Analysis of Images and Patterns*, pp. 740–747, Springer, 2009.

[45] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[46] R. Kindermann and L. Snell, *Markov Random Fields and their Applications*, vol. 1. American Mathematical Society, 1980.

[47] A. B. Chan and N. Vasconcelos, "Mixtures of dynamic textures," in *Tenth IEEE International Conference on Computer Vision*, vol. 1, pp. 641–647, IEEE, 2005.

[48] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[49] X. Zeng, W. Ouyang, and X. Wang, "Multi-stage contextual deep learning for pedestrian detection," in *IEEE International Conference on Computer Vision*, pp. 121–128, 2013.

[50] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[51] J. Shao, C. Change Loy, and X. Wang, "Scene-independent group profiling in crowd," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2219–2226, 2014.

[52] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 2169–2178, IEEE, 2006.

[53] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[54] G. Zhao and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, 2007.

[55] E. Sousa and H. Pedrini, "CENTRIST3D: A spatio-temporal descriptor for abnormality detection in video sequences," *The Visual Computer Journal (submitted)*, 2017.

[56] C. Meek, B. Thiesson, and D. Heckerman, "The learning-curve sampling method applied to model-based clustering," *Journal of Machine Learning Research*, vol. 2, no. Feb, pp. 397–418, 2002.

[57] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[58] S. Van Der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy array: A structure for efficient numerical computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, 2011.

[59] L. Dagum and R. Menon, "OpenMP: An industry standard API for shared-memory programming," *IEEE Computational Science & Engineering*, vol. 5, no. 1, pp. 46–55, 1998.

[60] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, no. 2-3, pp. 131–163, 1997.

[61] T. K. Ho, "Random decision forests," in *Third International Conference on Document Analysis and Recognition*, vol. 1, pp. 278–282, IEEE, 1995.

[62] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2008.

[63] L. Yeffet and L. Wolf, "Local trinary patterns for human action recognition," in *IEEE 12th International Conference on Computer Vision*, pp. 492–497, IEEE, 2009.

[64] J. Kim and K. Grauman, "Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2928, IEEE, 2009.

[65] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 fps in Matlab," in *IEEE International Conference on Computer Vision*, pp. 2720–2727, 2013.

[66] S. Wu, B. E. Moore, and M. Shah, "Chaotic invariants of Lagrangian particle trajectories for anomaly detection in crowded scenes," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2054–2060, IEEE, 2010.

[67] Y. Yuan, J. Fang, and Q. Wang, "Online anomaly detection in crowd scenes via structure analysis," *IEEE Transactions on Cybernetics*, vol. 45, no. 3, pp. 548–561, 2015.

[68] M. de Alcantara, T. Moreira, and H. Pedrini, "Motion silhouette-based real time action recognition," in *18th Iberoamerican Congress on Pattern Recognition*, vol. Lecture Notes in Computer Science - 8259, (Havana, Cuba), pp. 471–478, Springer-Verlag, Nov. 2013.

[69] M. de Alcantara, T. Moreira, and H. Pedrini, "Real-time action recognition based on cumulative motion shapes," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, (Florence, Italy), pp. 2941–2945, May 2014.

[70] T. Moreira, M. Alcantara, H. Pedrini, and D. Menotti, "Fast and accurate gesture recognition based on motion shapes," in *20th Iberoamerican Congress on Pattern Recognition*, vol. Lecture Notes in Computer Science - 9423, (Montevideo, Uruguay), pp. 247–254, Springer-Verlag, Nov. 2015.

[71] M. Alcantara, T. Moreira, and H. Pedrini, "Real-time action recognition using a multilayer descriptor with variable size," *Journal of Electronic Imaging*, vol. 25, pp. 013020.1–013020.9, Jan–Feb 2016.

[72] M. de Alcantara, T. Moreira, H. Pedrini, and F. Flórez-Revuelta, "Action identification using a descriptor with autonomous fragments in a multilevel prediction scheme," *Signal, Image and Video Processing*, vol. 11, pp. 325–332, Feb. 2017.

[73] T. Moreira, D. Menotti, and H. Pedrini, "First-person action recognition through visual rhythm texture description," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, (New Orleans, LA, USA), Mar. 2017.