PhD Thesis

"Contributions to the Problem of Fight Detection in Video"

Author
Ismael Serrano Gracia

Directors
Oscar Deniz Suarez
María Gloria Bueno García

Date

A mi hermanas, mis padres, mis abuelos y Mairena.
Os quiero.

Lucha por tus sueños, o los demás te impondrán los suyos
(Pablo Coelho)

Abstract

Whereas action detection is an important subject of research in computer vision, the detection of particular events such as violence, aggression or fights may be of direct use for video rating and in different video surveillance scenarios such as psychiatric wards, prisons or even in camera smartphones. In this thesis, a comprehensive review of state-of-the-art research in violence detection is provided. Publicly available violence datasets are also reviewed. Moreover, four novel methods are proposed for this task.

Contents

Figure List

Table List

Abbreviations

PhD Philosophiae Doctor

CCTV Closed-Circuit Television

STEC Spatio-Temporal Elastic Cuboid

UT UT-Interactions

ICCV International Conference of Computer Vision

NHL National Hockey League

VSD Violent Scenes Dataset

ROC Receiver Operating Characteristic

AUC Area Under Curve

TP True Positive

TN True Negative

FP False Positive

FN False Negative

TPR True Positive Ratio

FPR False Positive Ratio

PCRM Pixel Change Ratio Map

LBMD Local Binary Motion Descriptors

SCFL Semi-Supervised Cross Feature Learning

LDA Linear Discriminant Analysis

QDA Quadratic Discriminant Analysis

HoG Histogram of Oriented Gradients

KNN K-Nearest Neighbors

SVM Support Vector Machine

AdaBoost Adaptive Boosting

MPGD Motion-Pose Geometric Descriptor

RF Random Forest

VCT Violent Crowd Texture

GLCM Gray-Level Co-Occurrence Matrix

MWE Maximum Warping Energy

STIP Spatio-Temporal Interest Points

SIFT Scale-Invariant Feature Transform

MoSIFT Modified Scale-Invariant Feature Transform

BoW Bag-of-Words

HIK Histogram Intersection Kernel

MCF Motion Co-Ocurrence Feature

KDE Kernel Density Estimation

RIMOC Rotation-Invariant feature modeling MOtion Coherence

WLD Weber Local Descriptor

LaSIFT Lagrangian SIFT

OFCH Optical Flow Context Histogram

PCA Principal Component Analysis

ViF VIolent Flow

OViF Oriented VIolent Flow

AMV Acceleration Measure Vector

SFA Slow Feature Analysis

ASD Accumulated Squared Derivative

MFCC Mel Frequency Cepstral Coefficient

MBH Motion Boundary Histograms

3D-CNN 3D Convolutional Neural Network

CNN Convolutional Neuronal Network

LSTM Long Short-Term Memory

LMP Local Motion Patterns

UCF101-E UCF101 Extended

FFT Fast Fourier Transform

PSF Point Spread Function

ANOVA ANalysis Of VAriance

STEC Spatio-Temporal Elastic Cuboid

UT1 UT-Interactions dataset set 1

UT2 UT-Interactions dataset set 2

CNN Convolutional Neuronal Network

HC Handcrafted Features

ReLU Rectified Linear Units

BRISK Binary Robust Invariant Scalable Keypoints

FAST Features form Accelerated Segment Test

CHAPTER 1

INTRODUCTION

This chapter gives an introduction to this Philosophiae Doctor (PhD) thesis. While action detection has become an important line of research in computer vision, the detection of particular events such as violence, aggression or fights, has been relatively less studied. These tasks may be extremely useful in several video surveillance scenarios such as psychiatric wards, prisons or even in camera smartphones. The clear practical applications have led to a surge of interest in developing violence detectors. The chapter begins by providing a general definition of Computer Vision and action detection, a similar task to violence detection. Different real applications for violence detection will be also shown.

1.1 Preamble

This PhD thesis was carried out in the VISILAB research group which is located at the Higher Technical School of Industrial Engineering of Ciudad Real. The VISILAB group leads national and international research projects related to computer vision and artificial intelligence. The work developed in this PhD thesis has been partially funded by the following research project led by Oscar Deniz Suarez, PhD.

- Design and gradual implantation methodology for a violence detector in video. Project TIN2011-24367 from Spain's Ministry of Economy and Competitiveness.

The main objective of the project is to develop different systems for automatic detection of violence on surveillance or recorded videos.

1.2 Computer Vision

Computer vision is a field that deals with methods for acquiring, processing, analysing, and understanding images and, in general, high-dimensional data from the real world in order to produce numerical or symbolic information. In a sense, the objective is to duplicate the abilities of human vision by automatically perceiving and understanding an image. Understanding in this context means the transformation of visual images (the input to the retina) into descriptions of the world that can interface with other thought processes and elicit appropriate action. This image understanding task can be seen as the disentangling of symbolic information from image data using models constructed with the aid of geometry, physics, statistics, and learning theory.

As a scientific discipline, computer vision is concerned with the theory behind artificial systems that extract information from images. Image data can take many forms, such as video sequences, views from multiple cameras, or multi-dimensional data from a medical scanner. As a technological discipline, computer vision seeks to apply its theories and models to the construction of computer vision systems. In Figure 1.1 the relation between computer vision and various other fields is shown.

Sub-domains of computer vision include scene reconstruction, event detection, video tracking, object recognition, object pose estimation, learning, motion estimation, image restoration, action detection, action recognition, etc.

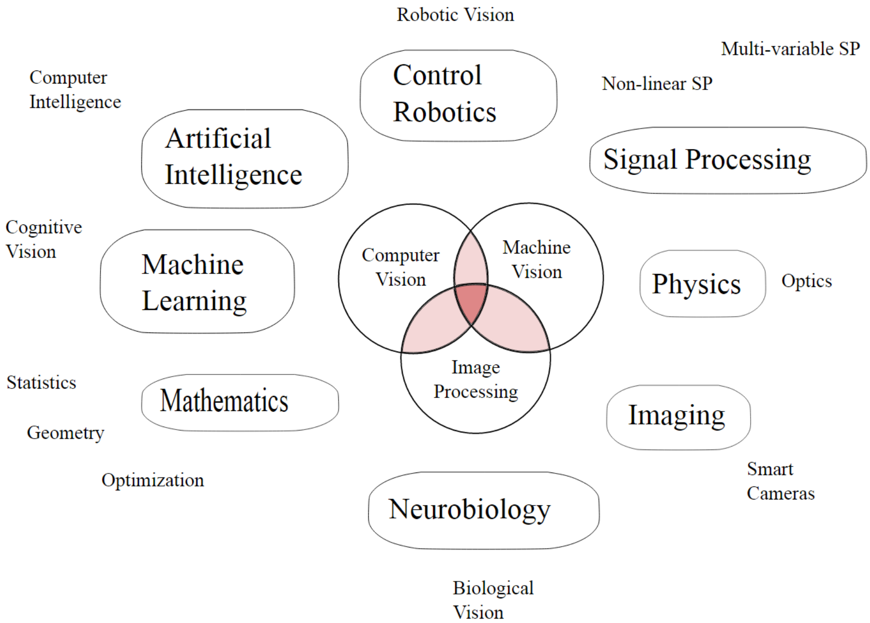

Figure 1.1: Relation between computer vision and other related fields (taken from [1])

1.3 Action Recognition

Action recognition aims to recognize the actions and goals of one or more individuals from a series of observations.

Human action recognition is an important topic in computer vision because of the many potential applications such as video surveillance, human machine interaction and video retrieval. One core problem behind these applications is automatically recognizing low-level actions and high-level activities of interest. The former is usually the basis for the latter.

1.4 Violence Detection

Action recognition techniques focus on many classes, however, the recognition of a single action may be amenable to more specific algorithms that provide either higher accuracy, better efficiency or both. The detection problem can be tackled in two ways. The first option is to process a long video sequence. Another option uses short input clips from the original sequence and applies a two-class (violent vs non-violent) recognition to each clip. The latter approach is predominant in the literature on the topic. The use of this core recognition functionality to build a fight detector is straightforward, and hence in this work "violence detection" generally refers to this approach.

Usually, "violent" and "aggressive" actions are used interchangeably in the literature. In this survey, both terms are used with the same meaning. According to a report from the 1996 Global Consultation on Violence and Health organized by the World Health Organization (WHO), violence is defined as [32]: *"The intentional use of physical force or power, threatened or actual, against oneself, another person, or against a group or community, that either results in or has a high likelihood of resulting in injury, death, psychological harm, maldevelopment, or deprivation"*. This global definition can be further divided into three main types, namely, self-inflicted, interpersonal, and collective, depending on the number of persons engaged in violence. Finally, depending on the setting and nature of violence e.g., verbal, physical, sexual, psychological and deprivation or neglect each category can be further divided [33]. In the light of this, action detection can be considered a subset of event detection, while part of violence detection can be considered a subset of action detection, see Figure 1.2.

In recent years, there has been a significant increase in the study of violence detection compared to other actions since it is closely related to our safety, security and social well being. Researchers working on the topic use different terms such as abnormal activity detection, violence detection, fight detection, etc. Figure 1.3 shows the number of published articles per year related to violence detection, where the growing interest is evident. The following key words have been strictly searched to create this graphic: "Violence Detection", "Violence Recognition", "Aggression Detection", "Aggression Recognition", "Fight Recognition" or "Fight Detection". The Scopus search engine was used.

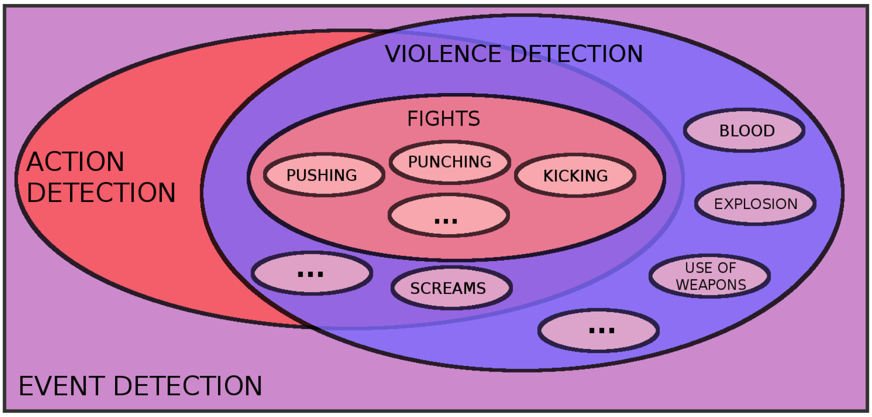

Figure 1.2: General overview of the considered events

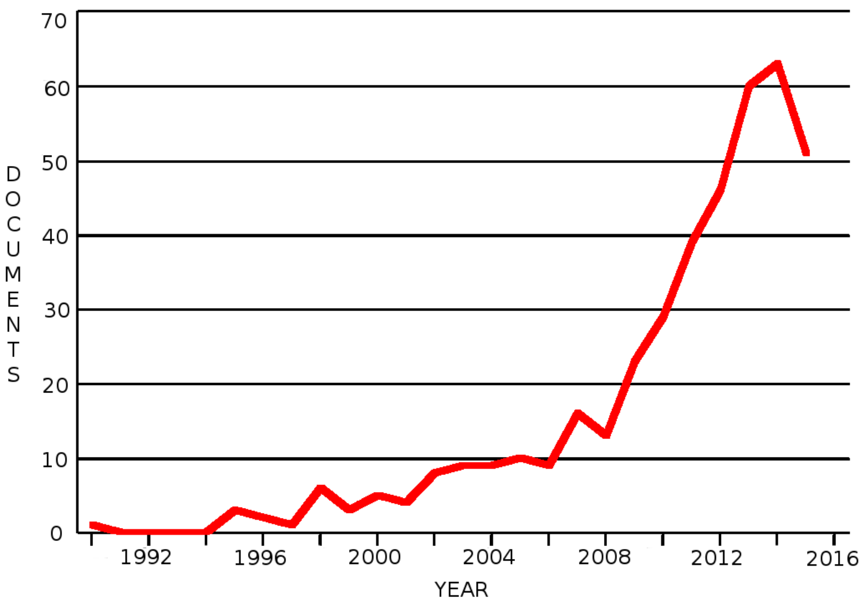

Figure 1.3: Published articles related to violence detection from 1990 to 2015.

Figure 1.4: Visual video surveillance (taken from [2])

1.5 Applications

Violence detection has an obvious importance in video surveillance. In order to keep cities safe, operators and law enforcement officials rely on video surveillance to detect and deal with theft, assaults, gang violence, drug dealing, and other criminal activities. Since the number of cameras and sensors deployed can be overwhelmingly large, see Figure 1.4, software solutions may be needed to present the data generated in a way that is meaningful and understandable to operators. Some commercial software examples are VMS[1], EyeLine[2] and Security Monitor Pro[3]. This type of software supports the central management of multiple cameras, recording of video and motion detection.

On the other hand, there is another type of software solutions that integrate video analytics on video surveillance systems. Intelligent video content analysis for public environment is a hot topic in the

[1] `http://www.asmag.com/showpost/20041.aspx`
[2] `http://www.nchsoftware.com/surveillance/`
[3] `http://www.deskshare.com/video-surveillance-software.aspx`

Figure 1.5: CyeWeb software application examples. Left top: person detector. Right top: unauthorized parking detector. Bottom: fire and smoking detector (taken from [3]).

video surveillance industry. The ultimate goal is to build a machine that can detect and recognize what happens or is about to happen. Once the machine notices something that requires attention, it alerts the operators or acts accordingly. One example is CyeWeb[4]. This technology has features such as object counting, illegal motion direction detection, illegal parking, suspicious object detection, loitering person detection, fire/smoke detection, etc. See some examples in Figure 1.5

The "CCTV Fightcams to detect violence"[5] project is another example of these more advanced systems. These "Fightcams" can automatically alert the police when violence breaks out in the street.

[4] `http://www.novosun.com/index.php/en/nvr-software`

[5] *http://www.cardiff.ac.uk/news/view/78769-fight-sensing-cameras-to-cut-crime-on-britains-streets*

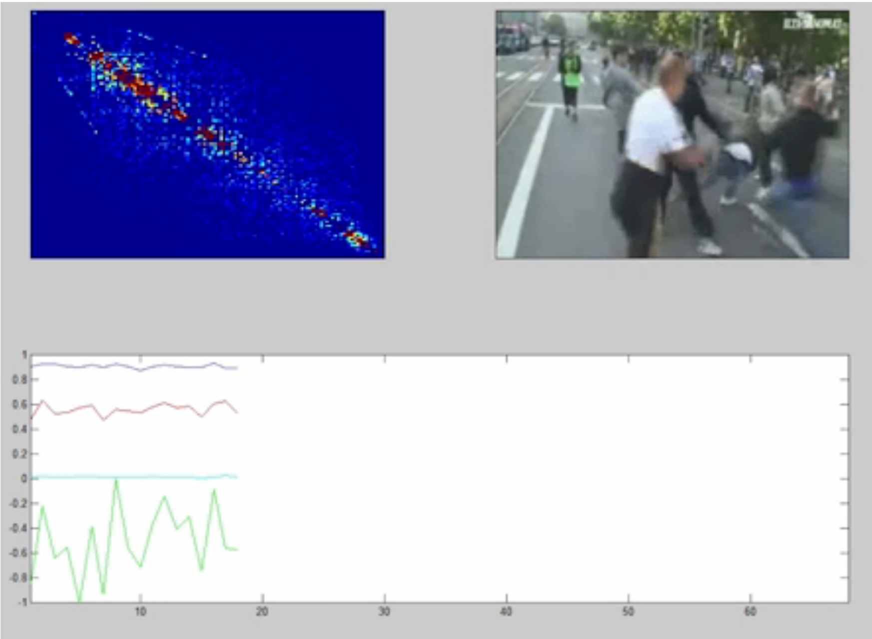

Figure 1.6: CCTV Fightcams project

The surveillance cameras use sophisticated technology to analyze how crowds are moving in public, triggering an alert when fists begin to fly. This is believed to be a world first in using CCTV to analyze people's behavior in public. Developed by Cardiff University researchers, the imaging technology alerts CCTV operators and police officers when fights are detected on city center cameras, see an example in Figure 1.6.

This work contributes to the aforementioned "intelligent" video surveillance systems.

1.6 Contributions

This PhD thesis focuses on fight and violence detection in video. The following main contributions are identified:

- A detailed review of state-of-the-art research in violence detection is provided. A categorization of violence detection methodologies is proposed. Violence detection methods are divided into seven categories according to the main algorithm used. Moreover, publicly available violence datasets are also reviewed.

- A novel method for detecting fights is proposed. Blobs of movement are first detected and then different features are used to characterize them. The proposed method makes no assumptions on number of individuals (it can be also used to detect vandalism).

- Inspired by results in human perception of other's actions, another novel method for detecting fights is proposed, using acceleration patterns as the main feature.

- The STEC Trajectories method for fight detection is proposed. This method is based on the use of blob movements to create trajectories that capture and model the different motions that are specific to a fight.

- Inspired by psychophysics experiments suggesting that motion features may be more important for this specific task, a hybrid approach for detecting fights is proposed. The method utilizes a hybrid framework by using both handcrafted and CNN learned features to recognize aggressive behaviors in videos.

1.7 PhD Thesis structure

This section provides an overview of the chapters of this PhD thesis starting from the second chapter.

- Chapter 2: State of the Art
  This chapter shows different public violence datasets. State-of-the-art violence detection methods are described and compared.

- Chapter 3: Motion blob characterization method
  This chapter describes this novel method for detecting fights in

video. The method is assessed using three different datasets and compared with five other methods.

- Chapter 4: Motion blur method
  This chapter explains the proposed method to detect fights in video. The method is tested using three different datasets and compared with five other methods.

- Chapter 5: STEC Trajectories method
  This chapter exposes a novel approach to detect fights in video. The proposed method is assessed using four different datasets and compared to nine related methods.

- Chapter 6: Convolutional neuronal network method
  This chapter describes this novel method to detect fights in video. The proposed method is tested using two different datasets and compared to other related methods.

- Chapter 7: Conclusions
  This chapter presents the conclusions reached in this PhD thesis and the scientific publications generated.

- Bibliography

- Appendix A
  This appendix explains the used measures for the exposed methods to classify violence actions.

CHAPTER 2

STATE OF THE ART

This chapter is organized as follows. Section 2.1 reviews publicly available violence datasets. Section 2.2 describes the most important methods for violence detection. Section 2.3 provides a comparison of the results of these methods on public violence datasets. Finally, in Section 2.4 the main conclusions are outlined.

2.1 Datasets

In this section, the main public video datasets for violence or fight detection are reviewed. Action datasets have a large and varied repertoire of different actions or activities. In some of them a part of the datasets also includes violent actions. On the other hand, violence datasets only focus on violent elements such as fists, explosions, collisions, kicks, fights, etc. as opposed to the other normal actions. Therefore, we divide datasets into action Datasets with violent actions and Violence Datasets.

2.1.1 Action Datasets with Violent Actions

Action datasets have been created for studying generic action recognition methods. Some of these datasets also contain a number of violent actions that can be used to assess violence detection methods. There are four important datasets that correspond to this type. These are shown below.

CAVIAR

Part of the work in the *CAVIAR*[1] project [4], which started in 2002, included the creation of a video dataset for action recognition. The *CAVIAR* dataset includes 9 activities: walking, browsing, slump, left object, meeting, fighting, window shop, shop entering, and shop exiting. The videos were recorded at two different places. The first set of video clips was filmed at the entrance lobby of the INRIA Labs, France. The second set was captured along and across the hallway in a shopping center in Lisbon. Set 1 contains violence actions. For each scenario, a ground truth file expressed in CVML is constructed. The file contains the bounding box coordinates, an activity label (appear, disappear, occluded, inactive, active, walking, running), and a scenario label (fighter role, browser role, left victim role, leaving group role, walker role, left object role) for each individual. Also, for each frame, a situation label (moving, inactive, browsing) and a scenario label (browsing, immobile, walking, drop down) are provided. Further information about the ground truth can be obtained in [34]. The *CAVIAR* project produced a significant amount of publications dedicated to different applications such as target detectors, activity recognition, human activity monitoring, clustering of trajectories, human activity identification, motion segmentation, tracking, multiagent activity recognition or violence detection. Some frames of the dataset are shown in Figure 2.1.

BEHAVE

The *BEHAVE*[2] project [5] started in 2004 under the coordination of the School of Informatics, University of Edinburgh. The project investigates two novel computer-based image analysis processes to prescreen video sequences for abnormal or crime-oriented behavior.

[1] `http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/`
[2] `http://groups.inf.ed.ac.uk/vision/BEHAVEDATA/INTERACTIONS/`

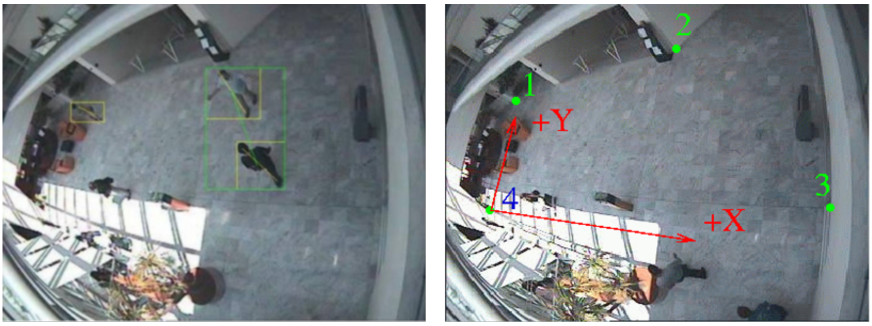

Figure 2.1: Two examples of typical frames in the *CAVIAR* dataset (set 1), recorded at INRIA Labs (taken from [4]).

The objectives of the project are: (1) to investigate and extend methods for classifying the interaction among multiple individuals, being capable of discriminating between subtly different behaviors; (2) to develop methods for flow-based analysis of the behavior of many interacting individuals; (3) to apply the results of these two approaches to the detection of criminal or dangerous situations in interactions between small groups and crowd situations; (4) to filter out image sequences where uninteresting normal activity is occurring.

The *BEHAVE* ground truth encodes target positions in a XML variant. A sample frame with overlaid target bounding boxes is shown in Figure 2.2. The dataset contains two violent actions: chases and fights. Overall, this dataset only has a 3.5% of violent actions.

UT-Interaction Dataset

The *UT*[3] dataset [35, 6], which was published in 2009, contains different videos of 6 classes of human-human interactions: shaking hands, pointing, hugging, pushing, kicking and punching. Also, the ground truth labels for these interactions are provided, including time intervals and bounding boxes. There are a total of 20 video sequences, the lengths are around 1 minute. Each video contains at least one execution per interaction, providing an average of eight examples

[3] `http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html`

Figure 2.2: Two example frames from the *BEHAVE* dataset with marked bounding boxes (taken from [5]).

of human activities per video. Several participants with more than 15 different clothing conditions appear in the videos.

The dataset is divided into two sets. Set 1 is composed of 10 video sequences taken at a parking. These videos were recorded with slightly different zoom rate, and their backgrounds are mostly static with little camera jitter. Set 2 (the other 10 sequences) were taken in a lawn on a windy day. Background is moving slightly (e.g. trees), and they contain more camera jitter. Frame examples are shown in Figure 2.3.

UCF101 Dataset

The *UCF101*[4] dataset [7] was introduced at the 2013 International Conference of Computer Vision (ICCV), which featured the first International Workshop on Action Recognition with Large Number of Classes. *UCF101* is an action recognition dataset of realistic action videos, collected from YouTube, having 101 action categories. A frame example for each action category is shown in Figure 2.4.

With $13,320$ videos from 101 action categories, *UCF101* gives a large diversity in terms of actions and with the presence of large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc.

[4] `http://crcv.ucf.edu/data/UCF101.php`

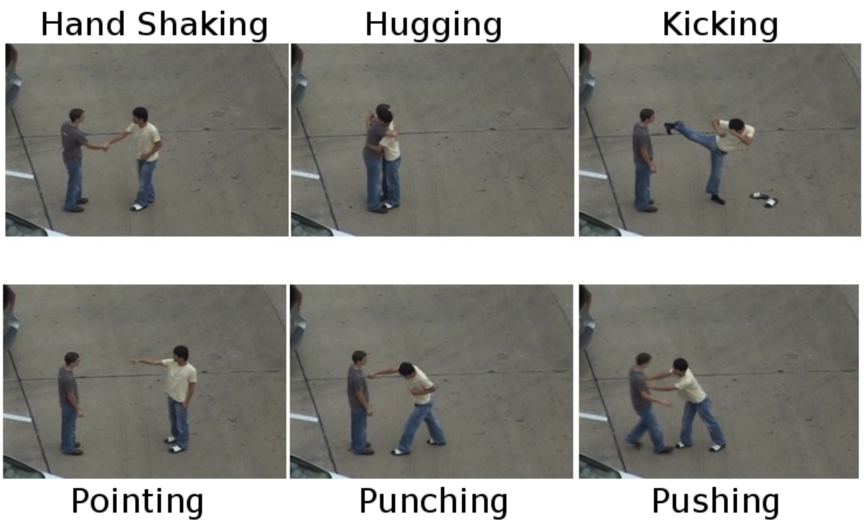

Figure 2.3: Frame examples from the *UT* dataset, one frame for each type of action (taken from [6]).

The 101 action categories are clustered into 25 groups, where each group can contain between 4 and 7 videos for each action. Videos from the same group may share some common features, such as similar background, similar viewpoint, etc.

UCF101 action categories are divided into five types: 1) Human-Object Interaction, 2) Body-Motion Only, 3) Human-Human Interaction, 4) Playing Musical Instruments and 5) Sports. A part of this dataset only contains violent actions: Punch and Sumo Wrestling categories.

2.1.2 Violence Datasets

These datasets have been specifically created with two categories: violent actions (such as pushing, fisting, explosions, collisions, kicking, fights, punching, etc.) and other non-violent actions. Four important datasets correspond to this type.

Figure 2.4: Example frames from the *UCF101* dataset, one frame for each type of action (taken from [7]).

Figure 2.5: Frame examples from the *Hockey* dataset.

Hockey Dataset

The *Hockey*[5] dataset [36] was released in 2011. The dataset consists of 1000 video clips divided into two groups, 500 fight and 500 non-fight video clips. The videos were obtained from hockey games of the National Hockey League (NHL). Most video clips have a length of 50 frames, roughly 2.5 seconds per video clip. The fight video clips usually involve two players fighting, pushing, beating, etc. The non-fight clips contain hockey sequences or isolated movements of players. Some frames of the dataset are shown in Figure 2.5. As far as the author knows, this is the most prominent violence dataset created.

Movies Dataset

The *Movies*[6] dataset [36] was introduced in 2011. This dataset consists of 200 video clips. The video clips were divided into two groups, 100 fight and 100 non-fight video clips. The fight video clips were extracted from violent sequences of movies or violent sports. Two actors are involved in some of these video clips. The non-fight

[5] `http://visilab.etsii.uclm.es/personas/oscar/FightDetection/`
[6] `http://visilab.etsii.uclm.es/personas/oscar/FightDetection/`

Figure 2.6: Frame examples from the *Movies* dataset.

video clips were extracted from public action recognition datasets. Most video clips have a length of 40 frames, roughly 2 seconds per video clip. Frame examples are shown in Figure 2.6.

Violent Scenes Dataset

The *VSD*[7] is a collection of ground-truth files based on the extraction of violent events in movies and web videos, together with high level audio and video concepts. It is intended to be used for assessing the quality of methods for the detection of violent scenes and/or the recognition of high level concepts related to violence.

The dataset was produced by Technicolor for the 2012 version and by the University of Fudan and the University of Science of Ho Chi Minh for the 2013 and 2014 versions. It is described in several publications, see [8, 37]. This dataset is used in the multimodal initiative MediaEval, as part of the 2011, 2012, 2013 and 2014 Task “Violent Scenes Detection”.

[7] `http://www.technicolor.com/en/innovation/research-innovation/scientific-data-sharing/violent-scenes-dataset`

This dataset contains 15 training and 3 test movies. The training set has a proportion of violent segments that varies from 0.76 % to 10.15 %, with an average of 5 %. Audio and visual concepts also appear. The following movies are used for the training set: “Armageddon”, “Billy Elliot”, “Eragon”, “Harry Potter 5”, “I am Legend”, “Leon”, “Midnight Express”, “Pirates Carib 1”, “Reservoir Dogs”, “Saving Private Ryan”, “The Sixth Sense”, “The Wicker Man”, “The Bourne Identity”, “Kill Bill‘” and “The Wizard of Oz”. The *VSD* dataset annotations are available on-line, see [37]. Due to evident copyright issues, the video content can not be distributed on-line. Some frames are shown in Figure 2.7.

Violent-Flows Dataset

The *Violent-Flows*[8] (or *Crowd Violence*) dataset [9] was introduced in 2012. The dataset contains 246 video clips, which are divided into two groups: 123 violent and 123 non-violent video clips. The dataset focuses on scenes with crowd violence. It is an assembled database of videos for use in both violence classification and violence detection tasks. The dataset was collected from YouTube. It therefore includes videos produced under uncontrolled, in-the-wild conditions, presenting a wide range of scene types, video qualities and surveillance scenarios. The average video clip duration is 90 frames, roughly 3.6 seconds per sequence. Some frames are shown in Figure 2.8.

Next, the different datasets are summarized using several criteria. In Table 2.1 the most important features about each dataset are shown: dataset name (Name), year of publication (Age), video source (Source), number of sequences (Seq), average number of frames per sequence (F/Seq), total length in hours (Length), percentage of violent actions (Violence), camera position (Camera) and video resolution in pixels (Resolution).

Table 2.2 shows a dataset ranking according to the estimated number of papers which directly reference the dataset.

[8] `http://www.openu.ac.il/home/hassner/data/violentflows/`

Figure 2.7: Frame examples from the *VSD* dataset for each class (taken from [8]).

Name	Age	Source	Seq	F/Seq	Length	Violence	Actors	Camera	Resolution
CAVIAR-1	2002	Recorded - Indoor	28	944	17.6h	9.5%	$\approx$ 1-10	Static	384×288
BEHAVE	2004	Recorded - Outdoor	4	56,238	37.5h	3.5%	$\approx$ 1-8	Static	640×480
UT	2009	Recorded - Outdoor	20	1,500	0.12h	50%	$\approx$ 1-6	Static	720×480
UCF101	2012	Web - In/Outdoor	13,320	120	27h	2%	-	Several	320×256
Hockey	2011	Recorded - Indoor	1,000	50	33.3h	50%	$\approx$ 1-11	Dynamic	360×288
Movies	2011	Web - In/Outdoor	200	40	5.3h	50%	$\approx$ 1-5	Several	720×576
VSD	2012	Movies - In/Outdoor	32,678	95	38.1h	5%	$\approx$ 1-20	Several	-
Violent-Flows	2012	Web - In/Outdoor	246	90	14.8h	50%	$\approx$ 5-150	Static	320×240

Table 2.1: Summary of the datasets considered. The Seq column corresponds to the number of sequences that contain the dataset, F/Seq is the average of frames per sequence, Violence is the total percentage of violent actions in the dataset and Actors column is the maximum number of involved actors in each sequence.

Figure 2.8: Frame examples from the *Violent-Flows* dataset (taken from [9]).

Rank	Name	Year	Number of references
1	UCF101	2012	255
2	UT	2009	84
3	BEHAVE	2004	47
4	Hockey	2011	43
5	Movies	2011	43
6	Violent-Flows	2012	43
7	CAVIAR	2002	31
8	VSD	2012	8

Table 2.2: Ranking of references for each dataset as retrieved from the Google Scholar search engine

Dataset	Color	Audio	Labels	BB	Sub
CAVIAR	✓	✗	✓	✓	✗
BEHAVE	✓	✗	✓	✓	✗
UT	✓	✗	✓	✓	✗
UCF101	✓	✗	✓	✗	✗
Hockey	✓	✗	✓	✗	✗
Movies	✓	✗	✓	✗	✗
VSD	✓	✓	✓	✓	✓
Violent-Flows	✓	✗	✓	✗	✗

Table 2.3: Additional features for each dataset. The BB column denotes when actions have been marked with bounding boxes. The Sub column refers to the dataset including subtitles with conversations and important events.

Table 2.3 shows additional features of each dataset, such as color, audio, labels, bounding boxes and whether subtitles are available.

2.2 Methods

State-of-the-art methods typically focus on the following basic approaches for violence detection. This section gives an overview of the most recent advances in violence detection during the past several years.

- **Spatial Descriptors.** Spatial descriptors are visual features of the contents in images. Elementary features such as shape, color, texture, frequency or location, among others can be described. Note that these descriptors are independently extracted for each image (i.e. for each frame).
- **Spatio-Temporal Descriptors.** Spatio-Temporal descriptors are formed by temporally stacking frames over a video sequence as a 3D cuboid. Spatio-temporal descriptors can be extracted from these cuboids.

- **Optical Flow.** Optical flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene.

- **Trajectories.** A trajectory is a path that a person (or object) describes as it moves. The trajectory of a tracked person in a scene is often used to analyze the activity of this person. The main problem of this approach is occlusions and fast movements that may affect tracking.

- **Deep Learning.** Traditional methods mostly rely on domain knowledge or classical features to build complex "handcrafted" features from inputs. On the contrary, deep models can act directly on the raw input and automatically extract features. Deep learning is a family of algorithms that attempt to model high-level abstractions in data by using multiple processing layers, with complex structures or otherwise composed of multiple non-linear transformations. Various deep learning architectures such as deep neural networks, convolutional deep neural networks, deep belief networks and recurrent neural networks have been proposed for violence detection.

- **Use of Audio.** Audio features can contain rich information to detect violent actions.

2.2.1 Spatial Descriptors

Nam et al. [38] present a novel technique to characterize and index violent scenes in general TV drama and movies. In particular, multiple "audio visual" signatures are exploited to create a perceptual relation for conceptually meaningful violent scene identification. Firstly, a motion sequence from a video are extracted. This sequence is processed by a temporal high-pass filtering of the 1-D wavelet transform along the time evolution of the intensity at each pixel in the spatially reduced coarse frames from 2-D wavelet decomposition of the original video. It also focuses on the temporal location of video frames, which show a noticeable increase in the number of bloody pixels, rather than the number of such pixels because blood-like colors may appear in non-violent scenes. The method uses a dataset of video clips captured from several R-rated movies ("The Untouchables", "Con Air", "Heat", "007-Golden Eye", "Under Siege 2", etc.). The method achieves an accuracy of about 88% on their dataset.

Clarin et al. [39] propose an automated system that detects violence in motion pictures. Color histograms of each pair of frames are used to compute the pairs of Euclidean distances which determine whether a scene cut operation will be performed. The system finds a representative frame within each of the scenes and these frames are passed to Self-Organizing Map for labeling of skin, blood and non-skin/non-blood pixels. Labeled pixels are grouped through Connected Components to determine which regions should be observed for possible occurrence of violent motion, these regions are characterized by the presence of relatively large skin and blood components. The pixels of these components are then cross-checked with the Pixel Change Ratio Map (PCRM) which represents the motion intensity of each pixel in all the frames of a scene. A scene is considered violent if the PCRM finds "high" motion intensity for the pixels within the skin and blood components determined earlier. The method is tested on four movies: "Kill Bill": fast paced and bloody, "Passion of the Christ": slow and bloody, "Gladiator": dark, fast/slow paced, sometimes bloody or not, and "What A Girl Wants": no violence. The method achieves an accuracy of 90% on their dataset.

Zajdel et al. [10] present a surveillance system called CASSANDRA, aimed at detecting instances of aggressive human behavior in public environments. A distinguishing aspect of CASSANDRA is the exploitation of the complementary nature of audio and video sensing to disambiguate scene activity in real-life, noisy and dynamic environments. At the lower level, independent analysis of the audio and video streams yields intermediate descriptors. At the higher level, a Dynamic Bayesian Network is used as a fusion mechanism that produces an additional aggression indication for the current scene. The method considers alternative cues based on an intuitive observation that aggressive behavior leads to highly energetic body articulation. (Pseudo-) kinetic energy of body parts using a "bag-of-points" body model is estimated. The approach relies on simple image processing operations and yields features which are highly correlated with aggression. The system is tested on a private dataset. A set of 13 audio-video clips was recorded in a train station. The video clips (between 100s and 150s) feature 2-4 professional actors who engage in a variety of activities ranging from normal (walking) through slightly excited (shouting, running, hugging), moderate aggressive (pushing, hitting a vending machine) to critically aggressive (football-supporters clashing). The recording place is taken from a platform of an actual train station (between two rail tracks, partially outdoor) and therefore incorporates realistic artifacts, like noise and vibrations from trains, variable illumination, wind, etc. See in Figure 2.9 the visual aggression features marked for the dataset. The method achieves an accuracy around 78% using their audio-visual feature fusion method on their dataset.

Chen et al. [40] implement a violence detection algorithm based on the extraction of features. The fist step is to extract Local Binary Motion Descriptors (LBMD) with a novel clustering algorithm to merge similar LBMD into a video codebook, to build a recognizer to classify behaviors. The proposed clustering algorithm is based on K-Means adding the idea of penalizing pairs that are considered as continuous components but do not belong to the same cluster. Theoretically, continuous components should be very similar in feature space. Under this assumption, the penalty is high if they do not belong to the same cluster.

Gong et al. [11] propose a three-stage method integrating visual and auditory cues to detect violence in movies. A modified semi-supervised learning technique based on Semi-Supervised Cross Feature Learning (SCFL) is exploited, since it is capable of combining different types of features and use unlabeled data to improve classification performance. Then, typical violence-related audio effects are further detected for the candidate shots. Finally, the first two-stage probabilistic outputs are integrated to generate the final inference, see the method overview in Figure 2.10. The evaluation of the proposed approach is performed in four movies involving violence: "The Terminator", "Kill Bill: Vol1", "The Rock" and "Hot Fuzz". The method

Figure 2.9: Visual aggression features are marked using CASSANDRA method [10] with their dataset (taken from [10]).

achieves a recall of 96.4% on their dataset. Recall is defined as "the number of true positives divided by the total number of elements that actually belong to the positive class".

Giannakopoulos et al. [41] present a method for detecting violence in video streams from movies, employing fusion methodologies. Towards this goal, a multi-step approach is followed based on automated auditory and visual processing probabilistic measures regarding particular audio and visual related classes. For training and test a private dataset is created with 50 videos that are ripped from 10 different movies. The overall duration of the data is 2.5 hours. These videos are manually annotated by three humans. According to the manual annotations, 19.4 % of the data are violent actions. The method achieves a classification recall of 47% on their dataset.

Giannakopoulos et al. [12] also propose a method for detecting violent content in video sharing sites. The proposed approach operates on a fusion of three modalities: audio, moving image and text data, the latter being collected from the accompanying user comments. The problem is treated as a binary classification task (violent

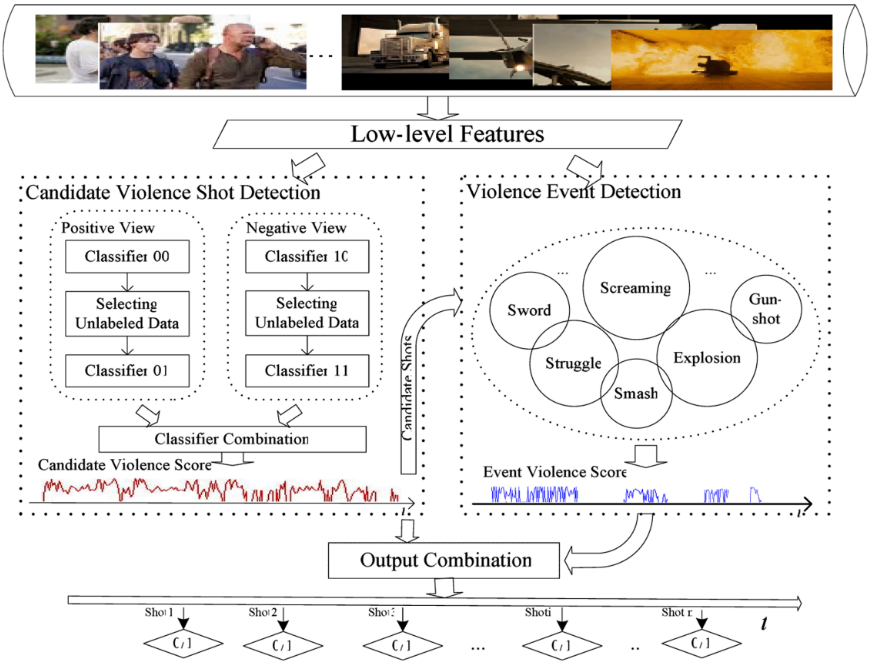

Figure 2.10: Overview of method [11] (taken from [11])

vs non-violent actions) on a 9-dimensional feature space, where 7 out of 9 features are extracted from the video clip, see the overall system architecture in Figure 2.11. A private dataset is created for training and test purposes. The videos are manually downloaded and annotated, exactly 210 videos from YouTube. The videos are chosen to cover a wide range of categories, e.g., music clips, TV interviews, news, sports and home videos. After the manual annotation process, 115 of the videos are tagged as "violent". The method achieves an accuracy of 82 % on their dataset.

Gninkoun et al. [42] propose a set of features and a classification framework for detecting violent scenes in movies. The features are extracted from audio, video, and subtitles. In the visual modality, video clip length, shot motion component, skewness of the motion vectors and video clip motion content are extracted. The detailed description of the technique used to compute the video clip motion

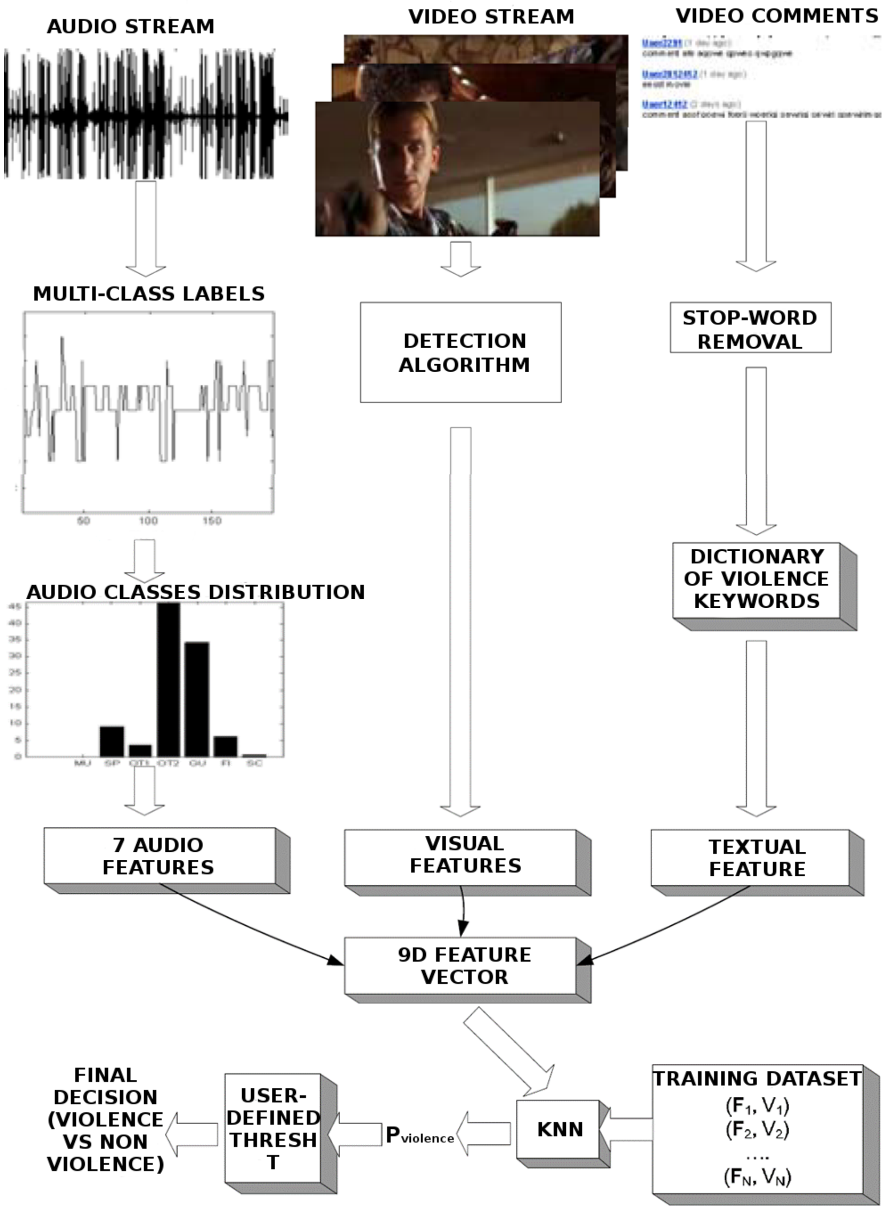

Figure 2.11: Overall system architecture [12] (taken from [12])

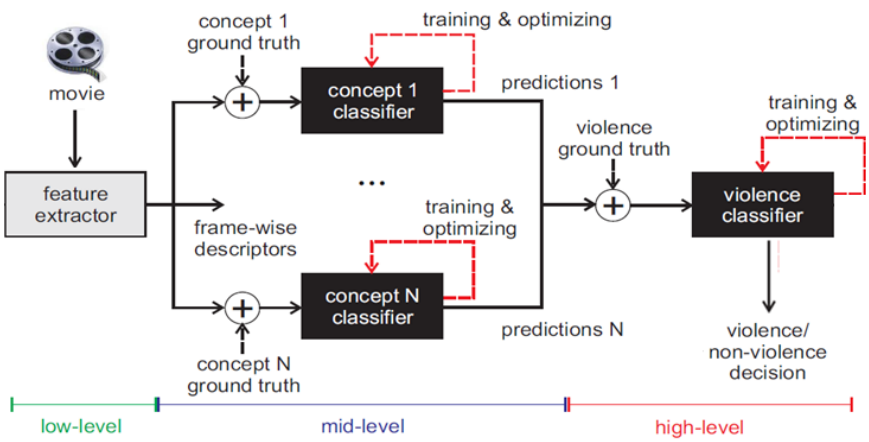

Figure 2.12: General diagram for method [13] (taken from [13])

component is given in [43]. Video clips are classified into violent and non-violent actions using Naive Bayesian, Linear Discriminant Analysis (LDA), and Quadratic Discriminant Analysis (QDA). The method is tested on the public *VSD* dataset. The result is shown and compared in Table 2.4.

Ionescu et al. [13] present a novel approach for violence detection in movies. The proposed method relies on an intermediate step consisting of prediction of mid-level violence concepts. Predicting these mid-level concepts from low-level features seem to be more feasible than directly predicting all forms of videos. The method extracts the following local descriptors: color features based on Color Naming Histogram and 81-dimensional HoG. Audio and temporal structure are also obtained. To account for temporal information a measure of visual activity is used (Cut Detector [44]), see the general diagram in Figure 2.12. Finally, a neuronal network classifier is trained by gradient descent on the cross-entropy error with back propagation. The method is tested on the public *VSD* dataset. The result is shown and compared in Table 2.4.

Serrano et al. [45, 46] present a method for fight detection in video clips. Inspired by results that suggest that kinematic features alone are discriminative for at least some actions, the method uses extreme acceleration patterns as the main feature. Motion blur entails a shift in image content towards low frequencies. Such behavior allows building an efficient acceleration estimator for video. First, the power spectrum of two consecutive frames is computed. When there is a sudden motion between two consecutive frames, the power spectrum image will depict an ellipse. The orientation of the ellipse is perpendicular to the motion direction, the frequencies outside the ellipse being attenuated. Basically, the method aims at detecting the sudden presence of such ellipse. Afterwards, when the features for every clip are extracted, K-Nearest Neighbors (KNN), SVM (with a lineal kernel) and AdaBoost classifiers are used. The proposed method is tested on the public *VSD*, *Movies*, *Hockey* and *UCF101* datasets. Moreover, in [46] other descriptors are extracted to classify these datasets, in order to compare the proposed method. The method achieves an accuracy of 85.4%, 90.1% and 93.4% on *Movies*, *Hockey* and *UCF101* datasets, respectively, see Tables 2.5 and 2.6. The *VSD* dataset results are shown in Table 2.4.

Alazrai et al. [47] propose view-invariant Motion-Pose Geometric Descriptor (MPGD) for representing human-human interactions using a Microsoft Kinect sensor to capture the interactions. The MPGD is used to classify and predict human-human interactions. In addition, the MPGD representation combines with appearance-based features increasing the performance of classification. This is due to the compensation from the appearance based representation towards reducing the effect of noise in the skeleton model obtained, when body-parts are occluded. They collected a private dataset that consists of 12 types of human-human interactions between two persons. The 12 interactions include handshaking, high-five, hugging, pushing, approaching, departing, pointing, punching and kicking. The method achieves an accuracy of 92.3% on their dataset. On the other hand, the public *UT* dataset is also used to test the method. The method gets an accuracy of 85.0% on *UT* dataset, see Tables 2.5 and 2.6.

Serrano et al. [48] develop an efficient method for violence detection in video clips. The method extracts blobs of movements from the absolute difference between consecutive image pairs. For each blob, different morphological and position descriptors are extracted such as area, centroid, compactness and perimeter. The distance between blobs is also calculated. Only the K largest blobs are selected for each image. With these descriptors, SVM, AdaBoost and RF classifiers are trained for each video clip. The method is tested on the public *Movies*, *Hockey* and *UCF101* datasets. The method achieves an accuracy of 97.8%, 82.4% and 79.5% on these datasets, respectively, see Tables 2.5 and 2.6.

Lloyd et al. [49] develop an automated method of violence detection in which a model of crowd dynamics using changes in crowd texture is proposed. This approach is termed Violent Crowd Texture (VCT). The method builds upon Haralick texture features that describe visual texture using statistics derived from grey level intensities. Gray-Level Co-Occurrence Matrix (GLCM) texture features are used, and temporal encoding is applied to create an effective method that describes crowd dynamics. The method is tested on the *Violent-Flows*, UCF Web Crowd Abnormality and UMN Crowd Abnormality datasets. The method achieves an accuracy of 91.3% on the *Violent-Flows* dataset, see Tables 2.5 and 2.6.

2.2.2 Spatio-Temporal Descriptors

Mecocci et al. [50] define a descriptor called Maximum Warping Energy (MWE) to describe the localized spatial-temporal complexity of color conformations. The CIELab model color is used. Reliability and general-purpose applicability is achieved by analyzing low-level features (the spatio-temporal behavior of colored stains), and by measuring some warping and motion parameters. In this way, it is not necessary to extract accurate target silhouettes, a critical task because of occlusions that are typical during interpersonal contacts. The method is tested on a private dataset.

Chen et al. [51] propose an automatic algorithm capable of recognizing aggressive behaviors from recorded video clips using LBMD. The method proposes an interest point detector that is based on the Harris corner detector. Instead of corner points in spatial positions, it extracts points along edges with velocity vectors by replacing the 2nd moment gradient matrix with gradient magnitudes of x, y, and

t. At each interest point, a cube is extracted that contains the spatio-temporally windowed pixel values in the video, then local binary features are computed for each cube. Local motion features extracted from the same body part contain similar motion information. Therefore, the method can cluster them to reduce the feature space into a fixed-size feature codebook using the K-means algorithm. A one-center SVM is used to train the models. The method is evaluated using the CareMedia aggression dataset and achieves an accuracy of 80%.

Lin et al. [52] present a novel method to detect violent shots in movies. The detection process is divided into two modalities: audio and video. From the audio-view, a weakly-supervised method is exploited to improve classification performance. From the video-view, a classifier to detect violent video clips is used. The auditory and visual classifiers are combined using co-training. Motion intensity and complexity are used to detect areas with fast tempo. The evaluation of the proposed approach is performed in five movies involving violence: “The Terminator”, “Kill Bill: Vol1”, “The Rock”, “Hot Fuzz” and “DOA Dead or Alive”. The method achieves an average recall of 92.02% on their dataset.

De Souza et al. [14] present a violence detector that extracts spatio-temporal features from the video sequence creating the concept of visual codebooks. The method extracts Spatio-Temporal Interest Points (STIP) [53] as features for each video clip, then these features are clustered using the “visual codebooks”, see an example in Figure 2.13. Finally, a SVM with linear kernel is used to classify the created codebooks. A private dataset is built with violent and non-violent actions. The method gives 99% of accuracy on their dataset.

Similary, in Bermejo et al. [36] local spatio-temporal features are extracted to obtain compact and descriptive representations of motions. In this approach, two prominent spatio-temporal descriptors, STIP, Scale-Invariant Feature Transform (SIFT) and Modified Scale-Invariant Feature Transform (MoSIFT) are extracted for each video clip. Afterwards, a Bag-of-Words (BoW) approach is applied to represent each video clip as a histogram over a set of visual words to generate a fixed-dimension encoding that can be processed using a

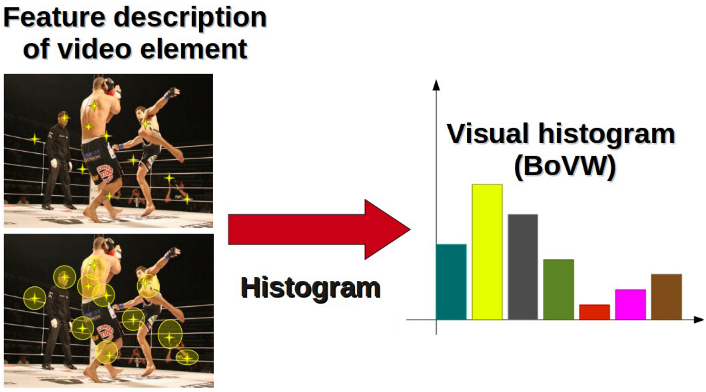

Figure 2.13: Visual histograms [14] are assembled to represent the video elements according to the occurrence of visual words in their space (taken from [14]).

standard classifier. A SVM classifier is used with different kernels: Histogram Intersection Kernel (HIK), radial basis function and Chi-Squared kernel. Two public violence datasets are introduced and used to test with the method: *Movies* and *Hockey* datasets. The method achieves an accuracy of 89.5%, and 91.7% on these datasets, respectively, see Tables 2.5 and 2.6.

Esen et al. [15] present a new method for the task of fight detection in surveillance cameras. A novel motion feature, namely Motion Co-Ocurrence Feature (MCF) is proposed. Firstly, motion vectors are extracted by using a block matching algorithm. Direction and magnitude values of motion vectors are quantized separately. Afterwards, direction and magnitude based MCF is calculated by considering both current and past motion vectors. Experimental results obtained using K-Nearest Neighbor classifier show that the proposed algorithm can discriminate fight scenes with high accuracy, see an overview of the method in Figure 2.14. The method is tested on the public *BEHAVE* and *CAVIAR* datasets. The method gets an accuracy of 83.7%, and 89.4% detecting violent actions on these datasets, respectively, see Tables 2.5 and 2.6.

Xu et al. [16] develop a method to detect violence in videos. To tackle this problem, the method employs the MoSIFT algorithm to extract low-level representations. To eliminate feature noise, Ker-

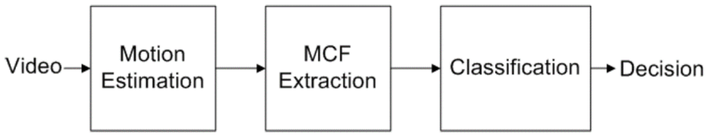

Figure 2.14: Overview of method [15] (taken from [15])

nel Density Estimation (KDE) is exploited for feature selection on the MoSIFT descriptor. In order to obtain the highly discriminative video feature, this method adopts sparse coding to further process the selected MoSIFT, see the framework of the proposed approach in Figure 2.15. The method is tested on the public *Hockey* and *Violent-Flows* datasets. The method achieves an accuracy of 94.3% and 89.1% on these datasets, respectively, see Tables 2.5 and 2.6.

Ribeiro et al. [17] present a method to detect real-life violent actions. A novel method called problem-specific RIMOC is proposed in order to capture its structure and discriminate the unstructured motions. Consecutive frames are extracted to apply the descriptor, which is based on the eigenvalues obtained from the second-order statistics of the histograms of optical flow vectors. RIMOC features are used to learn statistical models of normal coherent motion in a weakly supervised manner. A multi-scale framework applied on an inference-based method allows events with erratic motion to be detected in space and time as good candidates of aggressive events. Figure 2.16 shows an overview of the method. A structured database comprising more than 18 hours of video and 276 aggressions, both from real and lab recorded sequences is built, some with poor-quality CCTV cameras, and some with real-life violent actions. The method gives an accuracy of 82.2% on their dataset.

Mohammadi et al [18] present a novel video descriptor for violence detection based on the substantial derivative, an important concept in fluid mechanics, which captures the rate of change of a fluid property as it travels through a velocity field. Unlike standard approaches that only use temporal motion information, this descriptor exploits the spatio-temporal entangling of the substantial derivative. In particular, the spatial and temporal motion patterns are captured by

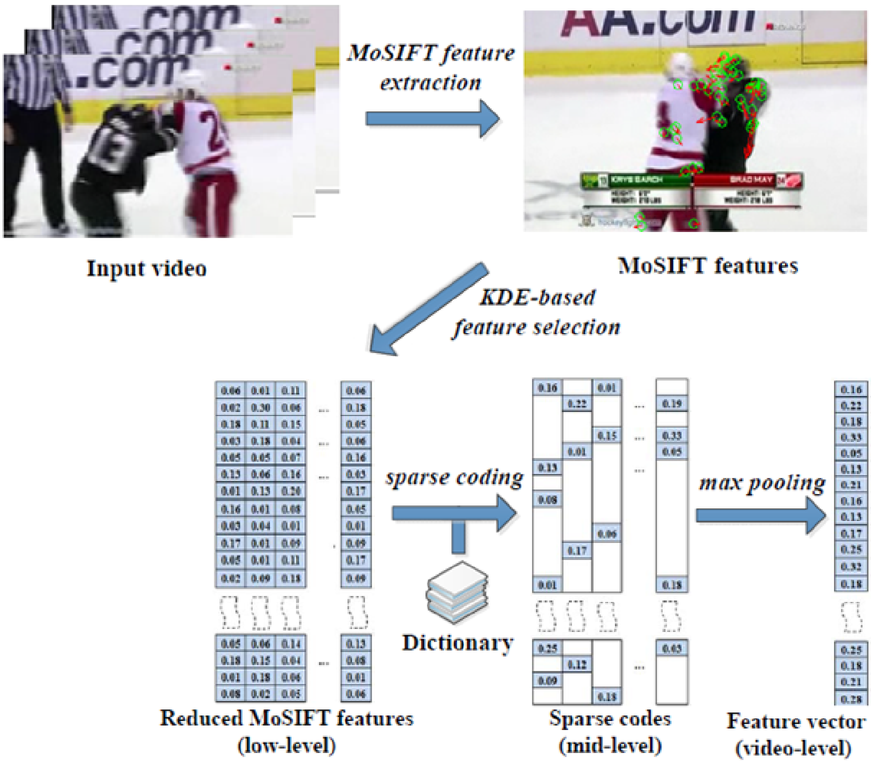

Figure 2.15: Framework of the proposed violence detection approach [16] (taken from [16]).

the convective and local accelerations respectively. After estimating the convective and local field from the optic flow, the standard BoW algorithm is applied for each motion pattern separately, and the two resulting histograms that form the final descriptor are concatenated, see the overview in Figure 2.17. The method is tested on the public *Movies*, *Violent-Flows* and *BEHAVE* datasets. The method gets an accuracy of 96.9%, 85.4% and 94.8% on these datasets, respectively, see Tables 2.5 and 2.6.

Zhang et al. [19] propose a novel descriptor for violence detection using the Motion WLD and makes it a powerful and robust descrip-

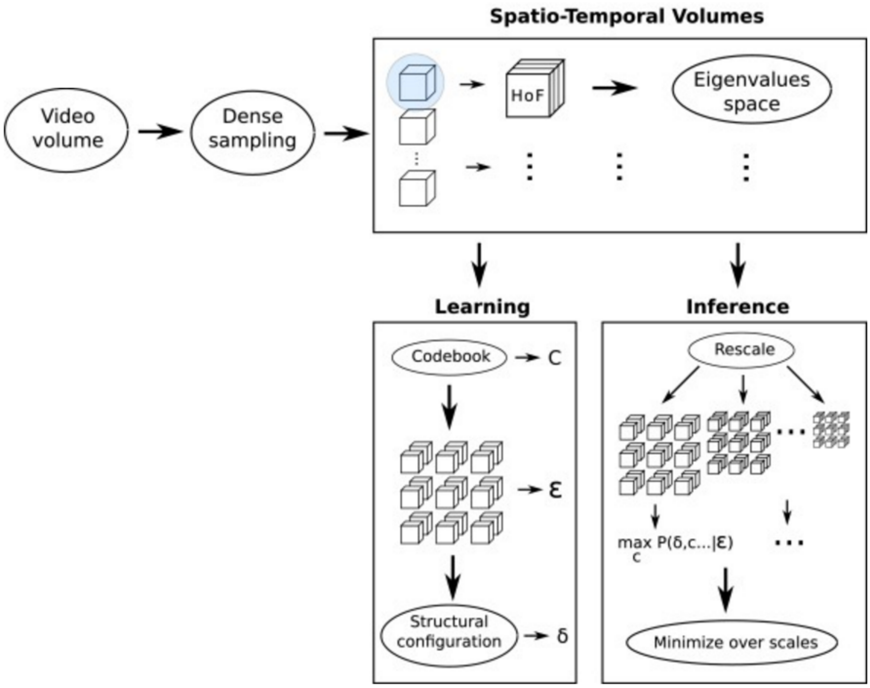

Figure 2.16: Overview of RIMOC method [17] (taken from [17])

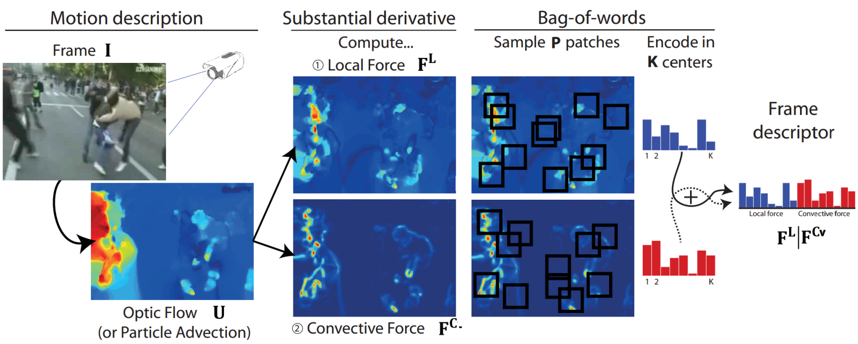

Figure 2.17: Overview of method [18] (taken from [18])

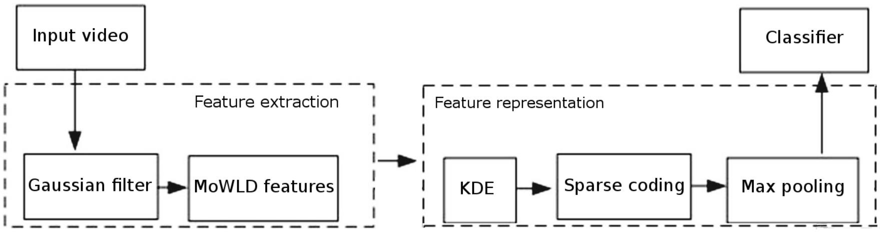

Figure 2.18: Overview of Motion WLD method [19] (taken from [19])

tor for motion images. The WLD spatial descriptions are extended by adding a temporal component to the appearance descriptor, which implicitly captures local motion information as well as low-level image information. To eliminate redundant and irrelevant features, non-parametric KDE is employed on the Motion WLD descriptor. In order to obtain more discriminative features, the sparse coding and max pooling schemes are adopted to further process the selected Motion WLD, see the overview in Figure 2.18. The method is tested on the public *Hockey*, *Violent-Flows* and *BEHAVE* datasets. The method achieves an accuracy of 94.9%, 89.8% and 87.2% on these datasets, respectively, see Tables 2.5 and 2.6.

Senst et al. [20] utilize the concept of Lagrangian measures to detect violent scenes. Lagrangian theory provides a diverse set of tools for continuous motion analysis. A novel local feature based on the SIFT algorithm that incorporates appearance and Lagrangian based motion models is proposed. The temporal interval of the used motion information is a crucial aspect and the authors also study its influence on the classification performance. Authors call the novel feature Lagrangian SIFT (LaSIFT), see an example in Figure 2.19. It is evaluated using a BoW approach. The method is tested on the public *Hockey* and *Violent-Flows* datasets. The method gets an accuracy of 94% and 92.4% on these datasets, respectively, see Tables 2.5 and 2.6.

Zhao et al. [21] propose a method to detect abnormal activities in video. The method uses spatio-temporal features and a Laplacian sparse representation is proposed to tackle this problem. To detect the abnormal activity, firstly interest points of a query video in the spatio-

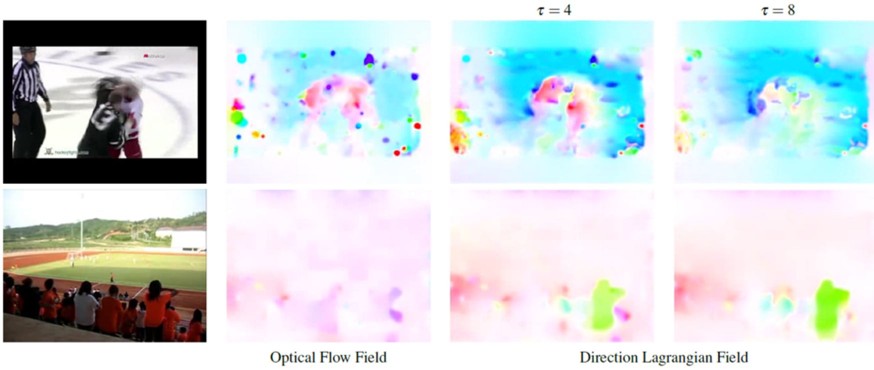

Figure 2.19: Optical flow and direction Lagrangian measures for different temporal scales applied to a fighting (top) and dancing crowd (bottom) sequence, method [20] (taken from [20]).

temporal domain are detected. Then normalized combinational vectors, called HNF, are computed around the detected STIP to characterize the video. After that, the Laplacian sparse representation framework and maximum pooling method are utilized, see the main framework in Figure 2.20. Finally, the SVM classifier is used to classify each feature vector as normal or abnormal action. The method is tested on the public *Violent-Flows* and *Hockey* datasets. The method achieves an accuracy of 82% and 95.3% on these datasets, respectively, see Tables 2.5 and 2.6.

2.2.3 Optical Flow

Chen et al. [54] propose a new feature called Optical Flow Context Histogram (OFCH) for detecting abnormal events, especially the fighting or violence events from a live camera stream. The OFCH is a log-polar histogram which combines the histogram of orientation and magnitude of optical flow. Then, 400 optical feature points are extracted. A human action is represented by using this histogram sequence. Principal Component Analysis (PCA) is adopted to reduce the dimension of the representation. RF, SVM and Bayesnet classifiers are employed. To verify the proposed method, different video clips from the Internet are downloaded. 50 fight and 50 non-fight

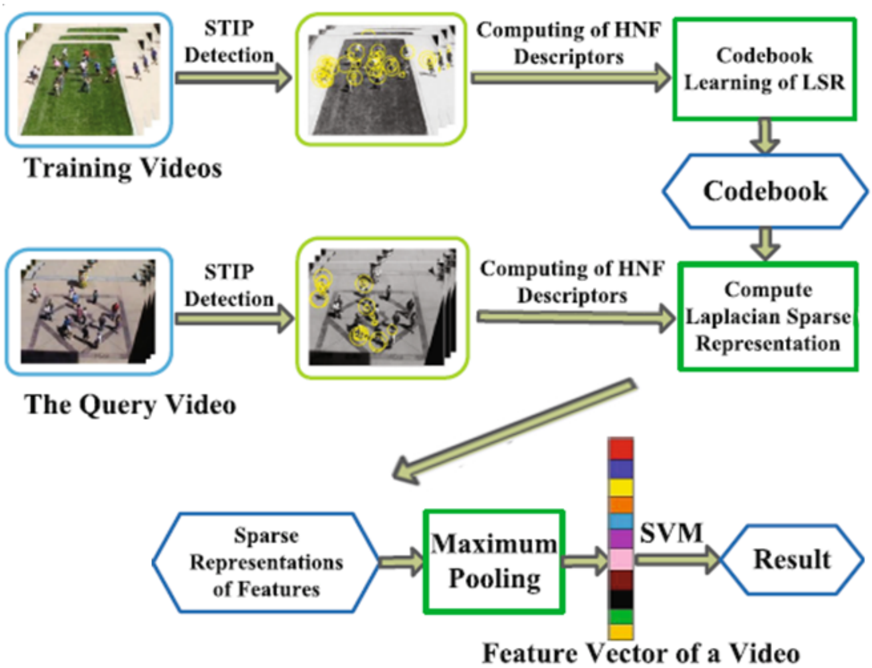

Figure 2.20: Overview of method [21] (taken from [21])

video clips are used for the experiments. The RF classifier achieves the highest accuracy with 96% on their dataset.

Hassner et al. [9] describe a novel approach to real-time detection of breaking violence in crowded scenarios. The method considers statistics of how flow-vector magnitudes change over time. These statistics, collected for short frame sequences, are represented using ViF descriptor. Firstly, optical flow between pairs of consecutive frames is obtained. The method also considers these optical flow magnitudes themselves in order to estimate how they change over time. ViF descriptors are used for classification in two different manners: (1) As global descriptor, extracted for a frame sequence as a whole or (2) as proxies used to produce a BoW representation for each sequence. Then, a SVM classifier with linear kernel is used. The proposed method is tested on the public *Violent-Flows*, *Hockey* and *ASLAN* datasets. The method gives an accuracy of 81.3% and 82.9% on *Violent-Flows* and *Hockey* datasets, respectively, see Tables 2.5 and 2.6.

Huang et al. [55] present a method for detection of violent crowd behavior. A statistical method based on optical flow field is introduced. The method considers the statistical characteristics of optical flow field and extracts a statistic of the optical flow (SCOF) descriptor. The SCOF descriptors are then categorized as either normal or violent actions using a SVM classifier with linear kernel. The proposed method is tested on the public *Hockey* and *Violent-Flows* datasets. The method achieves an accuracy of 83.4% and 86.4% on these datasets, respectively, see Tables 2.5 and 2.6.

Zhang et al. [22] propose a fast and robust framework for detecting and localizing violence in surveillance scenes. For this purpose, a Gaussian Model of Optical Flow (GMOF) is proposed to extract candidate violence regions, which are adaptively modeled as a deviation from the normal behavior of the crowd observed in the scene. Violence detection is then attempted on each video volume constructed by densely sampling candidate violence regions. To distinguish violent events from nonviolent events, a novel descriptor called Orientation Histogram of Optical Flow (OHOF) is proposed, which feeds a linear SVM classifier, see the overview in Figure 2.21. Finally, the proposed method is tested on the public *BEHAVE*, *CAVIAR* and *Violent-Flows* datasets. The method achieves an accuracy of 85.3%, 86.8% and 82.8% on these datasets, respectively, see Tables 2.5 and 2.6.

Zizi et al. [56] propose an approach to figure out human aggressive movements using the Horn-Schunk optical flow algorithm. The idea here is based on finding an amount of difference in two consecutive frames of a video sequence. This approach is suitable for simple motion detection only, it cannot highlight a specific region of moving objects. A private dataset is created with two low resolution videos captured by a static camera and used to validate the method.

Fu et al. [57] present a method for fight detection in videos. To extract more detailed motion information from optical flow images, a heuristic framework is proposed to classify motion regions, with

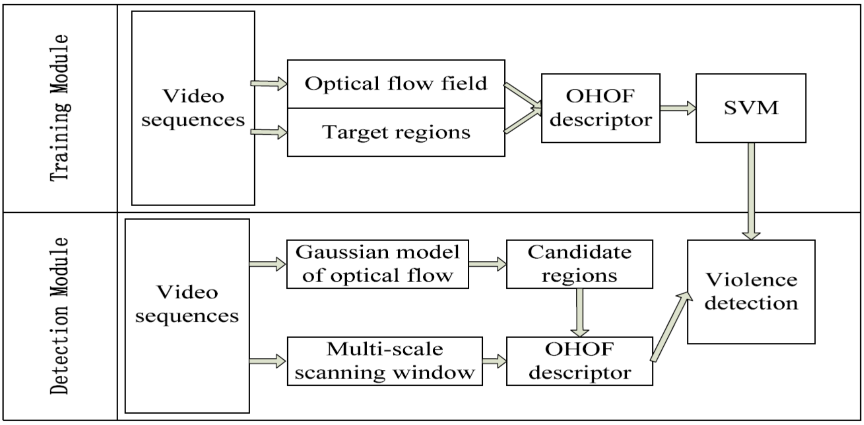

Figure 2.21: Overview of the OHOF method [22] (taken from [22])

respect to their size, count, and motion direction. The concept of motion attraction to measure the strength between two motion regions is also proposed. Finally, the proposed method is tested on the public *Hockey* and *Violent-Flows* datasets. The method gives an accuracy of 90.9% and 89.5% on these datasets, respectively, see Tables 2.5 and 2.6.

Gao et al. [23] develop a novel feature extraction method called OViF, which takes full advantage of the motion magnitude change information in statistical motion orientations based on the aforementioned ViF method [9]. ViF is a feature descriptor that can describe the changes of observed motion magnitudes well. However, in some cases, it may lose important information. For example, when the flow vectors of the same pixel in two consecutive frames have the same magnitude but only differ in their directions, the effect of ViF is limited. This is because ViF considers that there is no difference between these two flow vectors although actually they differ. Therefore, the new feature representation method OViF is proposed, which considers both motion magnitudes and motion orientations. The OViF approach for a video sequence is illustrated in Figure 2.22. Feature combination and multi-classifier combination strategies are adopted. The proposed method is tested on the public *Movies* and *Violent-*

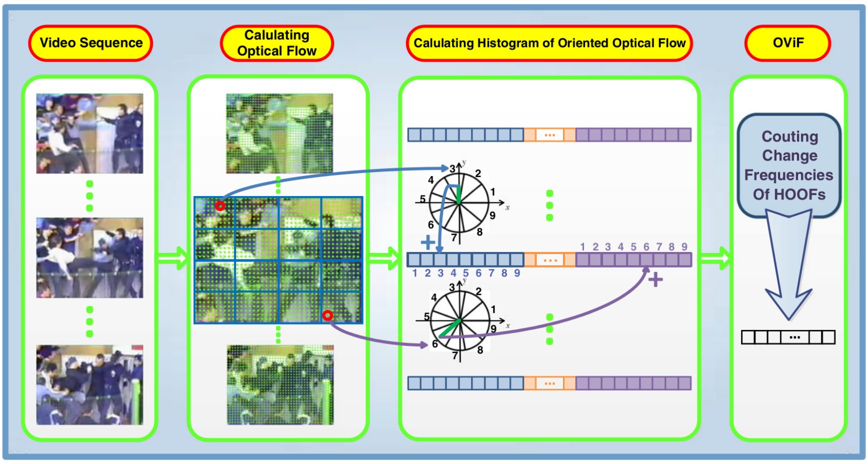

Figure 2.22: The OViF descriptor [23] for a video sequence (taken from [23])

Flows datasets. The method achieves an accuracy of 87.5% and 88% on these datasets, respectively, see Tables 2.5 and 2.6.

Zulkifley et al. [24] develop a Kalman filter-based aggressive behavior detection system. The developed system uses average optical flow velocity as an input to a bi-state tracker which represents normal or aggressive behavior, see the general diagram in Figure 2.23. A private dataset is created with five low resolution videos captured by a standard web camera to validate the method. The best accuracy obtained is 75%.

2.2.4 Trajectories

Datta et al. [58] define an Acceleration Measure Vector (AMV) descriptor that is composed with direction and magnitude of motion and jerk. In order to detect violence, the method relies on motion trajectory information and orientation information of a person's limb. AMV is calculated using the third derivative of its speed to infer jerk. During violent actions, the motion trajectory of a person experiences a drastic change after being hit by the other person and jerk is an effective way to capture this behavior. AMV encodes the phenomenon that if the person has been moving in a direction for the last frames

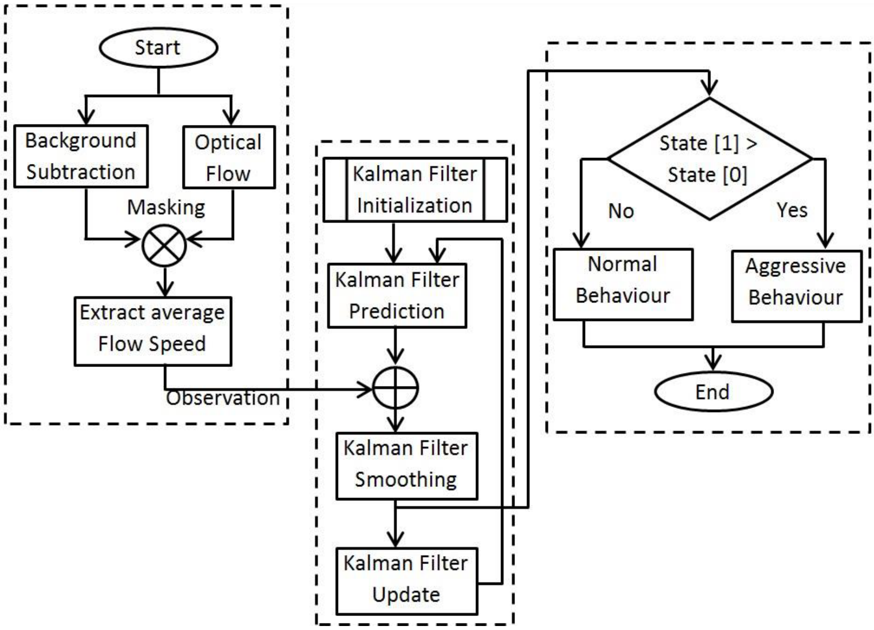

Figure 2.23: Flowchart of the aggressive behaviour detection method [24] (taken from [24])

and then motion suddenly changes direction and magnitude, then this person is a good candidate for having been hit. A private dataset is created with 15 sequences (roughly, 500 frames per sequence) with eight different people performing various violent and non-violent activities. The method achieves an accuracy of 95.8% on their private dataset.

Wang et al. [25] present an approach for detecting violence in videos, where Discriminative Slow Feature Analysis (SFA) is introduced to learn slow feature functions from dense trajectories derived from videos. SFA is an unsupervised learning technique to extract slowly varying features from a quickly varying input image. With the learned slow feature functions, the Accumulated Squared Derivative (ASD) features are extracted to represent videos. Finally, a linear SVM is trained for classification, see the general diagram in Figure 2.24. A private dataset is created with 200 violence videos and

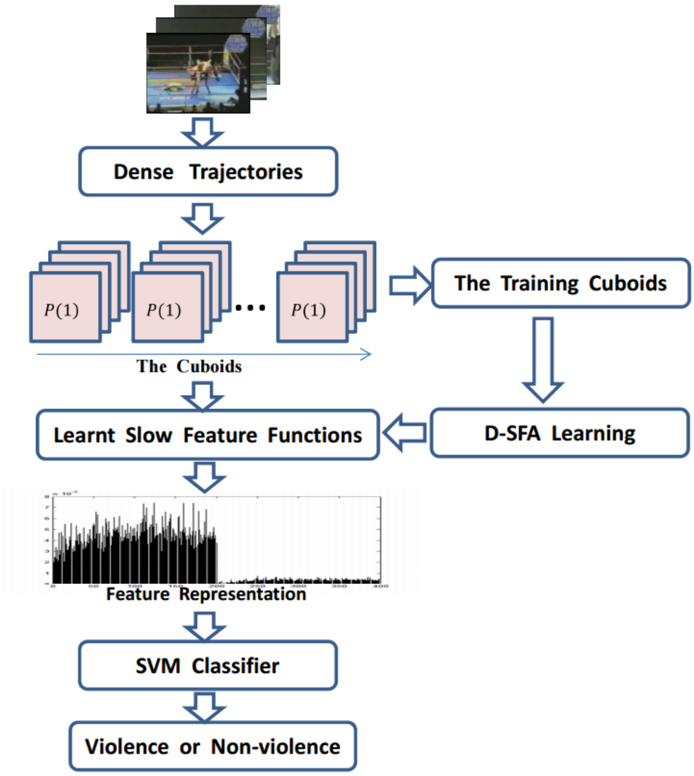

Figure 2.24: Overview of method [25] (taken from [25])

200 non-violence videos from movies and the Internet. The approach achieves an accuracy of 93.6% on this private dataset.

Khokher et al. [26] present a new method for violent scene detection using super descriptor tensor decomposition. Audio and visual local features are extracted from the video segments using Mel Frequency Cepstral Coefficient (MFCC), HoG, Motion Boundary Histograms (MBH)x and MBHy descriptors. The method refines the dense trajectories method by excluding extra trajectories and incorporating only those that are present in the region of interest. Feature descriptors are encoded through the super descriptor vector method. The encoded features are represented as tensors in order to retain the

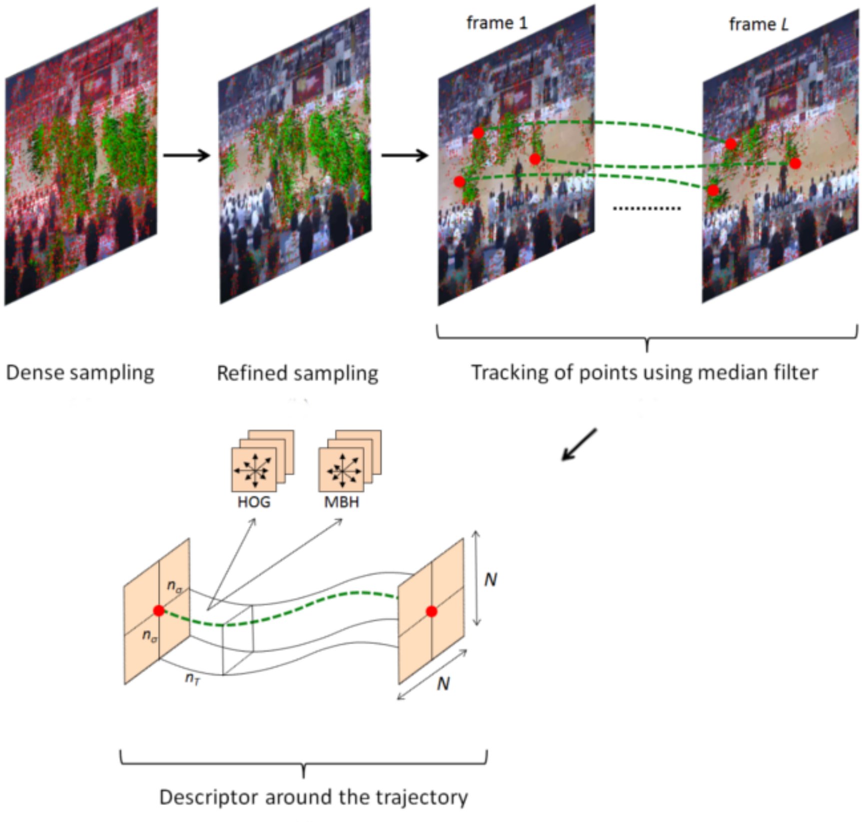

Figure 2.25: Flowchart example of the trajectory method [26] (taken from [26])

interactions between features. The number of features are significantly reduced through TUCKER-3 tensor decomposition and Fisher score based selection. This provides a way to extract the discriminant features required for the classification, in addition to dimensionality reduction. Finally, a linear SVM is used to recognize the violent and non-violent video segments, see an example sequence of the trajectory method in Figure 2.25. A public violence dataset is used with the proposed method: the *VSD*. The result is shown and compared in Table 2.4.

Rota et al. [59] propose a method to detect and localize dynamic human interactions in real videos, focusing on fight detection. In order to capture both shape and motion features, dense trajectories are used. The extraction of the dense trajectories is done with method [60]. Interest points are sampled using Shi and Tomasi algorithm and then tracked using Farneback's implementation of optical flow. The mean coordinates of these points in each trajectory are considered as the location point for the extracted feature. Descriptors used in this method are essentially shape features, i.e. zero-order and first-order motion features as HoG and MBH. Afterwards, these features are clustered using a BoW approach. Then a SVM classifier with linear kernel is used. The proposed method is tested on the public *UT* and *ReDID* datasets. The *ReDID* dataset contains 30 video sequences (between 20 to 242 seconds), where there are 73 different fight instances. The proposed method achieves 76.18% accuracy on the *ReDID* dataset. The method gives an accuracy of 92.3% on the *UT* dataset, see Tables 2.5 and 2.6.

Serrano et al. propose the Spatio-Temporal Elastic Cuboid (STEC) Trajectories approach for fight detection in video, explained in detail in chapter 5. This method is based on the use of blob movements to create trajectories that capture and model the different motions that are specific to a fight. The proposed method is tested on the public *UT*, *Movies* and *Hockey* datasets. The method achieves an accuracy of 99%, 98% and 82.6% on these datasets, respectively, see Tables 2.5 and 2.6.

2.2.5 Deep Learning

Ding et al. [27] develop a novel 3D-CNN for violence detection in video without using any prior knowledge about discriminant features. They extend CNN to 3D, that is, a convolution on the temporal image sequences is applied, see the 3D-CNN architecture in Figure 2.26, where there are 9 layers including input and output. The 3D input on the network is a small video clip that consists of a sequence of contiguous images, rescaled to 60×90. The output layer has 1 dimension that can be 1 or 0 if violence is detected in the video clip or not, respectively. The proposed method is tested on the public *Hockey*

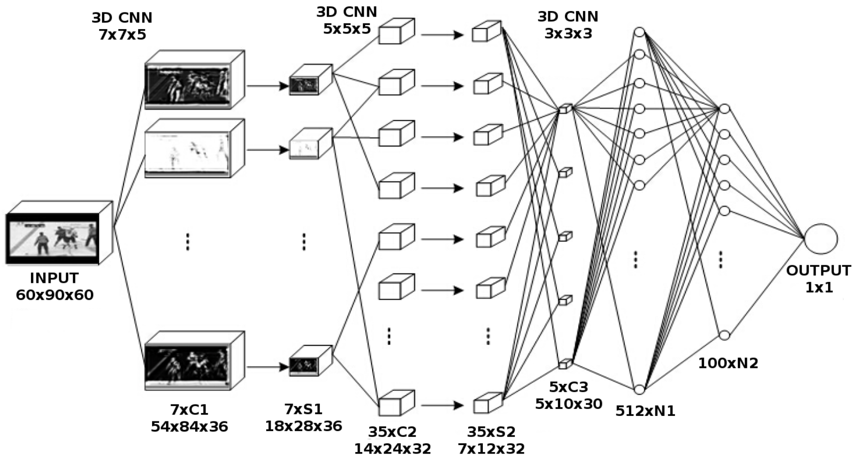

Figure 2.26: 3D-CNN architecture for violence detection [27] (taken from [27])

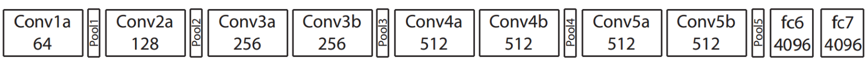

Figure 2.27: C3D architecture [28].

dataset. The method achieves an accuracy of 91% on this dataset, see Tables 2.5 and 2.6.

Tran et al. [28] also propose spatio-temporal feature learning using 3-dimensional convolutional neuronal networks (3D-CNN) for action recognition, called C3D. The authors argue that 3D-CNN is able to model temporal information thanks to the 3D pooling and 3D convolution layers. All 3D convolution kernels are set to 3 $\times$ 3 $\times$ 3 with a stride 1 in both spatial and temporal dimensions. The C3D method uses the architecture shown in Figure 2.27. The method achieves 85.2% action recognition accuracy on the challenging UCF101 dataset. The proposed method is tested on the public *Movies* and *Hockey* datasets. The method achieves an accuracy of 93.6% and 87.4% on these datasets, respectively, see Tables 2.5 and 2.6.

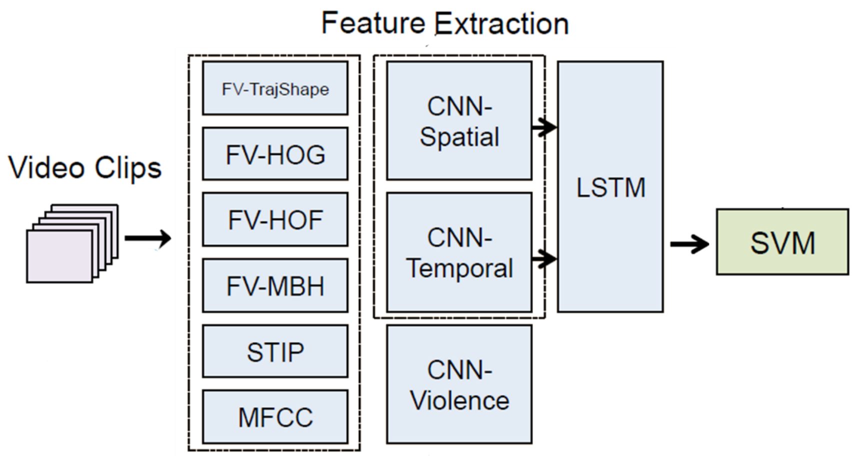

Figure 2.28: The key components of method [29] (taken from [29])

Dai et al. [29] propose a system that consists of several deep learning features. First, a CNN model is trained with a subset of ImageNet classes specifically selected for violence detection. An AlexNet [61] based model on video frames, which takes individual frames as network inputs followed by several convolutional layers, pooling layers and fully connected layers is trained. Second, a specially designed two-stream CNN framework is adopted to extract features on both static frames and optical flow images. Third, Long Short-Term Memory (LSTM) models are applied on top of the two-stream CNN features, which can capture longer-term temporal dynamics. In addition, several conventional motion and audio features are also extracted as complementary information to the deep learning features, see Figure 2.28 where the key components of the method are shown. The method is tested on the public *VSD* dataset. The result is shown and compared in Table 2.4.

Serrano et al. propose a hybrid approach for detecting fights, explained in detail in chapter 6. The method utilizes a hybrid framework by using both handcrafted and CNN learned features to recognize fights in videos. Handcrafted features are extracted to generate a temporal-voting model of action distribution by applying the Hough

Forest. A representative image for the video sequence is created by combining the temporal-voting model with the input sequence. A 2D CNN is applied to extract features from the representative image of the video and gives a final classification result. The proposed method is tested on the public *Movies* and *Hockey* datasets. The method achieves an accuracy of 99% and 94.5% on these datasets, respectively, see Tables 2.5 and 2.6.

2.2.6 Use of Audio

This section reviews state-of-the-art methods that use or add audio descriptors to detect violent actions. All but one method in this section have been already referenced in Section 2.2. Here we focus on the use of audio features.

In Nam et al. [38], an audio energy entropy is computed by dividing each (audio) frame into segments of K samples each. Signal energy is computed over each of these segments and normalized by the overall frame energy. The method achieves an accuracy of 88% on their dataset, using both audio and visual features.

Giannakopoulos et al. [62] propose a method to classify violent content using only audio features. Some popular frame-level audio features both from the time and frequency domain are employed. Afterwards, several statistics of the calculated feature sequences are fed as input to a SVM classifier (with linear, Gaussian radial basis, polynomial and sigmoid hyperbolic tangent kernels), which decides about the segment content. A private database of audio clips is extracted from several movies. The total duration of the samples is 20 minutes. Violent audio clips are extracted from scenes such as shots, explosions, fights and screams, while non-violent audio segments of music and speech are also extracted, along with non-violent sounds like fireworks, which have similar characteristics to violent sounds. The method achieves a recall of 90.5% on their dataset, using only audio features. However, the dataset is not representative. The total duration is relatively short. Besides, it does not consider silent violence actions, such as grabbing, pushing, pulling, etc.

Zajdel et al. algorithm [10] also performs audio processing in the time-frequency domain with an approach common in auditory scene

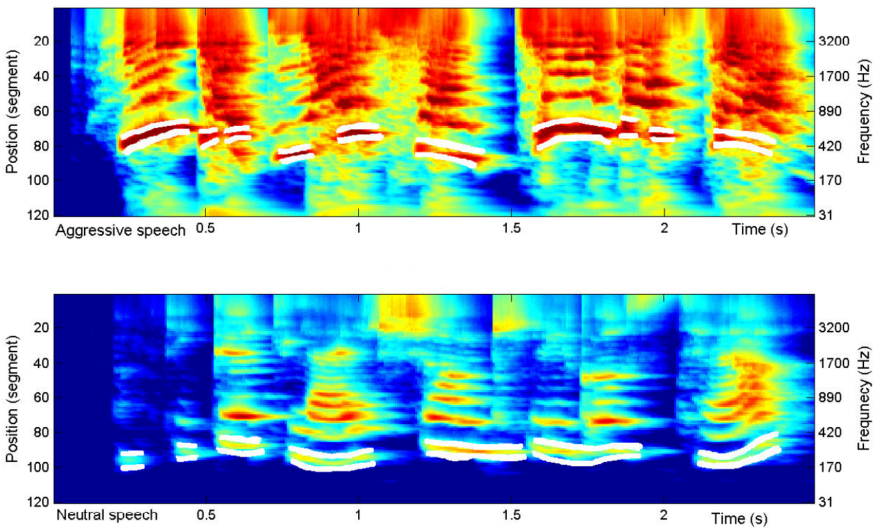

Figure 2.29: Typical cochleograms of aggressive and normal speech [10] (taken from [10])

analysis. The transformation of the time-signal to the time frequency domain is performed with a model of the human ear. This model is a transmission-line model with its channels tuned according to the oscillatory properties of the basilar membrane. Leaky-integration of the squared membrane displacement results in an energy-spectrum, called cochleogram, see Figure 2.29. The method achieves an accuracy around 78% using both audio and visual features.

Giannakopoulos et al. method [41] also calculates 12 audio features for each frame, different statistics with 20 msec window sizes such as spectogram, chroma 1 and 2, energy entropy, etc. Classification recall was 40.7% using only visual features. The use of audio features increased recall to 47%.

Giannakopoulos et al. [12] extract seven features based on the audio stream. To this end, the audio stream is mid-term segmented and a multi-class classification algorithm [63] is applied on each segment. Each mid-term window is 2-second long and there is a 50% overlap between successive windows. The resulting sequence of classification decisions is used to extract seven features. Each feature measures

Method	Year	Appr.	Recall	F1	Prec.
Gninkoun [42]	2011	S+A	77.4	28.9	11.8
Ionescu [13]	2012	S+A	54.4	49.9	46.1
Serrano [45]	2013	S	62.7	19.8	11.8
Khokher [26]	2014	S+A+T	-	60.2	-
Dai [29]	2015	DL	-	40.1	-

Table 2.4: Results obtained with the *VSD* dataset. Approach (Appr.): S = spatial descriptors, ST = spatio-temporal descriptors, OF = optical flow, T = trajectories, DL = deep learning and A = use of audio.

the percentage of mid-term segments that have been classified to the respective audio class. Seven audio classes are defined, the last 3 of which refer to violent content.

2.3 Results

In this section, different methods are compared using its classification accuracy and standard deviation. Only the methods that have been tested on important public violence datasets are compared.

Table 2.4 shows the methods that were tested on the *VSD* dataset. Appr. refers to the basic methodology used: local descriptors (L), spatio-temporal descriptors (ST), optical flow (OF), trajectories (T), deep learning (DL), use of audio (A) and others (O). The recall, F1 and precision measures are defined in Appendix A.

Tables 2.5 and 2.6 show the methods that use the other prominent datasets comparing classification accuracy (defined in Appendix A) and standard deviation.

2.4 Discussion

In this survey chapter, a detailed review of state-of-the-art violence datasets is carried out. A dataset division is presented and these datasets are compared. Eight public datasets have been described. A

Method	Year	Appr.	CAVIAR	BEHAVE	UT	UCF101
Bermejo [36]	2011	ST	-	-	-	-
Hassner [9]	2012	OF	-	-	-	-
Esen [15]	2013	ST	**89.4±-**	83.7±-	-	-
STIP [46]	2014	ST	-	-	-	72.5±0.7
MoSIFT [46]	2014	ST	-	-	-	81.3±1.0
ViF [46]	2014	OF	-	-	-	77.7±2.2
LMP [46]	2014	S	-	-	-	65.9±1.5
[46], chapter 4	2014	S	-	-	-	**93.4±6.1**
Huang [55]	2014	OF	-	-	-	-
Ding [27]	2014	DL	-	-	-	-
Xu [16]	2014	ST	-	-	-	-
[48], chapter 3	2015	S	-	-	-	79.5±0.9
Zhang [22]	2015	OF	86.8±0.2	85.3±0.2	-	-
Tran [28]	2015	DL	-	-	-	-
Fu [57]	2015	OF	-	-	-	-
Mohammadi [18]	2015	ST	-	**94.8±5.4**	-	-
Zhang [19]	2015	ST	-	87.2±0.1	-	-
Alazrai [47]	2015	S	-	-	85.0±-	-
Rota [59]	2015	OF+T	-	-	92.3±-	-
Senst [20]	2015	ST	-	-	-	-
Zhao [21]	2015	ST	-	-	-	-
Gao [23]	2016	OF	-	-	-	-
Lloyd [49]	2016	S	-	-	-	-
Serrano, chapter 5	2016	T	-	-	**99±-**	-
Serrano, chapter 6	2016	S+DL	-	-	-	-

Table 2.5: Accuracy and standard deviation obtained for each method tested on the first four public datasets. Approach (Appr.): S = spatial descriptors, ST = spatio-temporal descriptors, OF = optical flow, T = trajectories, DL = deep learning and A = use of audio.

Method	Year	Appr.	Hockey	Movies	Violent-Flows
Bermejo [36]	2011	ST	91.7±-	89.5±-	-
Hassner [9]	2012	OF	82.9±0.1	-	81.3±0.2
Esen [15]	2013	ST	-	-	-
STIP [46]	2014	ST	88.5±0.2	82.3±0.9	-
MoSIFT [46]	2014	ST	83.9±0.6	63.4±1.6	-
ViF [46]	2014	OF	82.3±0.2	96.7±0.3	-
LMP [46]	2014	S	75.9±0.3	84.4±0.8	-
[46], chapter 4	2014	S	90.1±0	85.4±9.3	-
Huang [55]	2014	OF	83.4±-	-	86.4±-
Ding [27]	2014	DL	91.0±-	-	-
Xu [16]	2014	ST	94.3±1.7	-	89.1±3.3
[48], chapter 3	2015	S	82.4±0.6	97.8±0.4	-
Zhang [22]	2015	OF	-	-	82.8±0.2
Tran [28]	2015	DL	87.4±1.2	93.6±0.8	-
Fu [57]	2015	OF	90.9±-	89.5±-	-
Mohammadi [18]	2015	ST	-	96.9±0.2	85.4±0.2
Zhang [19]	2015	ST	94.9±1.7	-	89.8±0.1
Alazrai [47]	2015	S	-	-	-
Rota [59]	2015	OF+T	-	-	-
Senst [20]	2015	ST	94.0±1.9	-	**92.4±6.7**
Zhao [21]	2015	ST	**95.3±-**	-	≈82±-
Gao [23]	2016	OF	-	87.5±1.7	88.0±2.5
Lloyd [49]	2016	S	-	-	91.3±-
Serrano, chapter 5	2016	T	82.6±-	98±-	-
Serrano, chapter 6	2016	S+DL	94.5±0.7	**99±0.6**	-

Table 2.6: Continuation. Accuracy and standard deviation obtained for each method tested on the remaining public datasets. Approach (Appr.): S = spatial descriptors, ST = spatio-temporal descriptors, OF = optical flow, T = trajectories, DL = deep learning and A = use of audio.

dataset ranking according to the estimated number of papers which directly reference each dataset is also shown.

On the other hand, a review of state-of-the-art methods for violence detection is carried out. These methods are clustered into the following approaches: spatial and spatio-temporal descriptors, optical flow, trajectories, deep learning, use of audio and others. Finally, state-of-the-art methods that use public datasets are compared by year and classification accuracy.

CHAPTER 3

Method 1: Motion Blob Characterization

This chapter describes a novel method for fight detection. Blobs of movement are first detected and then different features are used to characterize the blobs. The proposed method makes no assumptions on number of individuals (it can be also used to detect vandalism), body part detection or salient point tracking. Features extracted from motion blobs are used to discriminate fight and non-fight sequences. Although the method is outperformed in accuracy by other state-of-the-art methods, it is significantly faster, thus making it amenable for real-time applications.

This chapter is organized as follows. Section 3.1 describes the proposed method. Section 3.2 provides a comparison of the results with other related state-of-the-art methods. Finally, in Section 3.3 the main conclusions are outlined.

3.1 Motion Blob Characterization Method

The main steps of the proposed method are shown in Figure 3.1. It is hypothesized that motion blobs in fight sequences have a distinct position and shape. Firstly, the absolute image difference between consecutive frames is computed. The resulting image is then binarized, leading to a number of motion blobs, see Figure 3.2 where the largest blobs have been marked on a fight sequence and Figure 3.3 on a non-fight sequence. Only the K largest motion blobs are selected for further processing.

In order to characterize the K blobs, different measures are computed such as area, centroid, perimeter, etc. as well as distances between blob centroids. In the following, the method is described in detail.

A short sequence $S(s)$ of gray scale images is denoted as:

$$S(s) = I_t(x, y) \tag{3.1}$$

where $x = 1, 2, ..., N$, $y = 1, 2, ..., M$ and $t = 1, 2, ..., T$. N and M are the number of rows and columns of each frame I and T is the number of frames respectively.

Let $I_{t-1}(x, y)$ and $I_t(x, y)$ be two consecutive frames in the sequence. The absolute difference between consecutive images is then computed as:

$$E_t(x, y) = |I_{t-1}(x, y) - I_t(x, y)| \tag{3.2}$$

Then, E is binarized using a threshold h:

$$F_t(x, y) = \begin{cases} 1, & \text{if } E_t(x, y) > 255 \cdot h, \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

where $0 < h < 1$ can be arbitrarily chosen. The second step is to define each blob on the image $F_i(x, y)$. Each blob (B) is the set of pixels (x, y) for which $B_{b,t}(x, y) = b$, where $b = 0, 1, 2, ..., J$ and where J is the number of blobs in image $F_t(x, y)$. Each blob $B_{b,t}(x, y)$ is here defined as an image containing a set of adjacent points, neighborhood ($\mathcal{N}$), where $F_t(x, y) = 1$.

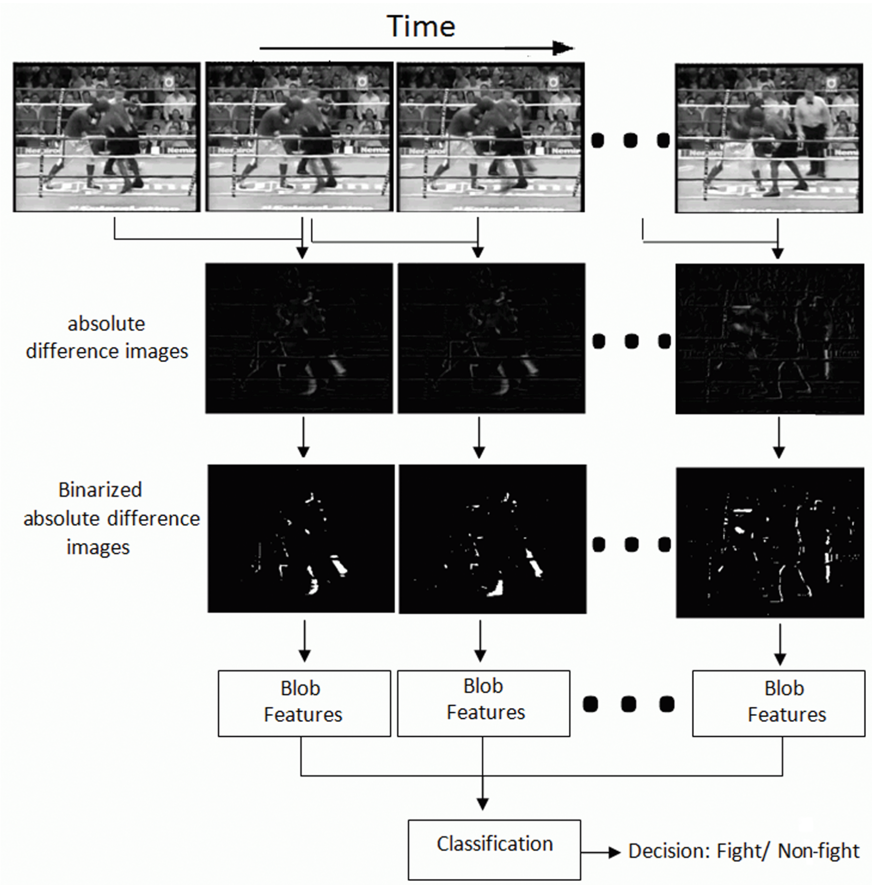

Figure 3.1: General diagram of the proposed method.

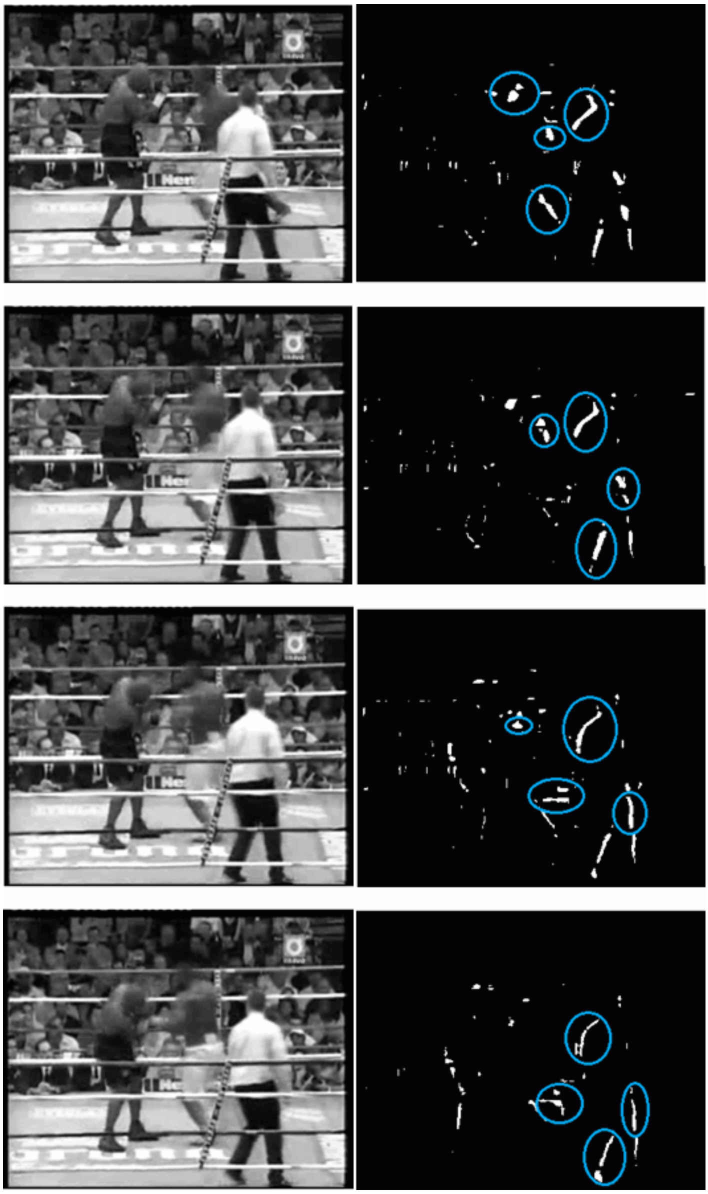

Figure 3.2: Four consecutive frames from a fight sequence where the four largest motion blobs have been marked.

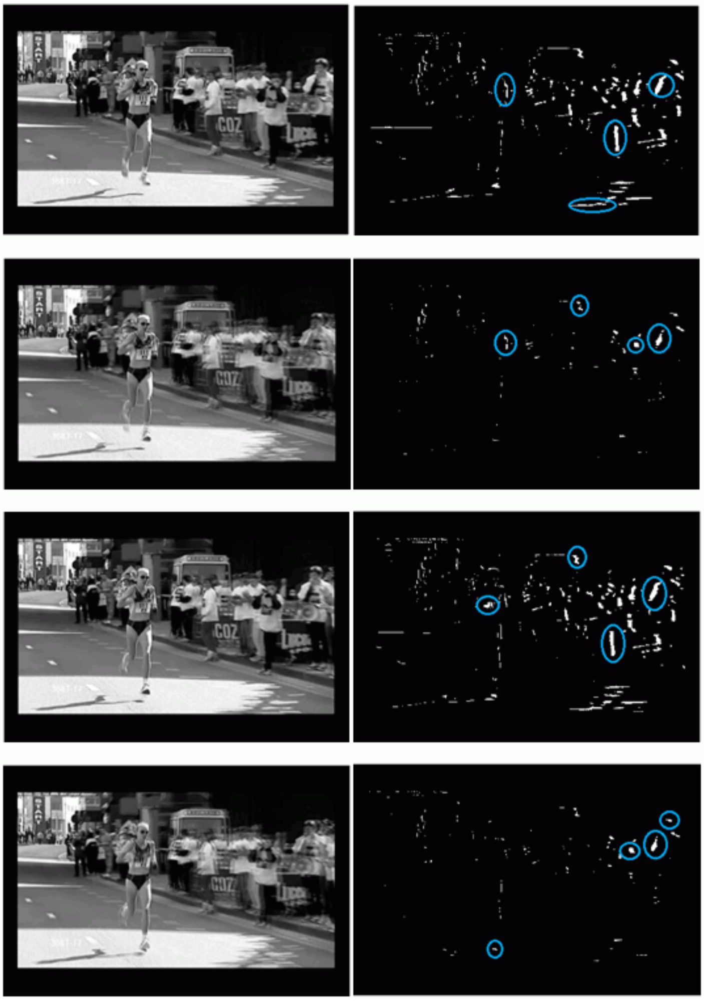

Figure 3.3: Four consecutive frames from of a non-fight sequence where the four largest motion blobs have been marked. Compare with those of Figure 3.2. In the previous figure the motion blobs are larger and clustered, whereas in the current figure the motion blobs are smaller and not clustered.

$$B_{b,t}(x,y) = \begin{cases} 1, & \text{if} \bigcup_{(x,y)\in\mathcal{N}_1} F_t(x,y) = 1, \sum((x,y)\in\mathcal{N}_1) = m_1, \\ 2, & \text{if} \bigcup_{(x,y)\in\mathcal{N}_2} F_t(x,y) = 1, \sum((x,y)\in\mathcal{N}_2) = m_2, \\ ... \\ J, & \text{if} \bigcup_{(x,y)\in\mathcal{N}_J} F_t(x,y) = 1, \sum((x,y)\in\mathcal{N}_J) = m_J, \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

where m_1, m_2, ... m_J are the number of adjacent pixels of each blob respectively and $m_1 + m_2 + ...m_J \leq N \cdot M$. These blobs are calculated for every frame. Afterwards, the K largest blobs are selected to extract useful information for the classification of the video sequences. The following features are extracted by the proposed method.

The blob areas ($A_{a,t}$) will be defined as:

$$A_{a,t} = \frac{\sum_{x=1}^{N}\sum_{y=1}^{M} B_{a,t}(x,y)}{a} \tag{3.5}$$

where $a = 1, 2, ..., K$. The centroids ($CX_{c,t}$ and $CY_{c,t}$) are

$$CX_{c,t} = \frac{\sum_{x=1}^{N} B_{c,t}(x,y)\cdot x}{\sum_{x=1}^{N} B_{c,t}(x,y)} = \frac{\sum_{x=1}^{N} B_{c,t}(x,y)\cdot x}{N} \tag{3.6}$$

$$CY_{c,t} = \frac{\sum_{y=1}^{M} B_{c,t}(x,y)\cdot y}{\sum_{y=1}^{M} B_{c,t}(x,y)} = \frac{\sum_{y=1}^{M} B_{c,t}(x,y)\cdot y}{M} \tag{3.7}$$

where $c = 1, 2, ..., K$. The best K blobs are selected according to area (largest K areas). Now, the distances $D_{d,t}$ between blobs are calculated as:

$$D_{d,t} = \sqrt{(CX_{d+1,t} - CX_{d,t})^2 + (CY_{d+1,t} - CY_{d,t})^2} \tag{3.8}$$

where $d = 1, 2, ..., \sum_{i=1}^{K-1} i$. Therefore, $D_{d,t}$ represents the distances between every pair in the K blobs.

Compactness ($C_{co,t}$) is also used to estimate the shape (rounded or elliptical) of the blob and is defined as:

$$C_{co,t} = \frac{P_{co,t}^2}{A_{co,t}} \tag{3.9}$$

where $co = 1, 2, ..., K$. To estimate the perimeter ($P_{p,t}$) the Sobel operator [64] has been used to detect the edges over $B_{b,t}(x, y)$. This operator ($G_{b,t}(x, y)$) is applied to calculate the edges for each blob. Edge pixels have value 1 and the rest 0.

$$P_{p,t} = \frac{\sum_{x=1}^{N} \sum_{y=1}^{M} G_{p,t}(x, y)}{p} \tag{3.10}$$

where $p = 1, 2, ..., K$.

Finally, note that the proposed method is intrinsically able to learn in the presence of global motion. Global motion leads to blobs with a distinct shape and position, typically large elongated blobs, see Figure 3.4.

3.2 Experiments

The proposed method was assessed using three different datasets and compared with five other methods. The *Hockey*, *Movies* and *UCF101* datasets are used, which were explained in Section 2.1.

In the experiments with the *UCF101* dataset, for the fight set we pooled the fight clips of both the *Hockey* and *Movies* dataset plus two of the 101 UCF actions that actually represented fights ("Punching" and "Sumo", see Figure 3.5). This gave a total of 1843 fight clips and 42278 non-fight clips (totaling approximately 2 Million frames). This extended dataset is called *UCF101-E*. In order to avoid unbalanced sets we used randomly chosen subsets of 500 fight and 500 non-fight clips. For each subset we performed a 10-fold cross-validation. This in turn was repeated 10 times.

In the following experiments, two variants of the proposed method are described, called v1 and v2.

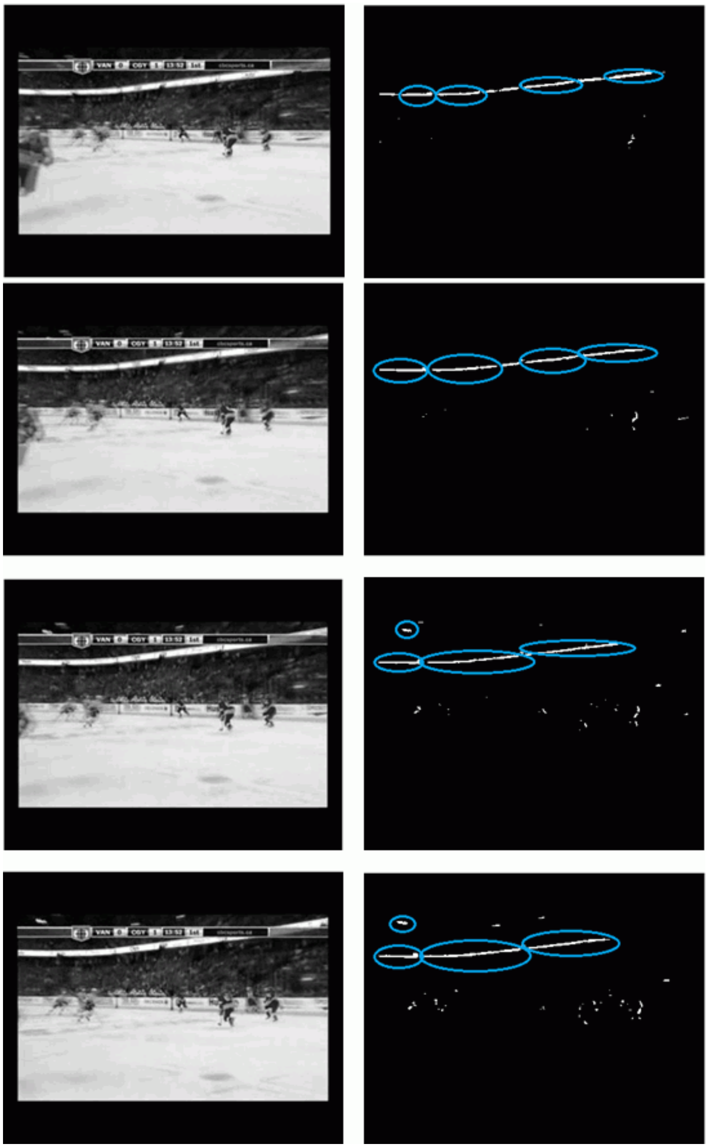

Figure 3.4: Four consecutive frames from with global movement where the four largest motion blobs have been marked.

Figure 3.5: Fight sequence on the original *UCF101* dataset. Sample frames of Sumo and Punch categories in the *UCF101* dataset.

As for the other related methods, the BoW framework was used with both STIP and MoSIFT features, as in [36]. In [36], STIP features performed poorly on the Movie dataset and so MoSIFT was considered the best descriptor. MoSIFT has been also proven in other action recognition works. For BoW+MoSIFT, and even with the use of parallel K-means, extracting vocabularies from the whole *UCF101-E* dataset was unfeasible. Therefore, a random subset of 1200 samples was first selected and then a vocabulary of size 500 (the best vocabulary size in [36]) was computed[1].

The ViF [9] mentioned in chapter 2.2 was also tested. Finally, results are also compared to the LMP [65]. This method is based on extracting simple statistics (variance, skewness and kurtosis) from temporal cuboids centered on tracked keypoints. Keypoints are located with a Harris detector. In [65] the authors claimed that this descriptor is both informative and efficient for action recognition. Since the number of extracted descriptor vectors varies in each video sequence we extracted fixed-size histograms for each feature, similar to what is done with the proposed method. The best results were obtained using 6 bins.

The first variant (v1) assumes that position and shape of K largest blobs can discriminate between fight and non-fight sequences. To

[1] The authors are not aware of any previous work using MoSIFT in a BoW framework with the *UCF101* dataset.

characterize position and shape, the centroid and distances between blob centroids are computed. In summary, this variant (v1) uses the best K areas, the K centroids of these areas and the distances between these centroids. Hence, a sequence S has $3K + K(K-1)/2$ features.

The second variant (v2) uses the following features to characterize position and shape, the centroid, distances between blob centroids and compactnesses. Finally, this variant (v2) uses the best K areas, the K centroids of these areas, the distances between these centroids and the K compactnesses. Hence, the sequence S has $2K + K(K-1)/2$ features.

In the experiments, the proposed method needs to set two parameters. First, a binary thresholding parameter h between $0 < h < 1$ to binarize this image. Second, the parameter K that represents the number of largest blobs (Areas, Centroids, distances between centroids and compactnesses. The *Movies* dataset has been used to fit these parameters. To find the best configuration of the proposed method Accuracy has been used. Tables 3.1 and 3.2 show the results measured using 5 repetitions of 10-fold cross-validation and three different classifiers KNN, AdaBoost (100 classifiers, weighted voting) and RF (50 trees) for both variants (v1 and v2) of the method. The aim is to set the best parameters h and K for these two variants. These tables (3.1 and 3.2) show the results. The h and K parameters have been set to 0.4 and 8 respectively for both variants. In the following, these two variants are compared with other extant methods.

The related methods are now compared with the two variants proposed, see Table 3.3 which shows the results for the three datasets using SVM, AdaBoost and RF classifiers are compared. See Figures 3.6, 3.7 and 3.8 where the ROC can be observed for the RF classifier for each method.

In the experiments above, the ViF and LMP implementations followed the original version in the respective papers. In order to evaluate the dependence of accuracy of these methods on the number of features, it was also deemed necessary to assess the accuracy vs speed trade-off offered by those two methods. Figure 3.9 shows the results obtained with ViF (using *UCF101-E* dataset). In this case the ViF

h	K	KNN	AdaBoost	RF
0.2	4	91.1 ± 0.7	79.5 ± 0.6	95.9 ± 0.5
	6	92.6 ± 0.5	84.9 ± 0.8	94 ± 0.6
	8	95 ± 0.4	87.7 ± 1.1	$\mathbf{96.4 \pm 0.5}$
	10	93.5 ± 0.2	89.2 ± 0.4	95.5 ± 0.4
	12	94.7 ± 0.4	90.6 ± 0.6	95.2 ± 0.5
	14	93.5 ± 0.2	90.2 ± 0.6	95.0 ± 0.6
	16	93.3 ± 0.8	89.4 ± 0.4	95.1 ± 0.5
	18	93 ± 0.4	88.1 ± 0.6	95.2 ± 0.8
	20	92.8 ± 0.4	87.2 ± 0.3	94.8 ± 0.4
0.05	8	88.6 ± 0.1	78.5 ± 0.7	93.8 ± 0.3
0.1		90.2 ± 1	78.4 ± 0.8	94.4 ± 0.7
0.15		88.5 ± 0.4	83.6 ± 0.5	96.1 ± 0.3
0.25		93.6 ± 0.7	88 ± 0.9	96.1 ± 0.3
0.3		94.8 ± 0.4	87.8 ± 0.6	97.4 ± 0.4
0.35		90.6 ± 1.9	85.1 ± 0.4	97.1 ± 0.3
0.4		63.4 ± 3.8	81.8 ± 0.2	$\mathbf{97.7 \pm 0.4}$
0.45		91.7 ± 1.1	79.6 ± 0.8	94.8 ± 0.4
0.5		94 ± 0.2	77.7 ± 0.3	92.7 ± 0.4

Table 3.1: This table shows the mean and standard deviation of the Accuracy for these three classifiers for variant 1 (v1) of the proposed method

h	K	KNN	AdaBoost	RF
0.2	4	92.7 ± 0.8	79.3 ± 0.8	95.9 ± 0.3
	6	93.2 ± 0.6	84.5 ± 1.5	95.7 ± 0.6
	8	95.2 ± 0.8	88.6 ± 0.6	$\mathbf{96.4 \pm 0.3}$
	10	93.7 ± 0.4	90.2 ± 0.4	95 ± 0.8
	12	94.5 ± 0.5	90.6 ± 0.7	95.9 ± 0.9
	14	93.9 ± 0.5	90.7 ± 0.4	95.1 ± 0.9
	16	94 ± 0.4	90 ± 0.2	95.5 ± 0.8
	18	93.7 ± 0.3	88.2 ± 0.5	95.2 ± 1
	20	92.6 ± 0.5	87.2 ± 0.3	94.4 ± 0.5
0.05	8	88.8 ± 1.2	77.3 ± 0.4	93.9 ± 0.5
0.1		91.2 ± 0.6	78.8 ± 1.4	94.9 ± 0.7
0.15		89.7 ± 1.3	83.1 ± 1.4	96.1 ± 0.5
0.25		93 ± 0.4	88.4 ± 1.2	95.8 ± 0.6
0.3		94.8 ± 0.4	88.4 ± 0.6	97.3 ± 0.2
0.35		88.2 ± 2	85.3 ± 0.6	97.4 ± 0.2
0.4		65.1 ± 2.7	81.8 ± 0.4	$\mathbf{97.8 \pm 0.4}$
0.45		91.2 ± 2.3	79.5 ± 0.5	94.9 ± 0.5
0.5		91 ± 4.3	77.2 ± 0.3	92.6 ± 0.3

Table 3.2: This table shows the mean and standard deviation of the Accuracy for three classifiers for variant 2 (v2) of the proposed method

Method	Classifier	Movies	Hockey	UCF101-E
BoW+ STIP	SVM	82.3±0.9/0.88	88.5±0.2/0.95	72.5±1.5/0.74
	AdaBoost	75.3±0.83/0.83	87.1±0.2/0.93	63.1±1.9/0.68
	RF	97.7±0.5/0.99	96.5±0.2/0.99	87.3±0.8/0.94
BoW+ MoSIFT	SVM	63.4±1.6/0.72	83.9±0.6/0.93	81.3± 1/0.86
	AdaBoost	65.3±2.1/0.72	86.9±1.6/0.96	52.8±3.6/0.62
	RF	75.1±1.6/0.81	**96.7±0.7/0.99**	86.3±0.8/0.93
ViF	SVM	96.7±0.3/0.98	82.3±0.2/0.91	77.7±2.16/0.87
	AdaBoost	92.8±0.4/0.97	82.2±0.4/0.91	78.4±1.7/0.86
	RF	88.9±1.2/0.97	82.4±0.6/0.9	77±1.2/0.85
LMP	SVM	84.4±0.8/0.92	75.9±0.3/0.84	65.9±1.5/0.74
	AdaBoost	81.5±2.1/0.86	76.5±0.9/0.82	67.1±1/0.71
	RF	92±1/0.96	77.7±0.6/0.85	71.4±1.6/0.78
Deniz[46]	SVM	85.4±9.3/0.74	90.1±0/0.95	**93.4±6.1/0.94**
	AdaBoost	**98.9±0.22/0.99**	90.1±0/0.90	92.8±6.2/0.94
	RF	90.4±3.1/0.99	61.5±6.8/0.96	64.8±15.9/0.93
Variant-v1	SVM	87.9±1/0.97	70.8±0.4/0.75	72.1±0.9/0.78
	AdaBoost	81.8±0.5/0.82	70.7±0.2/0.7	71.7±0.9/0.72
	RF	97.7±0.4/0.98	79.3±0.5/0.88	74.8±1.5/0.83
Variant-v2	SVM	87.2±0.7/0.97	72.5±0.5/0.76	71,2±0.7/0.78
	AdaBoost	81.7±0.2/0.82	71.7±0.3/0.72	71±0.8/0.72
	RF	97.8±0.4/0.97	82.4±0.6/0.9	79.5±0.9/0.85

Table 3.3: This table shows the results for the three datasets using SVM, AdaBoost and RF classifiers for the related methods and the two proposed variants. Three measures have been calculated: mean accuracy, standard deviation accuracy and AUC.

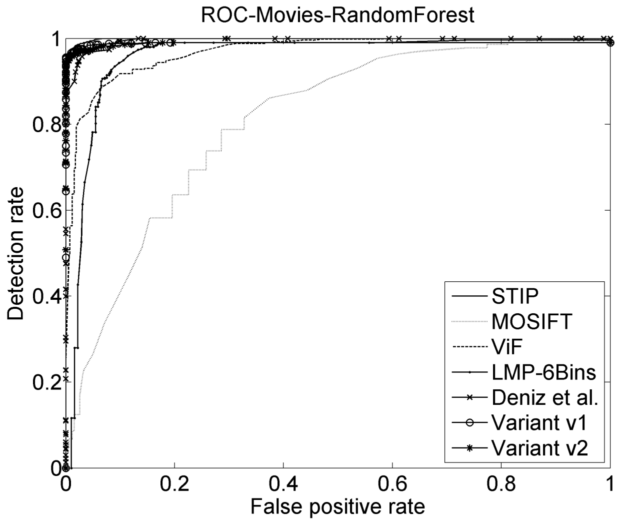

Figure 3.6: ROC curve for the five related methods and the two considered variants. The RF classifier on the *Movies* dataset is used.

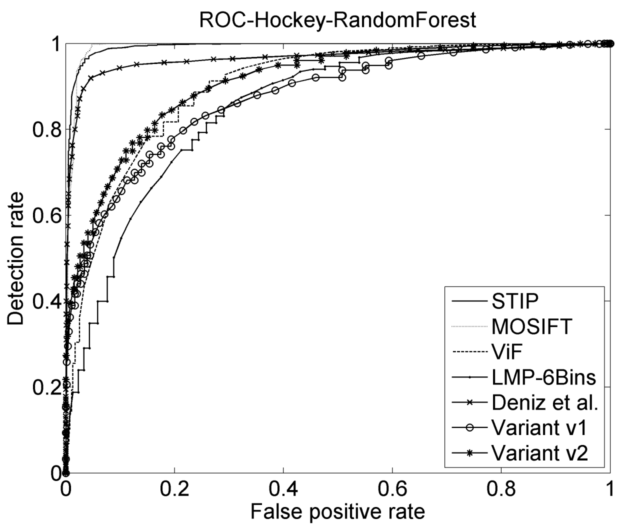

Figure 3.7: ROC curve for the five related methods and the two considered variants. The RF classifier on the *Hockey* dataset is used.

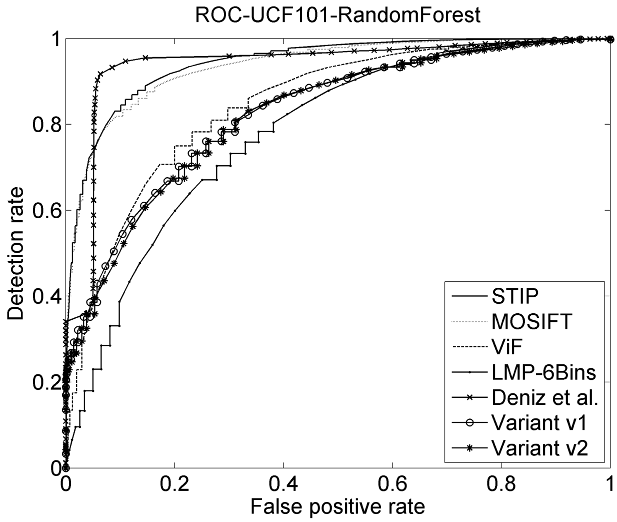

Figure 3.8: ROC curve for the five related methods and the two considered variants. The RF classifier on the *UCF101* dataset is used.

results were obtained by varying the width of the coarsest level of the optical flow computation. As for LMP, the results (see Figure 3.10) were obtained by varying the number of temporal scales considered by the algorithm.

Table 3.4 shows the number of features used for classification and the computational cost (for feature extraction). The code for both STIP and MoSIFT was compiled, while the code for rest of the methods was interpreted and used no parallelization. These results show a significant improvement in speed. The proposed method, using 2548 and 2940 features respectively, is computationally 12 and 16 times faster than the STIP and MoSIFT methods.

From the results it can be observed that the proposed method is not better than BoW+STIP, BoW+MoSIFT and Deniz et al. [46] in terms of accuracy. However, the proposed method gives slightly better accuracies than the ViF and LMP methods. Most importantly, see Table 3.4, the proposed method is approximately 26, 12, 18, 6 and 7 times faster than the BoW+STIP, BoW+MoSIFT, ViF, LMP and Deniz et al. [46] methods respectively.

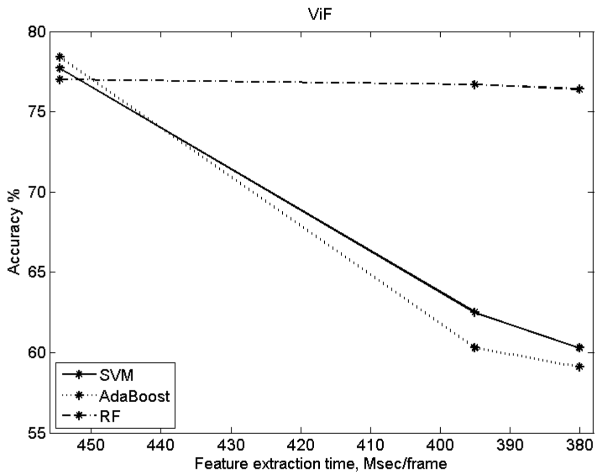

Figure 3.9: ViF results, comparing accuracy vs feature extraction time.

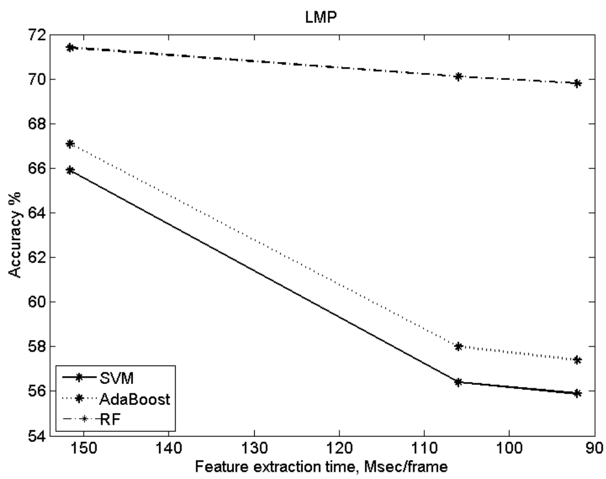

Figure 3.10: LMP results, comparing accuracy vs feature extraction time.

Method	Features/sequence	Msecs/frame
BoW+MoSIFT	500	661.5
BoW+STIP	500	293.5
ViF	96000	454.5
LMP	10368	151.6
Deniz [46]	14	162.4
Variant-v1	2548	22.5
Variant-v2	2940	26.5

Table 3.4: Feature extraction times. Average times measured with the non-fight videos in the *UCF101-E* dataset, on an Intel Xeon computer with 2 processors at 2.90Ghz.

The proposed method is amenable to real-time applications since it requires less than $1/25$ seconds per frame. In fact, the time spent in deciding whether an input sequence (50 frames) contains a violent action is between 0.0225 and 0.0265 seconds per frame.

The novel method has a processing average time of 0.02391 seconds per frame (between v1 and v2). The method has different substeps for each frame. Some of these sub-steps can be parallelized when the image can be splitted in different smaller sub-images without affecting the result. Each sub-image can be handled by a processor. These sub-steps are shown in Table 3.5.

Given a computer with N processors and parallelizable processes, the processing time can be theoretically estimated from Table 3.5 as $T = parallelizable\_processes/N + not\_parallelizable\_processes$. For example, using a computer with 4 processors, the processing time would be $T = 0.0121/4 + 0.01181$, $T = 0.01483$ seconds per frame. This is an increase of 37.95% on the processing time.

3.3 Discussion

This chapter describes a novel method for detecting fights. Blobs of movement are first detected and then different features are used

Sub-step	MSecs per frame	Parallelizable?
Load a frame	6.08	No
Convert RGB to Gray image	4.12	Yes
Rescale	7.49	Yes
Binarization	0.41	Yes
Find the regions	5.51	No
Select the K best regions	0.22	No
Calculate the features	0.08	Yes

Table 3.5: Sub-steps of the proposed method

to characterize them. The proposed method makes no assumptions on number of individuals (it can be also used to detect vandalism), body part detection or salient point tracking. Experiments show that the method does not outperform the best methods considered. However, it is much faster while still maintaining useful accuracies ranging from 70% to near 98% depending on the dataset. While other methods tackle a more general problem (such as action recognition) or resort to computationally intensive optical flow computations, the proposed method opens up the possibility of practical implementations.

CHAPTER 4

Method 2: Motion Blur

Inspired by results in human perception of other's actions, this chapter presents a novel fight detection method that uses acceleration patterns as the main feature. Experiments show that accuracy improvements of up to 8% are achieved with respect to state-of-the-art action recognition methods in the most challenging dataset considered. Most importantly, the proposed method is significantly faster.

This chapter is organized as follows. Section 4.1 describes the proposed method. Section 4.2 provides a comparison of the results with other related state-of-the-art methods. Finally, in Section 4.3 the main conclusions are outlined.

4.1 Motion Blur Method

As mentioned above, acceleration patterns are highly informative in the task of fight recognition. In general, acceleration can be inferred from tracked point trajectories. However, we have to note that extreme acceleration implies image blur (see for example Figure 4.1), which makes tracking less precise or even impossible in this case. In this work we propose a different method that can potentially work even with extreme accelerations. Motion blur entails a shift in image

Figure 4.1: Movie dataset. Two consecutive frames in a fight clip from a movie. Note the blur on the left side of the second frame.

content towards low frequencies. Such behavior allows to build an efficient acceleration estimator for video. First, we compute the power spectrum of two consecutive frames. It can be shown that, when there is a sudden motion between the two frames, the power spectrum image of the second frame will depict an ellipse [66], see Figure 4.2. This is due to the fact that any motion blur implies a shift towards lower frequencies. The orientation of the ellipse is perpendicular to the motion direction, the frequencies outside the ellipse being attenuated. Most importantly, the eccentricity of this ellipse is dependent on the acceleration. Basically, the proposed method aims at detecting the sudden presence of such ellipse. In the following, the method is described in detail.

Let I_{i-1} and I_i be two consecutive frames. Motion blur is equivalent to applying a global low-pass oriented filter C, where $\mathcal{F}(\cdot)$ denotes the Fourier Transform.

$$\mathcal{F}(I_i) = \mathcal{F}(I_{i-1}) \cdot C \tag{4.1}$$

Then:

$$C = \frac{\mathcal{F}(I_i)}{\mathcal{F}(I_{i-1})} \tag{4.2}$$

The global low-pass oriented filter in C is the above-mentioned ellipse. Figure 4.3 shows C for the sample image above.

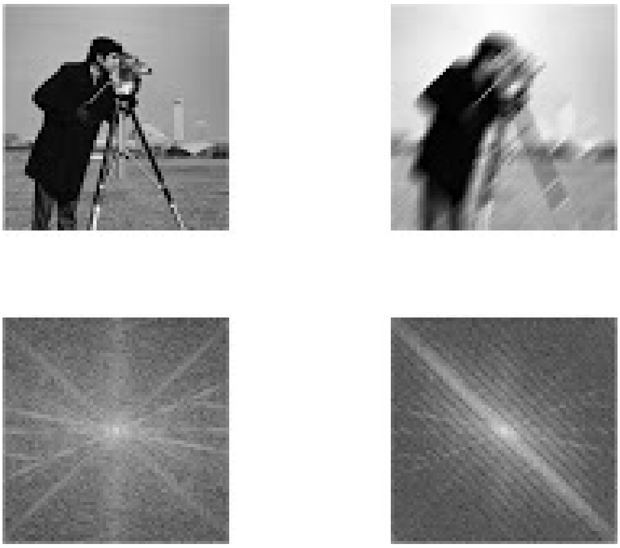

Figure 4.2: Fourier transforms. Top-Left: Sample image. Top-right: With simulated camera motion at 45°. Bottom: Fourier transform of the images.

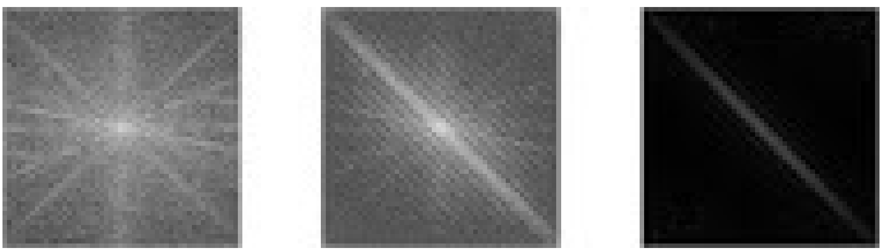

Figure 4.3: Left and center: Fourier transforms of the images in Figure 4.2. Right: C image.

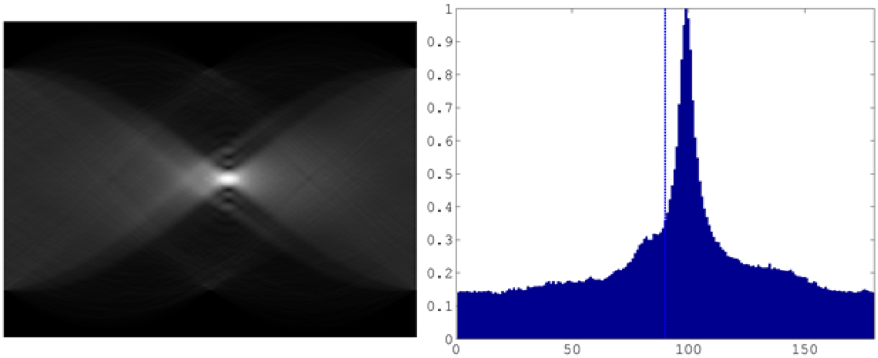

Figure 4.4: Radon transform. Left: Radon transform image of Figure 4.2-left under a simulated camera motion at 98°. The horizontal axis represents angles between 0 and 180°. Right: vertical projection of the Radon image.

For each pair of consecutive frames, we compute the power spectrum using the 2D Fast Fourier Transform (FFT) (in order to avoid edge effects, a Hanning window was applied before computing the FFT). Let us call these spectra images as P_{i-1} and P_i. Next, we simply compute the image:

$$C = \frac{P_i}{P_{i-1}} \tag{4.3}$$

When there is no change between the two frames, the power spectra will be equal and C will have a constant value. When motion has occurred, an ellipse will appear in C. Our objective is then to detect such ellipse and estimate its eccentricity, which represents the magnitude of the acceleration. Ellipse detection can be reliably performed using the Radon transform, which provides image projections along lines with different orientations, see Figure 4.4.

After computing the Radon transform image R, its vertical *maximum* projection vector vp is obtained and normalized to maximum value 1 (see Figure 4.4-right). When there is an ellipse in C, this vector will show a sharp peak, representing the major axis of the ellipse. The kurtosis K of this vector is therefore taken as an estimation of the acceleration.

Note that kurtosis alone cannot be used as a measure, since it is obtained from a normalized vector (i.e. it is dimensionless). Thus, the average power per pixel P of image C is also computed and taken as an additional feature[1]. Without it, any two frames could lead to high kurtosis even without significant motion.

The previous paragraphs have described a procedure that obtains two features K and P for each pair of consecutive frames. Deceleration was also considered as an additional feature, and it can be obtained by swapping the consecutive frames and applying the same algorithm explained above. For video sequences, we compute histograms of these features, so that acceleration/deceleration patterns can be inferred. Even though some actions are highly-energetic and still do not represent fights, the hypothesis is that fights and energetic actions will have differences in the acceleration patterns.

In a variant of the proposed method, ellipse eccentricity can be estimated by first locating the position p of the maximum of vp. This maximum is associated to the major axis of the ellipse. The minor axis is then located at position:

$$\begin{aligned} q &= p + 90^\circ \quad if(p + 90^\circ) \leq 180^\circ \\ q &= p - 90^\circ \quad otherwise \end{aligned} \tag{4.4}$$

Note that the proposed method does not involve tracking or optical-flow estimations. In line with other fight detection work, it adopts a global approach in which all blur in the scene will contribute to the system's output. On the other hand, it is important to note that global (i.e. camera) motion can also cause blur in the image. In order to remove such blur, it is necessary to perform a deconvolution preprocessing step. The phase correlation technique ([67]) is first used to infer global motion between each pair of consecutive frames. If global motion is detected, the estimated angle and length of the displacement is used to form a Point Spread Function (PSF) with which to perform deconvolution of the second frame (we used the Lucy-Richardson iterative deconvolution method). This is intended to re-

[1] The average power per pixel is simply the sum of pixel values in C divided by the total number of pixels.

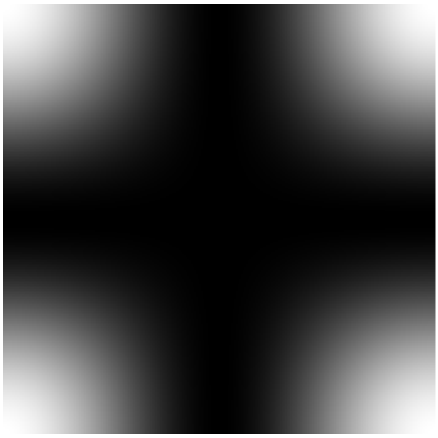

Figure 4.5: Inverted Hanning window, i.e. $1 - H_{x,y}$. Brighter zones represent higher values.

move the blur caused by global motion (camera motion), while any local blurs will remain. The method described above is then applied to the pair of frames as shown in detail in Algorithm 1.

When backgrounds are relatively uniform and displacements small, global motion estimation may still fail. The fail mode is typically represented by real global motion which goes undetected, i.e. an incorrect (0,0) displacement is estimated. Since the proposed method is heavily dependent on global motion, further measures must be taken in practice to at least detect the presence of global motion versus local motion. The window function mentioned above restricts processing to the inner part of the image. It is reasonable to assume that, when motion is global, changes in the outer part of the image will be relatively on a par with those in the inner part. Thus, an additional 'Outer Energy' O feature was computed and used as the other ones:

$$O_{x,y,t} = \frac{|f_{x,y,t} - f_{x,y,t-1}| \cdot (1 - H_{x,y})}{M \cdot N} \tag{4.5}$$

where $f_{x,y,t}$ and $f_{x,y,t-1}$ are (MxN-pixel) frames captured at times t and $t-1$ and H is the Hanning window, see Figure 4.5.

Algorithm 1 shows the detailed steps of the proposed method.

There are two reasons why the proposed method is not simply detecting all 'action sequences' (in a cinematic sense). On the one hand, the blur being detected is only associated to the most extreme motion, mostly associated to fights. On the other hand, the additional use of histograms account for patterns of such extreme motion that separate fights from other other types of actions.

4.2 Experiments

The proposed method was assessed using three different datasets and compared with five other methods. The *Hockey* and *Movies* datasets are used, which were explained in Section 2.1. Also the *UCF101-E* dataset is used, which was introduced in Section 3.2.

In the experiments, the Radon transform was computed between 0 and 180 in steps of $\theta = 9$ degrees. Histograms of 8 bins were computed for each of the three main features (acceleration, deceleration, power and outer energy, see the previous Section).

The STIP, MoSIFT and ViF methods, which were explained in Section 3.2, are used to compare the proposed method. Also, the previously proposed method that creates a descriptor for detecting fights based on blobs of movements, which was explained in detail in chapter 3.

Table 4.1 shows the results measured using 5 repetitions (10 repetitions for the *UCF101-E* dataset) of 10-fold cross-validation and three different classifiers: SVM (linear kernel), AdaBoost (100 classifiers, weighted voting) and RF (50 trees).

Figure 4.6 shows the ROC curves of the first four related methods, for the SVM and RF classifiers, (AdaBoost is not shown since it was consistently the worst classifier).

The results on the *Movies* dataset show that the proposed method does not improve upon the other considered methods, although accuracy is not very far. Some of the results on the *Hockey* dataset fares well among other previously reported results, see Table 4.2, the best result of the proposed method in this dataset is 97.4%.

Input: S = (Short) sequence of gray scale images. Each image in S is denoted as $f_{x,y,t}$, where $x = 1, 2, \ldots, N$, $y = 1, 2, \ldots, M$ and $t = 1, 2, \ldots, T$.
Result: $4 \cdot n\_bins$ discriminant features

for $t = 1$ to T **do**

1. Using phase correlation, estimate angle and length of displacement l between $f_{x,y,t-1}$ and $f_{x,y,t}$

2. Perform deconvolution over the second frame $f_{x,y,t}$

3. Apply a Hanning Window to $f_{x,y,t}$:

$$g_{x,y,t} = f_{x,y,t} \cdot H_{x,y}$$

where $H_{x,y} = h(N) \cdot h(M)'$ and h is a column vector given by:

$$h(L) = \frac{1}{2}\left[1 - \cos\left(2\pi\frac{l}{L}\right)\right], \text{ for } l = 1, 2, \cdots, L$$

4. Apply FFT to $g_{x,y,t}$: $F_{v,w,t} = \mathcal{F}(g_{x,y,t})$

5. Compute $C_{v,w} = F_{v,w,t}/F_{v,w,t-1}$

6. Compute Radon transform of C: $R_{d,\theta} = \mathcal{R}(C_{v,w})$

7. Compute vertical max projection of R: $p_\theta = \max_d(R_{d,\theta})$

8. Normalize $p_\theta = p_\theta/m$, where $m = \max_\theta(p_\theta)$

9. Compute feature $A_t = Kurtosis(p_\theta)$

10. Compute feature $P_t = mean_{v,w}C_{v,w}$

11. Compute feature D_t (deceleration) using the same steps above but swapping t for $t-1$

12. Compute feature O_t using Eq. (4.5)

end
return
$Histogram(A, n\_bins), Histogram(P, n\_bins), Histogram(D, n\_bins), Histogram(O, n\_bins)$

Algorithm 1: Algorithm for computing the main features in the proposed method.

Method	Classifier	Movies	Hockey	UCF101-E
BoW+ STIP	SVM	82.5±1.1/0.88	88.5±0.2/0.94	72.5±1.5/0.74
	AdaBoost	74.3±2.3/0.81	86.5±0.2/0.94	63.1±1.9/0.68
	RF	97.7±0.5/0.99	96.5±0.2/0.99	87.3±0.8/0.94
BoW+ MoSIFT	SVM	84.2±1.5/0.92	91.2±0.6/0.84	81.3± 1/0.86
	AdaBoost	74.3±2.3/0.92	86.9±1.6/0.92	52.8±3.6/0.62
	RF	75.1±1.6/0.81	96.7±0.7/0.99	86.3±0.8/0.93
ViF	SVM	96.7±0.3/0.98	82.3±0.2/0.91	77.7±2.16/0.87
	AdaBoost	92.8±0.4/0.97	82.2±0.4/0.91	78.4±1.7/0.86
	RF	88.9±1.2/0.97	82.4±0.6/0.9	77±1.2/0.85
LMP	SVM	84.4±0.8/0.92	75.9±0.3/0.84	65.9±1.5/0.74
	AdaBoost	81.5±2.1/0.86	76.5±0.9/0.82	67.1±1/0.71
	RF	92±1/0.96	77.7±0.6/0.85	71.4±1.6/0.78
Serrano[48]	SVM	87.2±0.7/0.97	72.5±0.5/0.76	71,2±0.7/0.78
	AdaBoost	81.7±0.2/0.82	71.7±0.3/0.72	71±0.8/0.72
	RF	**97.8±0.4/0.97**	82.4±0.6/0.9	79.5±0.9/0.85
Proposed	SVM	85.5±1.0/0.91	90.1±0.1/0.94	94.0±4.2/0.94
	AdaBoost	81.6±1.5/0.90	90.1±0.1/0.90	92.3±6.2/0.92
	RF	91.6±0.7/0.97	**97.4±2.0/0.97**	**97.4±2.1/0.89**

Table 4.1: This table shows the results for the three datasets using SVM, AdaBoost and RF classifiers for the related methods and the proposed method. Three measures have been calculated: mean accuracy, standard deviation accuracy and AUC for each method and classifier.

Reference	Year	Accuracy
Bermejo [36]	2011	91.7%
Hassner [9]	2012	82.9%
Huang [55]	2014	86.9%
Ding [27]	2014	91.0%
Xu [16]	2014	94.3%
Zhao [21]	2015	95.3%

Table 4.2: Some of the best results reported from the state-of-the-art *Hockey* dataset, sorted by year of publication.

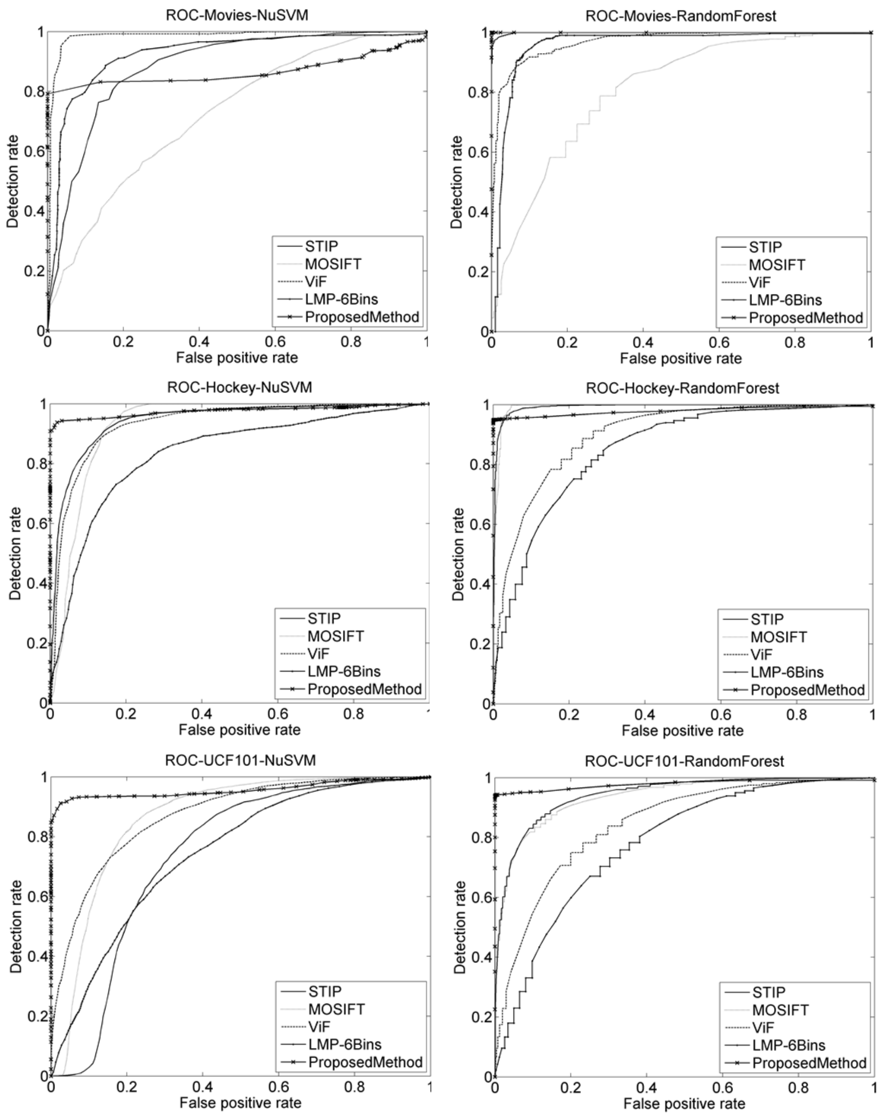

Figure 4.6: ROC curves of the five methods considered. Top row: *Movies* dataset, Middle row: *Hockey* dataset, Bottom rom: *UCF101-E* dataset. Left column: SVM classifier, Right column: RF classifier.

Reference	Year	Result
Laptev [68]	2013	91% accuracy in Sports group
Hyunjong [69]	2013	71.3% accuracy in Sports group
Hyunjong [69]	2013	100% accuracy in Punching, 66.7% accuracy in Sumo
Karen [70]	2014	100% recall in Sumo, 90% recall in Punching

Table 4.3: Results of either the "Punching" and "Sumo" classes or the Sports group of *UCF101*.

In the more challenging *UCF101-E* and *Hockey* datasets the proposed method is more robust than the other methods considered. Although we extended the fight set with the *Hockey* and *Movies* fights, to put these results in context we now consider the results in action recognition studies using the *UCF101-E* dataset. The best result of the proposed method with this dataset is 95.1% accuracy. Table 4.3 shows some of the latest results obtained in such studies for either the "Punching" and "Sumo" classes or the Sports group of the *UCF101* dataset (which includes "Punching" and "Sumo").

Figure 4.7 shows the results of a one-way ANOVA followed by a multiple comparison (using the Tukey-Kramer method), performed with the five methods considered for each considered dataset.

The effect of the step in θ for computing the Radon transform is obviously a trade-off between accuracy and speed. Figure 4.8 shows the accuracy and speed versus the step for θ in the UCF101-E dataset. $\theta = 9$ gave the best accuracy, although larger values do not provide much worse accuracies.

Table 4.4 shows the number of features used for classification and the computational cost measured (for feature extraction). The code for both STIP and MoSIFT was compiled, while the code for rest of the methods was interpreted and used no parallelization. These results show a significant improvement in speed. The fact that only 32 features are necessary is an additional advantage for practical implementations.

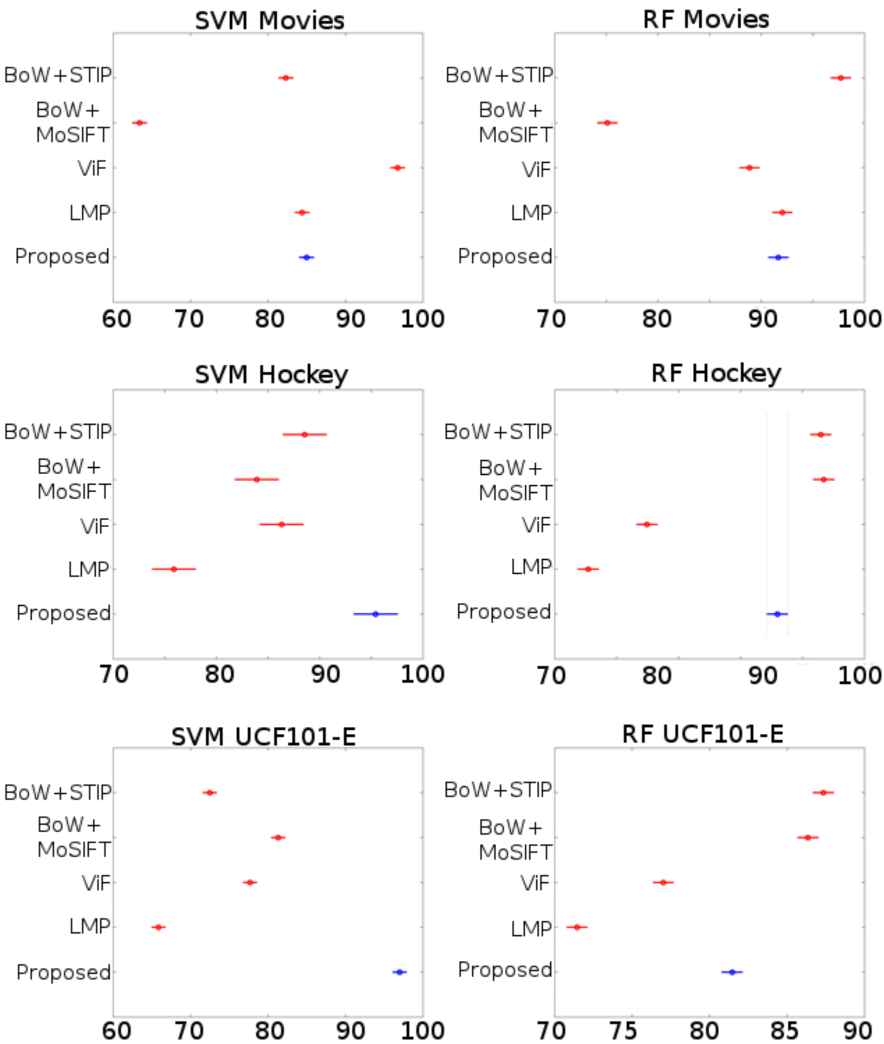

Figure 4.7: One-way ANOVA. Methods vs Accuracies. Top row: *Movies* dataset, Middle row: *Hockey* dataset, Bottom row: the UCF101-E dataset. Left column: SVM classifier, Right column: RF classifier.

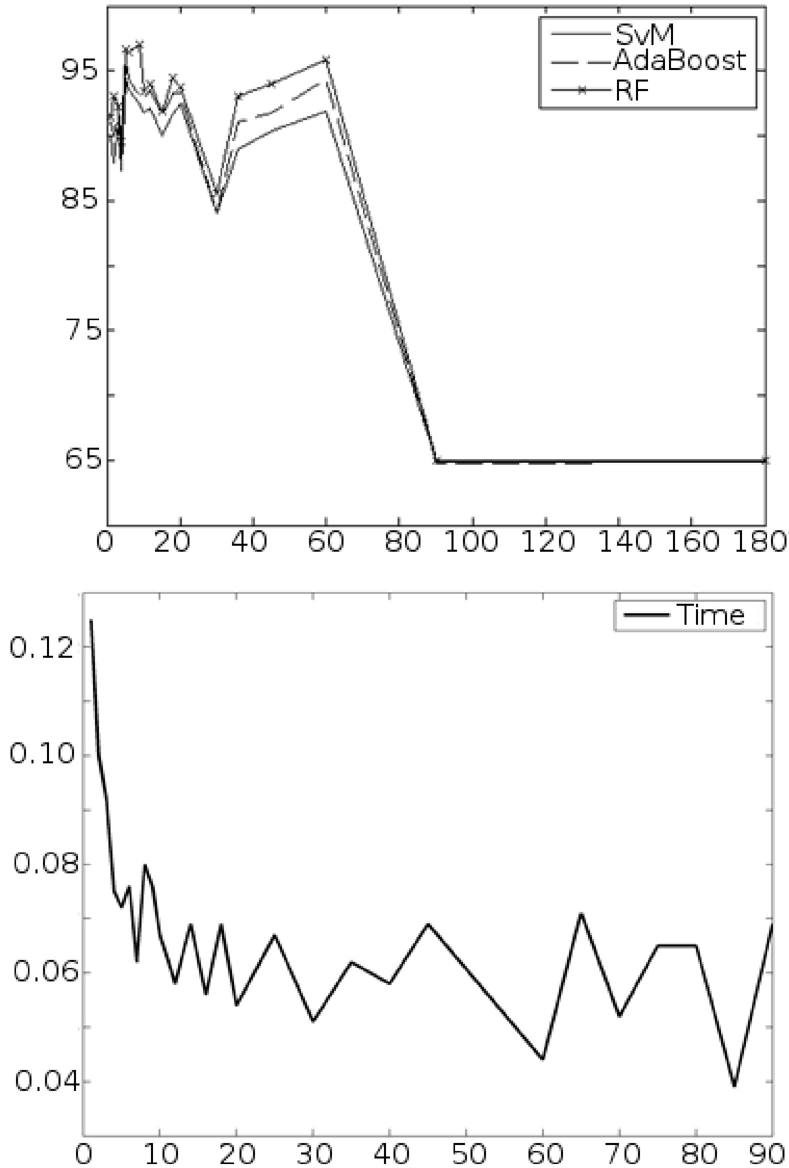

Figure 4.8: Top: Accuracy vs step for θ. Bottom: Feature extraction time per frame (seconds) vs step for θ.

Method	Features/sequence	Msecs/frame
BoW+MoSIFT	500	661.5
BoW+STIP	500	293.5
ViF	96000	454.5
LMP	10368	151.6
Serrano [48]	2940	26.5
Proposed Method	32	72.5

Table 4.4: Feature extraction times. Average times measured with the non-fight videos in the *UCF101-E* dataset, on an Intel Xeon computer with 2 processors at 2.90Ghz.

Taken together these results suggest that the method may effectively work as an efficient fight detector for generic settings.

Finally, we note that many powerful descriptors for action recognition consider both motion and appearance. However, similar to the case of human perception, our results suggest that motion is sufficient in this task. When motion alone is considered, some methods resort to optical flow and point trajectories, estimating both magnitude and direction. Again, our results suggest that magnitude is sufficient for this application. Note also that magnitude and direction are antagonistic measures. Because of the involved blur, the larger the motion magnitude the more difficult it is to measure direction. This may be the reason behind the better performance of the proposed method as compared to the others.

4.3 Discussion

Based on the observation that kinematic information suffices for human perception of other's actions, in this work a novel detection method is proposed which uses acceleration/deceleration patterns as the main discriminating feature. The method shows promising features for surveillance scenarios and it also performs relatively well when considering challenging actions such as those that occur in sports. The proposed method was assessed with five other methods using three different datasets. Accuracy improvements of up to 8% with respect to state-of-the-art action recognition techniques were achieved in the most challenging dataset (two of the three datasets). We hypothesize that when motion is sufficient for recognition, appearance not only takes significant additional computation but it also may confuse the detector.

Another interpretation is that a sort of overfitting may be occurring in that case. When considering motion alone, extant methods estimate both magnitude and direction. However, direction becomes less precise as magnitudes involved become larger. Being based only on magnitude, the proposed method performs well in these cases.

This method makes no assumptions on number of individuals (it can be also used to detect vandalism), body part detection or salient point tracking. Additionally, it is faster than other methods and uses only 32 features, which opens up the possibility of practical implementations.

CHAPTER 5

Method 3: STEC Trajectories

In this chapter, Spatio-Temporal Elastic Cuboid (STEC) trajectories are proposed for fight detection. This method is based on the use of blob movements to create trajectories that capture and model the different motions that are specific to a fight. The proposed method is robust to the specific shapes and positions of the individuals.

The proposed method consists of two independent stages: the Spatio-Temporal Elastic Cuboid (STEC) trajectories (descriptor) and Hough Forests classifier.

This chapter is organized as follows. Section 5.1 describes the proposed method. Section 5.2 explains the adapted Hough Forest classifier. Section 5.3 provides a comparison of the results with other related state-of-the-art methods. Finally, in Section 5.4 the main conclusions are outlined.

5.1 Spatio-Temporal Elastic Cuboid trajectories

The first step is to define the novel descriptor called Spatio-Temporal Elastic Cuboid STEC trajectories, see an example in Figure 5.1 (bottom) with a fist in motion compared with a classical cuboid (top). STEC trajectories are always centered around tracked parts,

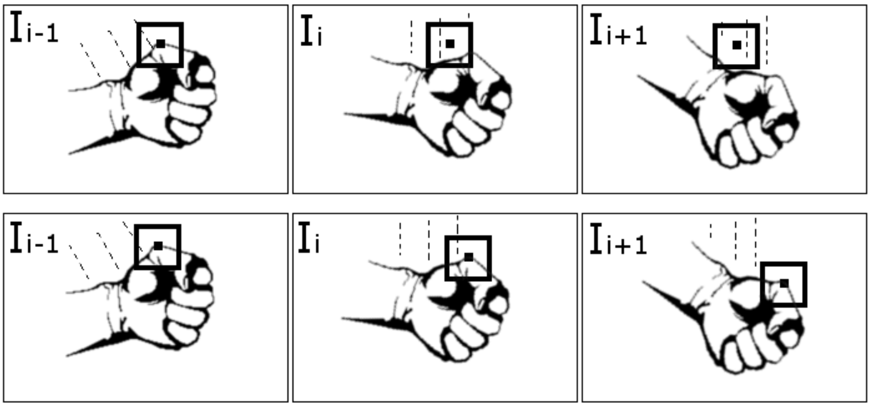

Figure 5.1: A fist in motion. Top row: with a classical cuboid. Bottom row: with a STEC trajectory.

whereas the classic cuboids are on a fixed position. Thus, the proposed descriptor only focuses on moving parts. Furthermore, in order to represent these STEC trajectories, an extension of Hu et al. [71] method is proposed. These authors propose a novel approach to analyze the topological features of hand postures at multiple scales. We extend this approach in time to represent different parts involved in an action.

In this work, it is hypothesized that a fight can be described with the position, size, rotation and shape of motion areas. Figure 5.2 shows a simple example of a STEC trajectory that tracks a part in motion (the fist). State-of-the-art methods [72, 73, 74, 75, 76] etc. that extract spatio-temporal features use cuboids from a grid (Figure 5.2 top) or from salient features (Figure 5.2 middle) in videos. In this work we use “elastic” cuboids (or STEC trajectories) (Figure 5.2 bottom) that track different parts in motion and then extract features. The "elastic" concept is used to differentiate them from the classic cuboids that have a fixed spatial position throughout neighboring frames.

The process to obtain the STEC trajectories is summarized in the following:

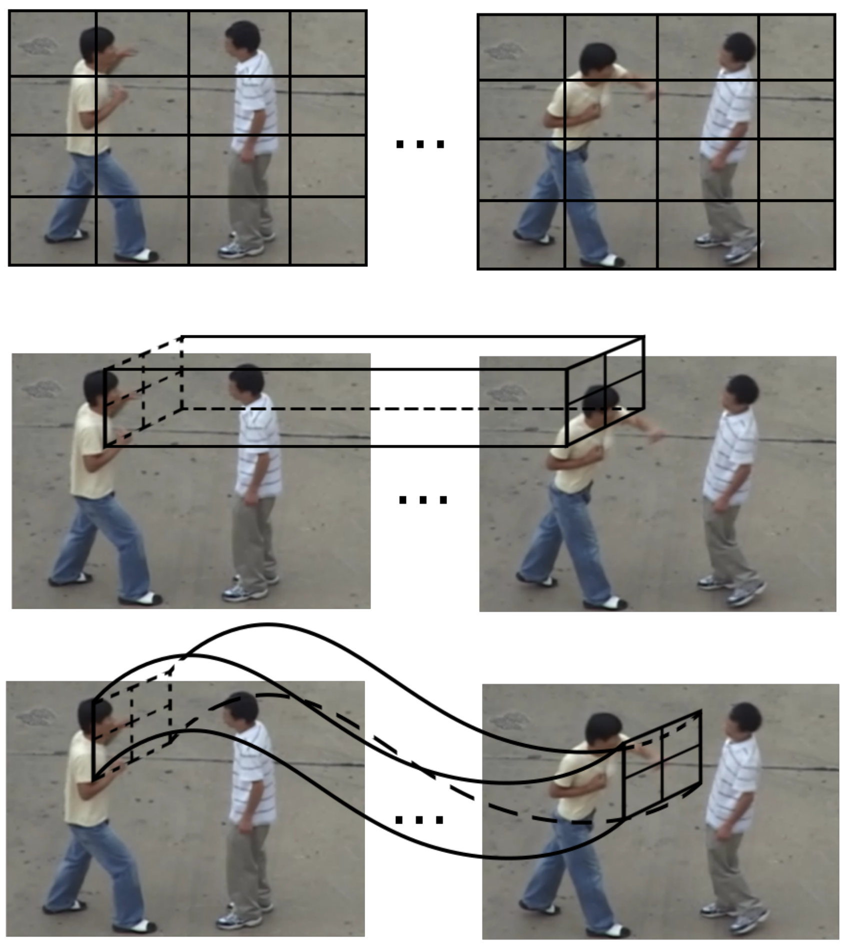

Figure 5.2: A fight sequence (from UT-Interaction dataset) showing a fist. Top row: with dense cuboids. Middle row: with sparse cuboids (only one around the fist). Bottom row: with a STEC trajectory (also around the fist).

1. A binarized image is obtained containing motion between consecutive frames. A standard binarization method is applied using an adaptive threshold. An image with motion blobs is obtained for every two consecutive frames.

2. The best K blobs are selected from the binarized images.

3. A body part in motion (for example, the hand in a punch) is modeled by a trajectory of motion blobs (STEC). Here, it is necessary to link the motion blobs in the previous image with those in the current image. A method is proposed to associate two consecutive blobs using distance, area and shape. With these linked blobs a STEC trajectory is built.

4. Characterize STEC trajectories. To model a STEC trajectory, relative sizes, positions, orientations and shapes are obtained.

In the following, these steps are described in detail:
Firstly, a short sequence $S(s)$ of gray images is denoted as:

$$S(s) = I_t(x, y) \tag{5.1}$$

where $s \in \mathbb{Z}$ (number of sequences), $x \in \{1, 2, \cdots, N\}$, $y \in \{1, 2, \cdots, M\}$ and $t \in \{1, 2, \cdots, T\}$. N and M are the number of rows and columns of each frame I and T is the number of frames.

Let $I_{t-1}(x, y)$ and $I_t(x, y)$ be two consecutive frames in the sequence. The absolute difference between consecutive images is then computed as:

$$E_t(x, y) = |I_{t-1}(x, y) - I_t(x, y)| \tag{5.2}$$

Then, E is binarized using a traditional binarization method:

$$F_t(x, y) = \begin{cases} 1, & \text{if } E_t(x, y) > 255 \cdot H_t, \\ 0, & \text{otherwise} \end{cases} \tag{5.3}$$

where H_t is an adaptive threshold used to binarize each $E_t(x, y)$ image, $0 < H_t < 1$ and $H_t \in \mathbb{R}$, calculated with the Otsu method. These binarized images contain the areas with motion.

The second step is to locate and select a set of blobs in each image $F_t(x, y)$. The K largest blobs are selected for each binarized image. The blobs are represented as a function $B_{b,t}(x, y) = b$ where $b \in \{0, 1, \cdots, J\}$ is the unique index for each blob in the image, and J is the number of blobs in the image $F_t(x, y)$. Each blob $B_{b,t}(x, y)$ is here defined as an image that contains a set of adjacent points, neighborhood ($\mathcal{N}_b$), where $F_t(x, y) = 1$.

$$B_{b,t}(x, y) = \begin{cases} b, & \text{if } \bigcup\limits_{(x,y)\in\mathcal{N}_b} F_t(x, y) = 1, \\ 0, & \text{otherwise} \end{cases} \tag{5.4}$$

and

$$m_b = \sum_{(x,y)\in\mathcal{N}_b} F_t(x, y) \tag{5.5}$$

where $m_1, \cdots, m_b, \cdots, m_J$ are the number of adjacent pixels of each blob respectively and $m_1 + \cdots + m_b + \cdots + m_J \leq N \cdot M$. These blobs are calculated from each binarized image. Afterwards, the K largest blobs are selected using their blob areas ($A_{b,t}$) that are mathematically defined as:

$$A_{b,t} = \frac{1}{b} \sum_{B_{b,t}(x,y)=b} B_{b,t}(x, y) \tag{5.6}$$

Here it is assumed that the blobs with largest areas are the most representative in each frame. However, when there is little or no movement in the frame, some of these K largest blobs may not be representative. Therefore the m_t (minimum threshold area) is defined to reject blobs with little area as:

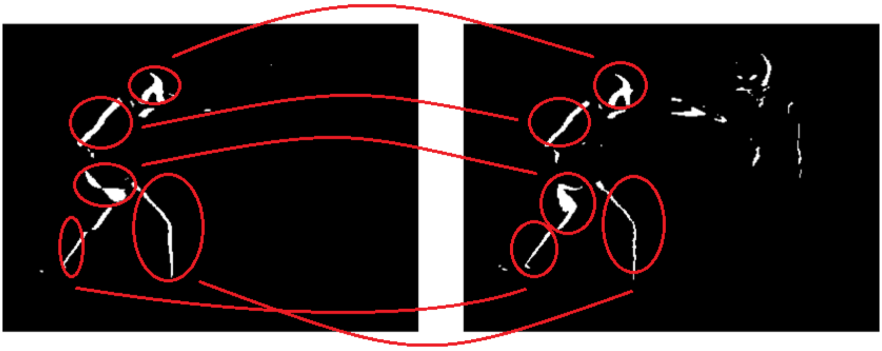

Figure 5.3: A punching sequence with the best five blobs marked in two consecutive frames. These five blobs have been also linked in this pair of frames.

$$m_t = \begin{cases} \frac{1}{J}\sum_{b=1}^{J} A_{b,t}, & \text{if } \frac{1}{J}\sum_{b=1}^{J} A_{b,t} > M_a \cdot a_{max} \\ M_a \cdot a_{max}, & \text{otherwise} \end{cases} \tag{5.7}$$

where M_a (minimum areas) is a parameter to set, $0 > M_a > 1$, $M_a \in \mathbb{R}$ and $a_{max} = N \cdot M$ (pixel area of one frame). Finally, if $A_{b,t} < m_t$ then the blob B_b is rejected. Note that K is thus a maximum of blobs, and if there is little motion fewer blobs will be selected.

The third step is to link a selected blob in the current frame with another one in the next frame. Let us use a sequence where K blobs have been selected (example, see the sequence in Figure 5.3 using 5 marked blobs). For each blob in frame t we need to find where the next blob is in $t+1$. For this purpose, a match measure will be defined for a blob in t and every blob in $t+1$. This measure is calculated based on the relative areas, distances and shapes of the two blobs. Let us use $B_{b,t}$ for $B_{b,t}(x, y)$, this measure is defined as:

$$\varphi(B_{b,t}; B_{b',t+1}) = \frac{1}{area(B_{b,t}, B_{b',t+1}) \cdot dist(B_{b,t}, B_{b',t+1}) \cdot shape(B_{b,t}, B_{b',t+1})} \tag{5.8}$$

where $area(B_{b,t}, B_{b',t+1})$ represents the absolute difference of two area blobs, and can be rewritten as:

$$area(B_{b,t}, B_{b',t+1}) = \mid A_{b,t+1} - A_{b',t} \mid \tag{5.9}$$

The term $dist(B_{b,t+1}, B_{b',t})$ represents the distance between two points. The centroids of the blobs are used to represent each blob position and calculate their distances. This part of the equation can be rewritten as:

$$\begin{aligned} &dist(B_{b,t}, B_{b',t+1}) = \\ &\sqrt{(CX_{b,t} - CX_{b',t+1})^2 + (CY_{b,t} - CY_{b',t+1})^2}, \end{aligned} \tag{5.10}$$

where $CX_{b,t}$ and $CY_{b,t}$ are the centroids of a blob that is defined as:

$$CX_{b,t} = \frac{1}{m_b} \sum_{B_{b,t}(x,y)=b} x, CY_{b,t} = \frac{1}{m_b} \sum_{B_{b,t}(x,y)=b} y \tag{5.11}$$

The last part of the equation is the term $shape(B_{b,t}, B_{b',t+1})$ that provides a shape difference between the two blobs. The blob $B_{b,t}$ shape is estimated using HoG method that is applied on the difference image region for blob $B_{b,t}$. The HoG counts occurrences of gradient orientation in localized portions of an image. This algorithm is similar to that of edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts, but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy. An example is shown in Figure 5.4, where the HoG method is applied on the blob center to estimate its shape (as the predominant angle). The vector of maximum gradient is used ($HOG_{max}(B_{b,t})$). Then this part of the equation can be rewritten as:

$$\begin{aligned} &shape(B_{b,t}, B_{b',t+1}) = \\ &\mid\mid HOG_{max}(B_{b,t}) - HOG_{max}(B_{b,t+1}) \mid\mid \end{aligned} \tag{5.12}$$

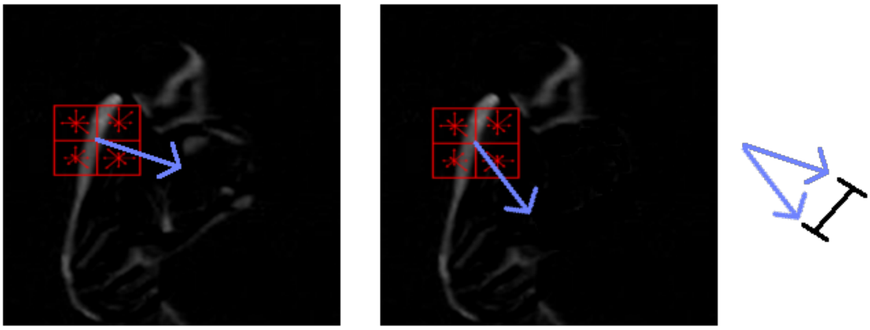

Figure 5.4: Left and middle: 2 consecutive difference images from a punching sequence where the HoG method is applied on the blob center, the blue lines are the predominant direction vectors (amplified). Right: these blue line vectors are subtracted and the magnitude is calculated, the black line.

In summary, these two HoG vectors have been obtained, subtracted and the magnitude of this result vector is calculated.

Now each pair of blobs should be linked with the following maximization function where the best blob b' is selected:

$$\max_{b'} \left(\varphi(B_{b,t}; B_{b',t+1})\right) \tag{5.13}$$

Each pair of blobs is linked in time t. This process is repeated each time step to build the trajectories. Parameter L (length) limits the number of blobs paired along a trajectory, for example see Figure 5.5 with $L = 5$ blobs. In order to reduce mismatches, i. e., pairs of blobs that are far away and very different, parameter M_t (minimum trajectories) is introduced to reject trajectories, where M_t can be chosen between $0 > M_t > 1$ and $M_t \in \mathbb{R}$. Finally, if $area(B_{b,t}, B_{b',t+1}) > a_{max} \cdot M_t$ or $dist(B_{b,t}, B_{b',t+1}) > d_{max} \cdot M_t$ then the current trajectory is rejected, where $d_{max} = \sqrt{N^2 + M^2}$ (diagonal distance of one frame).

The last step is how to describe each trajectory. In order to build a robust descriptor, relative features between pairs of consecutive blobs in the trajectory are calculated. Different features of the blobs are

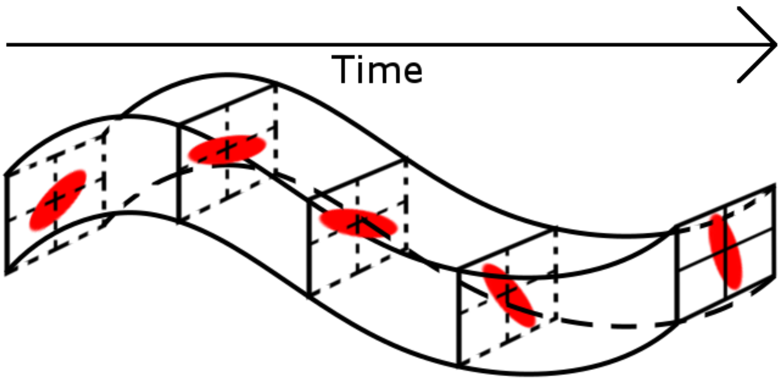

Figure 5.5: A 5-length STEC trajectory where the blobs have been linked.

used to encode the position, size, orientation and shape. It is proposed to use 9 relative measures of each blob ($9(L-1)$ features), the length trajectory (1 feature) and a set of features that encodes the shape in space-time (x, y, t) ($N_e \cdot K \cdot L$ features). The first 9 relative measures are centroid (x and y), area, distance, perimeter, major and minor axes, orientation and number of other blobs. These relative measures are calculated by subtracting the values at t and $t+1$. The centroid, area and distance were already defined. To estimate the perimeter ($P_{b,t}$) the Sobel operator is used to detect the edges on $B_{b,t}(x, y)$. In summary, the proposed method obtains $L \cdot (N_e \cdot K + 9) - 8$ features per trajectory.

The features that can model the shape of the trajectories in space-time ($L \cdot N_e$ features), come from an extension of [71]. In [71] the authors propose a novel approach to analyze the topological features of hand postures at multiple scales. They compute the convex hull on the hand region and considered the complementary space of the hand as holes. Then they apply multiple morphological erosion operations (N_e) over these holes and count the number of regions that are formed. In this paper, an extension of this approach is proposed. This method is applied for each blob on the trajectory. Then the sequence shape in space-time can be modeled. An example using the multiscale erosion method in space-time is shown in Figure 5.6 for K trajectories.

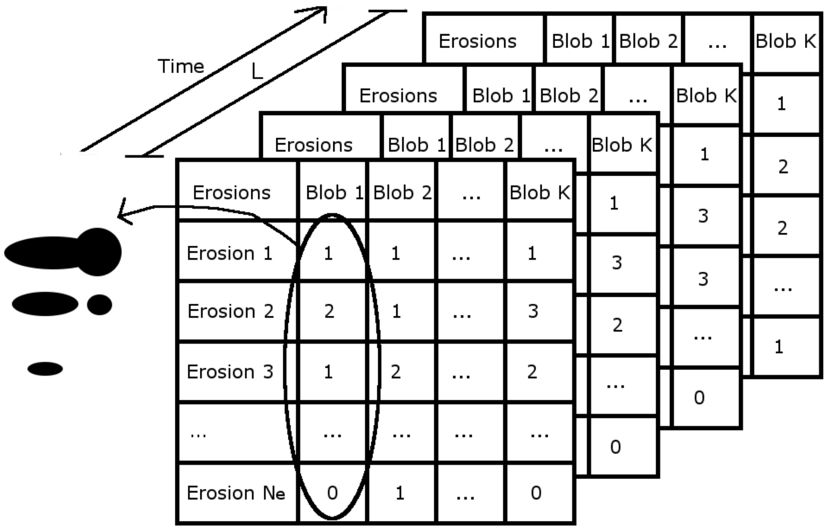

Figure 5.6: Left: the first blob with three erosions. Right: the matrix result for K trajectories of L blobs using the extended erosion method. The column with the ellipse is the result for the first blob that counts the number of regions for each erosion step.

5.2 Hough Forests

The second step is an adaptation of Hough Forests ([77], [78], [76]) are proposed to classify STEC trajectories. In contrast, the classical BoW [79] model assumes independence between spatio-temporal "words" and does not make use of the rich spatio-temporal relationships inherent in actions. Hough Forests leverage this important information.

Hough Forests consist of a set of random trees [80] that are trained to learn a mapping from densely-sampled D-dimensional feature cuboids to their corresponding votes in a Hough space $\mathbb{H} \subseteq \mathbb{R}^H$. The Hough space encodes the hypothesis for an object/action position in scale(time)-space and class. The term *cuboid* below is used in a generalized sense to represent a local image patch ($D = 2$) or video space-temporal neighborhood ($D = 3$) depending on the task.

Each tree τ in Hough Forests $\tau = T_{tree}$ is constructed from a set of feature *cuboids* $P_i = (F_i, c_i, d_i)$ that are randomly sampled from the image or video sequence; where F_i are the extracted features from a *cuboid* of fixed size (D) in $\mathbb{R}^D$; c_i is the class label for the sample and d_i is a displacement vector from the cuboid pointing toward the spatio-temporal center of the action. The negative classes have $d_i = 0$. In [78] and [76], cuboids are used with fixed dimensions 16×16 and $16 \times 16 \times 5$ for images and videos respectively. Then for each *cuboid* the grayscale intensity, absolute value of x, y and $time$ derivatives, absolute value of optical flow in x and y are obtained. In this work, Hough Forests are trained with the STEC trajectories.

Each leaf node L stores the probability of the cuboids belonging to the object class $\varphi(c \mid L)$, estimated as the proportion of cuboids per class label reaching the leaf after training, and $D_c^L = \{d_i\}_{c_i=c'}$ the cuboids respective displacement vectors. In this work, each non-leaf node is modified to assign a binary test from a set of input vector features (F). The binary test is now defined by a composition of two features values $(p, q) \in \mathbb{R}^D$ with some offset O_s. Where D is the dimension of the input vector, p and q correspond with two different feature positions from the input vector. The binary test (b) on a non-leaf node is redefined as:

$$b_{p,q,O_s}(F) = \begin{cases} 1, & if F(p) < F(q) \cdot O_s \\ 0, & \text{otherwise} \end{cases} \tag{5.14}$$

The next steps of the Hough Forests method are constructed according to [76]. The Hough Forests output for a sequence (set of features) is an image with their corresponding votes in Hough space $\mathbb{H} \subseteq \mathbb{R}^H$. Besides, the maxima class probability can be searched by applying a Parzen estimator with a Gaussian kernel. The accumulation of the probabilities is made with the summation criteria to make the final decision.

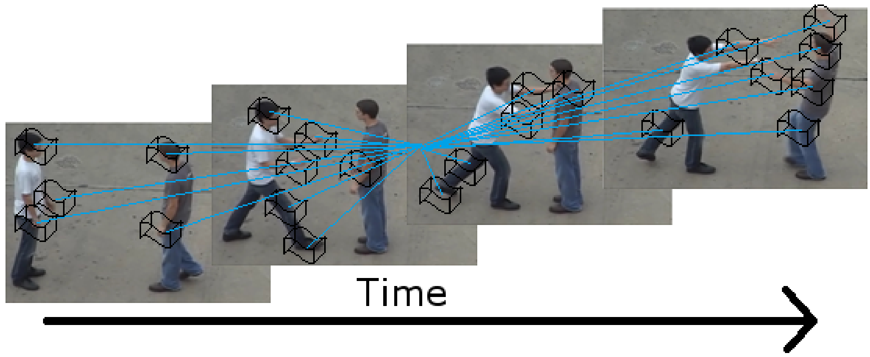

Figure 5.7: A pushing sequence with some STEC trajectories represented. Each STEC trajectory points to the action center.

The main reason to use Hough Forests in this work is: this classifier does not need a previous BoW to cluster and represent the trajectories for each sequence. Each sequence has a different number of trajectories that depend on the amount of motion. Then no matter what classifier is to be used, a previous clustering is needed. On the other hand, under a BoW framework the sequencing of the STEC trajectories would be lost, and this aspect is very important to model a fight action. Hough Forests use all trajectories (features) independently without losing temporal information, see an example in Figure 5.7 where each STEC trajectory gives spatio-temporal information about the action and its center.

The Hough Forests classifier creates a Hough Voting Space when a new sequence (set of features) is tested. Each feature produces a vote in space-time and class. To see an example on a fight *Hockey* sequence, the mean of the space votes in time is represented in Figure 5.8. It is now possible to know where the fight/non-fight is. Besides, the accumulation of votes is calculated for each time step, see Figure 5.9, so it is also possible to know when the fight/non-fight occurs.

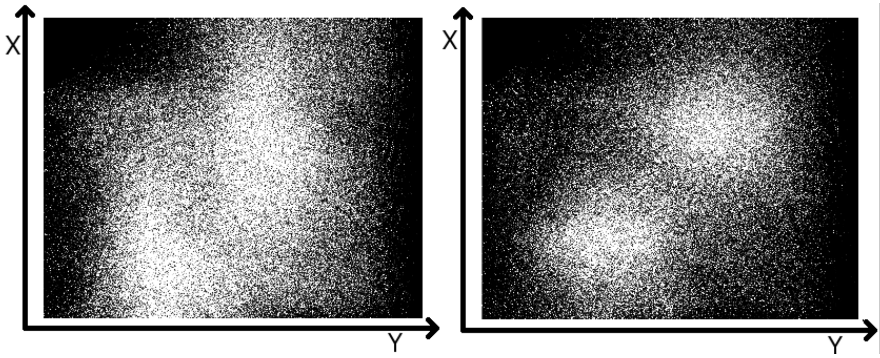

Figure 5.8: The sum space votes in time from a fight and non-fight *Hockey* sequence, respectively. Left: the fight space votes. Right: the non-fight space votes

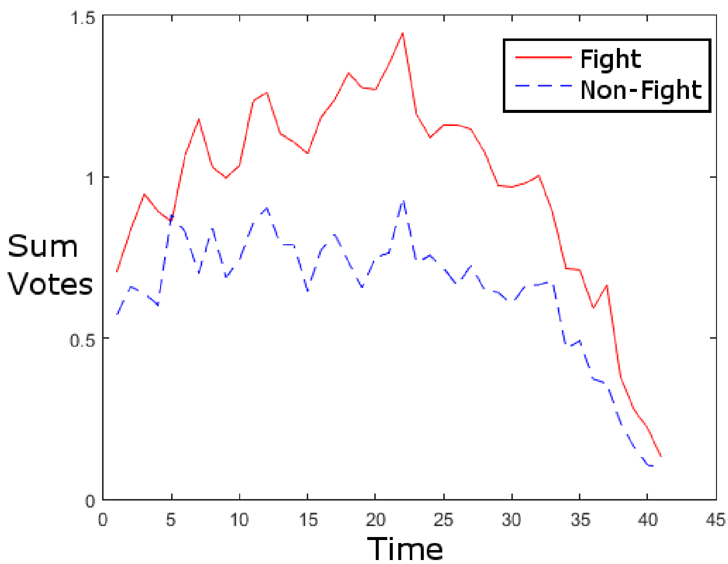

Figure 5.9: The sum votes in space for each time step. Using the same fight *Hockey* sequence.

5.3 Experiments

The proposed method is assessed using three different datasets: *UT*, *Movies* and *Hockey* datasets and compared to nine related methods. In order to evaluate the performance of the proposed descriptor and classifier, a sparse set of STEC trajectories is obtained from each sequence. Then, using this sparse set of features, a Hough Forests classifier is trained and tested.

5.3.1 Method parameters

In order to obtain an appropriate set of parameters, a grid search optimization method was used. This simple optimization method is performed on the training/validation part of the *UT1* dataset (80% and 10% of the dataset, respectively). The value being optimized is the average classification accuracy with 10-fold cross validation. Besides, to reduce variance, three runs are separately performed and the average of accuracy is considered. The following parameters of the proposed algorithm are optimized: the maximum number of motion blobs per frame (K), minimum areas threshold (M_a), minimum trajectories threshold to link blobs (M_t), trajectory length (L) and number of erosions (N_e). For the optimization, the parameter space is divided into a rough grid with manually selected points.

The optimization converged to the following set of parameters: the maximum number of motion blobs per frame $K = 10$; minimum areas $M_a = 0.0001$; minimum trajectories to link blobs $M_t = 0.2$; trajectory length $L = 4$; and number of erosions $N_e = 6$. The proposed method also estimates the shape between blobs with the HoG algorithm, and it has been used with these standard values: the block size is 2×2, cell size is 8×8, overlap is half the block size, number of bins is 9 and non-signed orientation is used.

5.3.2 Experimental Results

In all subsequent experiments, the parameters used were those obtained as discussed in Section 5.3.1. The Hough Forests classifier was trained with 10 trees, depth 15 and 100 splits per node.

In order to compare the proposed method, the following methods are explained and applied to some or all datasets. The first five results are obtained from five methods that recognize human-human actions on the *UT1* and *UT-Interactions dataset set 2 (UT2)* datasets. These five methods ([81], [77], [82], [83] and [84]) were explained in Section 2.2. Note that these methods consider the 6 classes of UT-Interactions. Therefore, an approximation is performed to cluster the 6 classes in 2 new classes (fight and non-fight). Their confusion matrices have been used to estimate the results. Furthermore, the method [18], which was also explained in Section 2.2, is used on *Movies* dataset.

The three tested methods are: the ViF, the LMP method and the method [48] that were explained in Section 2.2. The LMP method needs fixed-size histograms to extract the features. The best results are obtained using 6 bins. These three methods have been used with RF classifier because it is more akin to the classifier used and usually gets the best accuracy.

These nine related methods are now compared with the proposed method on some or all datasets, see Table 5.1. Note that the results in the first column of Table 5.1 were obtained with the test part of the *UT1* dataset, which is disjunct from the validation part mentioned in Section 5.3.1.

Table 5.2 shows the number of features used for classification and the computational cost (for feature extraction). These results show that the proposed method is not the fastest one, although the difference is very small compared to the fastest [48]. However, the accuracy is significantly better. Note again that the proposed method does not give a fixed number of STEC trajectories per sequence, because this method obtains the STEC trajectories (features) where motion occurs. Therefore it depends on the input sequence.

The STEC trajectories method is a simple yet efficient tool for violence or fight recognition. The accuracy has been compared to the state-of-the-art methods with four datasets containing violence actions. Accuracy was significantly higher in the four compared datasets. The boost in accuracy can be explained by the ability of STEC trajectories to capture the different motion parts in space-time.

Method	UT1	UT2	Movies	Hockey
Vahdat [81]	95%	96.7%	-	-
Waltisberg [77]	96.7%	91.7%	-	-
Nebel [82]	93.4%	-	-	-
Mukherjee [83]	91.7%	83.4%	-	-
Zhang [84]	98.3%	-	-	-
Mohammadi [18]	-	-	96.9%	-
ViF+RF	81.7%	78.3%	88.9%	82.4%
LMP+RF	60%	53.3%	92%	77.7%
Serrano [48]+RF	86.7%	91.7%	97.8%	82.4%
Proposed	**98.6%**	**99.5%**	**98%**	**82.6%**

Table 5.1: The nine related method compared with the proposed method on some or all datasets. Note that the first five methods consider the 6 classes of UT-Interactions. Here, an estimation has been used by clustering the 6 classes in the 2 new classes using their confusion matrices.

Method	Features/sequence	Msecs/frame
ViF	96000	454.5
LMP	10368	151.6
Serrano [48]	2940	26.5
Proposed	-	61.8

Table 5.2: Feature extraction times. Average times measured on *UT1* dataset, on an Intel Xeon computer with 2 processors at 2.90Ghz.

Each STEC trajectory models a part of the action and it points to the action center in space-time. Additionally, the Hough Forests classifier does not lose temporal-transition information that is considered useful. Another positive aspect is the low extraction times, which can be explained because a sparse sampling is used to represent the actions. This is in contrast with most state-of-the-art methods, which need a dense sampling to represent the actions. Again, efficiency in this task is paramount.

5.4 Discussion

The Spatio-Temporal Elastic Cuboid (STEC) trajectories descriptor is a novel proposed descriptor to model the different parts in motion in fight or non-fight sequences. These STEC trajectories capture the relative position, size, orientation and shape of the movements. The descriptor is robust to scale, orientation and position changes due to the use of relative measures between consecutive frames. Furthermore, this method does not need a person detector to find the actors as other *state-of-the-art* methods because the whole sequence is used. Also, the method does not use trackers that are generally computationally expensive and may require manual initialization. The proposed method does not use a BoW method to cluster the trajectories (features), for the adapted Hough Forests classifier is used to keep the temporal order of the action.

The proposed method has been compared with nine other methods in some or all of the four datasets. Experiments show that the method is better than the related methods in the four datasets. Although it is not the fastest one, the trade-off between computational time and accuracy is clearly better. This method gives an accuracy between 98% and 99.5% in *UT1*, *UT2* and *Movies* datasets. The accuracy in the *Hockey* dataset is 82.5%.

While other methods tackle a more general problem (such as action recognition) or resort to computationally intensive optical flow computations, the proposed method opens up the possibility of practical implementations.

CHAPTER 6

Method 4: 2D Convolutional Neuronal Network

"Handcrafted" spatio-temporal features that account for both motion and appearance information can achieve high accuracy rates, albeit the computational cost of extracting some of those features is still prohibitive for practical applications. The deep learning paradigm has been recently applied for the first time to this task too, in the form of a 3D Convolutional Neural Network 3D-CNN (see Figure 2.26) that processes the whole video sequence as input. However, results in human perception of other's actions (and results with the proposed method was described in previous chapters) suggests that, in this specific task, motion features are crucial. This means that using the whole video as input may add both redundancy and noise in the learning process. In this chapter, a hybrid handcrafted/learned feature framework is proposed which provides better accuracy than the previous feature learning method, with similar computational efficiency.

This chapter is organized as follows. Section 6.1 gives a overview of Deep Learning. Section 6.2 describes the feature extraction step. Section 6.3 explains the Convolutional Neuronal Network used. Section 6.4 provides a comparison of the results with other related state-of-the-art methods. Finally, in Section 6.5 the main conclusions are outlined.

6.1 Overview of Deep Learning

Before presenting the proposed method in this chapter, an overview of deep learning and convolutional neuronal networks will be given (based on [85, 86, 87]).

Deep learning is a branch of machine learning based on a set of algorithms that attempt to model high-level abstractions in data by using multiple processing layers, with complex structures or otherwise, composed of multiple non-linear transformations.

Deep learning is part of a broader family of machine learning methods based on learning representations of data. An observation (e.g., an image) can be represented in many ways such as a vector of intensity values per pixel, or in a more abstract way as a set of edges, regions of particular shape, etc. Some representations are better than others at simplifying the learning task (e.g. face recognition or facial expression recognition) from examples. One of the promises of deep learning is replacing handcrafted features with efficient algorithms for unsupervised or semi-supervised feature learning and hierarchical feature extraction.

Research in this area attempts to make better representations and create models to learn these representations from large-scale unlabeled data. Some of the representations are inspired by advances in neuroscience and are loosely based on interpretation of information processing and communication patterns in a nervous system, such as neural coding, which attempts to define a relationship between various stimuli and associated neuronal responses in the brain.

There are many variants of deep architectures. Most of them are branched from some original parent architectures. It is not always possible to compare the performance of multiple architectures, because they are not all evaluated on the same data sets. Deep learning is a fast-growing field, and new architectures, variants, or algorithms appear every few weeks. The prevalent architecture for processing visual data is convulutional neuronal networks (CNN).

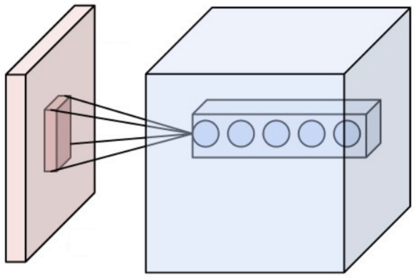

Figure 6.1: A neuron of a convolution layer connected to their receptive field (taken from [30]).

A CNN is a type of feed-forward artificial neural network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex, whose individual neurons are arranged in such a way that they respond to overlapping regions tiling the visual field. CNN is inspired by biological processes and are variations of multilayer perceptrons designed to use minimal amounts of preprocessing. Applications in image and video recognition, recommender systems and natural language processing are widely used.

A CNN architecture is formed by a stack of distinct layers that transform the input volume into an output volume (see Figure 6.1) through a differentiable function. A few distinct types of layers are commonly used. The more important concepts of a CNN architecture are shown below:

- **Convolution Layer.** The Convolutional layer is the core building block of a CNN. The layer's parameters consist of a set of learnable filters (or kernels), which have a small receptive field, but extend through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product between the entries of the filter and the input and producing a 2-dimensional activation map of that filter. As a result, the network learns filters that activate when they see some specific type of feature at some spatial position in the input.

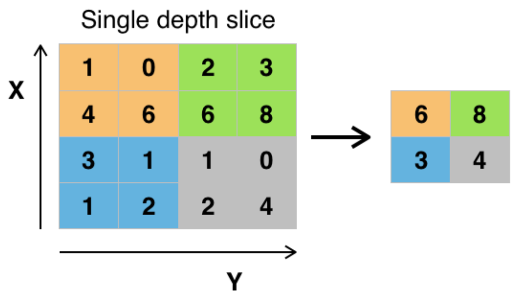

Figure 6.2: Max pooling with a 2 x 2 filter (taken from [30])

- **Pooling Layer.** An important concept in CNN is pooling, which is a form of non-linear down-sampling. There are several non-linear functions to implement pooling among which max pooling is the most common. It partitions the input image into a set of non-overlapping rectangles and, for each such sub-region, the maximum is computed. The intuition is that once a feature has been found, its exact location is not as important as its rough location relative to other features. The function of the pooling layer is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control overfitting. It is common to insert a pooling layer in-between successive convolution layers in a CNN architecture. The pooling operation provides a form of translation invariance, see an example in Figure 6.2.

- **ReLu Layer.** ReLU is the abbreviation of Rectified Linear Units. This is a layer of neurons that applies the non-saturating activation function $f(x) = max(0, x)$. It increases the nonlinear properties of the decision function and of the overall network without affecting the receptive fields of the convolution layer.

- **Fully Connected Layer.** Finally, after several convolutional and max pooling layers, the high-level reasoning in the neural network is achieved via fully connected layers. Neurons in a fully connected layer have full connections to all activations in the previous layer, as seen in regular Neural Networks. Their

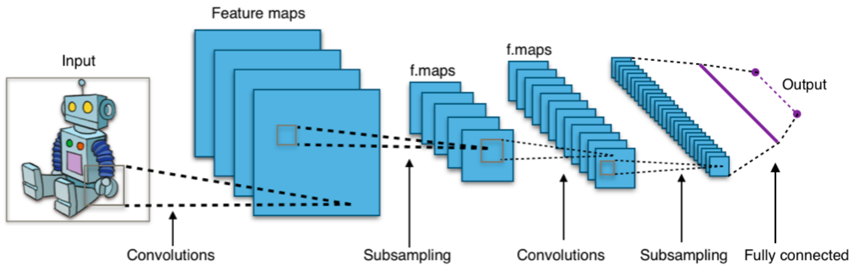

Figure 6.3: A typical CNN architecture

activations can hence be computed with a matrix multiplication followed by a bias offset.

- **Loss Layer.** The loss layer specifies how the network training penalizes the deviation between the predicted and true labels and is normally the last layer in the network. Various loss functions appropriate for different tasks may be used there. Softmax loss is used for predicting a single class of K mutually exclusive classes.

A typical CNN architecture for image recognition is shown in Figure 6.3.

The proposed method is summarized in the following two steps, see Figure 6.4:

1. The feature extraction step aims to obtain a representative image from each input video sequence.

2. A 2D CNN is used to classify the representative image and obtain the final decision for the sequence.

The main step of feature extraction is performed as follows. Both motion and appearance in the video sequence may be encoded in a representative input image. To compute this image, a technique similar to motion history images [88] is used, but applied to the raw frames instead of motion images. Frames are accumulated with a

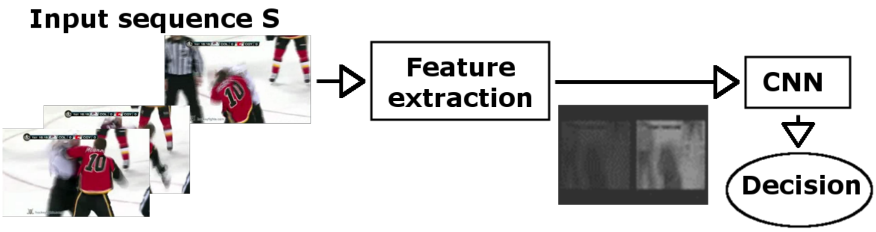

Figure 6.4: General diagram of the proposed method.

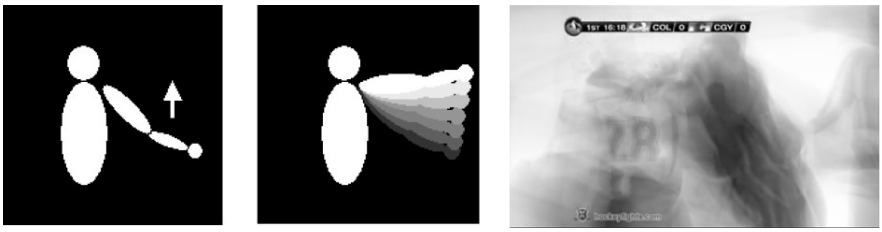

Figure 6.5: Left: example input sequence in which an individual moves his arm. Middle: a motion history image calculated from the input sequence. Right: a motion history image calculated from a *Hockey* fight sequence.

weight related to their temporal position (frame number in our case), see two examples in Figure 6.5. Then, a temporal-voting model is also used to obtain a weight for each class for these motion history images.

The resulting image is a concise representation of both appearance and motion throughout the video. Such representation can be also computed efficiently since it is readily parallelizable (all operations can be performed frame-by-frame). Given a frame sequence as:

$$S = \{I(1), I(2), \cdots, I(t), \cdots, I(T)\} \tag{6.1}$$

where $I(t)$ represents frame t and T is the number of frames for this sequence S. The result representative image F is obtained by concatenating the following two images for each sequence as:

$$F = P' \| P \tag{6.2}$$

where

$$P = \frac{\sum_{t=1}^{T} t \cdot n(t) \cdot I(t)}{N} \tag{6.3}$$

and

$$P' = \frac{\sum_{t=1}^{T} t \cdot n'(t) \cdot I(t)}{N} \tag{6.4}$$

where N is a normalization factor defined as:

$$N = \sum_{t=1}^{T} t \cdot 255 \tag{6.5}$$

The $n(t)$ and $n'(t)$ values are obtained from the temporal-voting Hough Forests model for the sequence S that correspond for each class (this will be explained below). Furthermore, these parameters have been normalized as:

$$n(t) = \frac{k(t)}{m} \tag{6.6}$$

and

$$n'(t) = \frac{k'(t)}{m} \tag{6.7}$$

where

$$m = \max\{\max_t k(t), \max_t k'(t)\} \tag{6.8}$$

The values $k(t)$ and $k'(t)$ are also obtained from the temporal-voting Hough Forests model that correspond with each class, specifically fight and non-fight classes, respectively. These values aim to weight the action distribution in time, i.e. modeling where each action is more likely to occur.

Finally, each generated F is a single 2D image which will be used as input a 2D Convolutional Neural Network.

The main steps are next described in more detail:

1. The feature extraction step.
   (a) Features are extracted for each sequence.
   (b) Apply a Hough Forests classifier to the previously extracted features to create a temporal-voting model of action distribution in time. This will provide $n(t)$ and $n'(t)$, i.e. Ecs. (6) and (7).
   (c) Combine the previous temporal-voting model with the input sequence to create an image representative of the video sequence, as shown in Ecs. (2), (3) and (4).
2. Use a 2D CNN to process the representative image and obtain the final decision for the sequence.

These four steps are shown in Figure 6.6 and further described in what follows.

6.2 Feature Extraction

The feature extraction step aims to obtain a representative image from each input video sequence that correspond with the following three sections.

6.2.1 Extract Spatial Features

The first step (1.a) is to extract spatial features from each sequence. The BRISK descriptor is used to extract features [89]. The BRISK descriptor extracts local image features in space characterized by a high variation of the image values. It uses a scale-space detector in combination with the assembly of a bit-string descriptor from intensity comparisons retrieved by dedicated sampling of each

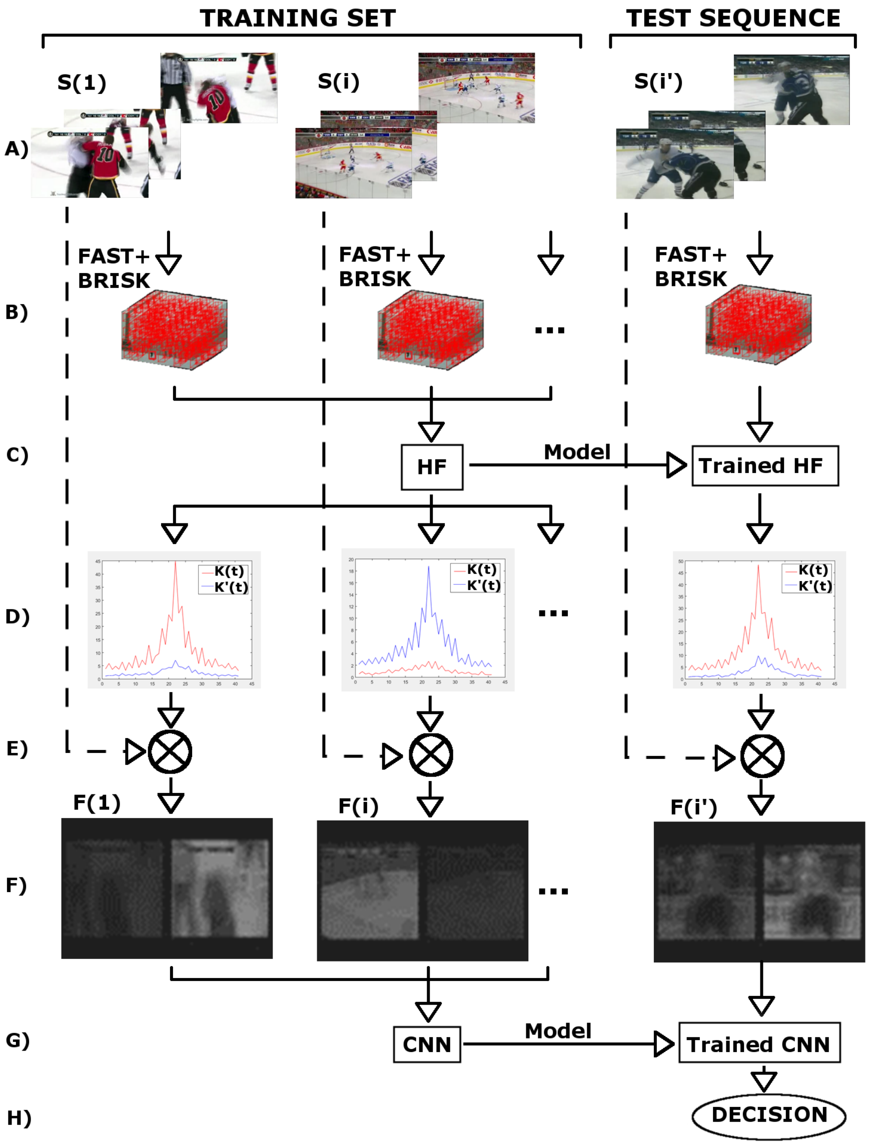

Figure 6.6: Detailed diagram of the proposed method. From top to bottom rows: A) Input sequences. B) FAST detector and BRISK features (step 1.a). C) Hough Forests (HF) classifier and D) temporal-voting models (step 1.b). E) Combination step and F) representative images (step 1.c). G) CNN with the H) final decision (step 2).

keypoint neighborhood. This method is rotation and scale invariant and very efficient.

Descriptors may be used with dense or sparse sampling. Sparse sampling is used along with the salient spatial keypoints and is computationally more efficient. The keypoints are obtained using the FAST detector [90]. It requires consecutive pixels with are sufficiently brighter or darker than the central pixel to select it. Also, the FAST detector is computationally very fast.

To extract spatial features, the FAST detector is applied to each frame of the sequence. For these located keypoints in space, the BRISK features are extracted.

6.2.2 Hough Forests

Hough Forests (step 1.b) classifier was explained in detail in Section 5.2.

The Hough Forests classifier creates a Hough Voting Space when a new sequence (set of features) is tested. Each feature produces a vote in space-time and class. Besides, the accumulation of votes is calculated for each time step, see Figure 6.7, so it is possible to know when a fight/non-fight action ($k(t)$ and $k'(t)$, respectively) occurs. This is the temporal-voting model that will be used for the following step. Really, it is possible to make a final decision from this temporal-voting model for the sequence. However the accuracy is still worse than the proposed method as it will be shown in Section 6.4.

The original output of Hough Forests is a voting volume with the space-time probabilities for each class. The local maximum can be calculated by applying a Parzen estimator with a Gaussian kernel. The local maximum encodes the class, position and length of the action. Then, the temporal-voting model may be generated from the accumulation of these probabilities at time. The summation is preferred over multiplication due to better stability, as shown in [78].

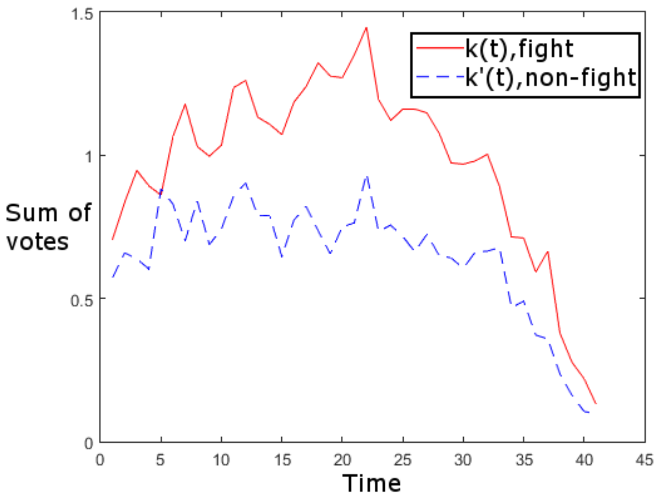

Figure 6.7: The sum votes in space for each time step, the temporal-voting model for each class, for a *Hockey* fight sequence.

6.2.3 Representative Image

Finally (step 1.c), the previous temporal-voting model is combined with each input sequence, as it was explained before, to create a representative image for each video sequence.

6.3 2D Convolutional Neuronal Network

Images computed as described above do not appear natural, see Figure 6.8, although both appearance and patterns of motion are recognizable. These images embed useful discriminant information that the convolutional network, when properly dimensioned and trained, can leverage. While the whole input sequence matrix contains complete appearance and motion information, the images proposed here are a better representation of motion than appearance.

Note that the proposed method bears some resemblance with the method [46] described above. Fuzzy regions caused by strong motion

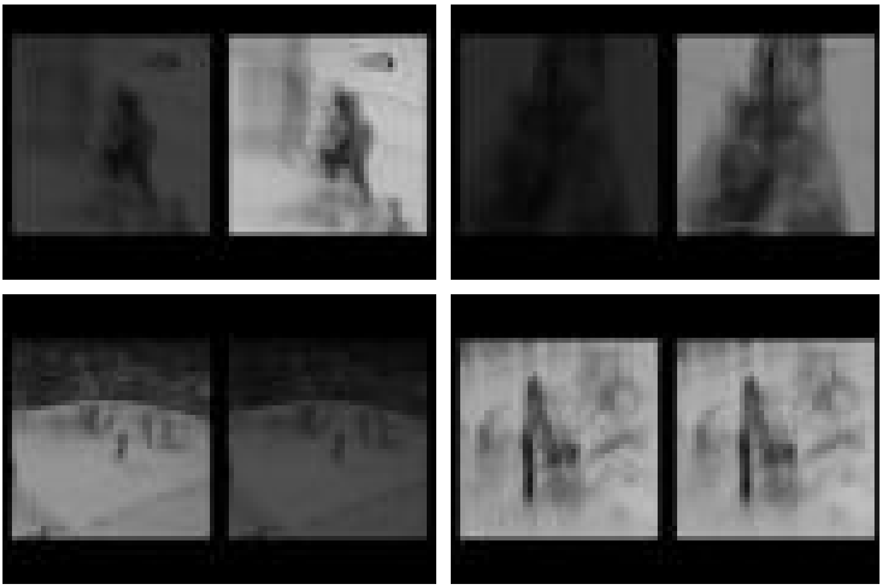

Figure 6.8: Representative $F(s)$ images obtained from four *Hockey* video sequences. Top row: two *Hockey* fight sequences. Bottom row: two non-fight sequences.

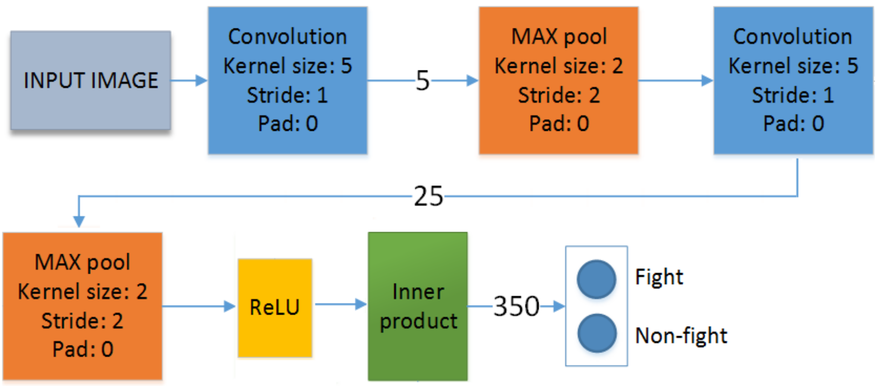

Figure 6.9: The CNN architecture.

were the main source of information in [46]. In the proposed method, motion gives rise to fuzzy regions in the input image.

The neural network contains two convolutional layers, see Figure 6.9, each followed by a MAX pooling layer. This results in a reduced-resolution output feature map which is robust to small variation in the location of features in the previous layer. Additionally, a fully-connected Inner Product layer is used followed by the Rectified Linear Units (ReLU) operation, which adds non-linearity.

6.4 Experiments

The proposed method is assessed using two different datasets: the *Movies* and *Hockey* datasets and compared to other related methods.

Since the proposed method is based on strong motion patterns, energetic actions such as those in the above-mentioned *Hockey* dataset are particularly challenging. The proposed method uses the following parameters for each step. The BRISK descriptor is used with the standard parameters used by BRISK authors in [89]. The FAST descriptor is also used with with the standard parameters used in [90]. The Hough Forests classifier uses the following parameters: 10 trees, 15 depth for each tree and 2000 tests per node in the training step [78]. Finally, the CNN used frames of size 60×90 pixels. This was done in order to allow for a better comparison with [27], in which frames were also resized to 60×90 frames. The network of Figure 6.9 was trained with a batch size of 64 and a maximum of 100 iterations. Optimization was done using stochastic gradient descent (base learning rate=0.01, momentum=0.9, weight decay factor=0.0005). The learning rate was subject to inverse decay and a SoftMax loss function was used. The method described in [91] ("Xavier's method") was used to fill the network weights prior to start of training.

Figure 6.10 shows an example of the network features and layer outputs. The first layer of features represent a mixture of both appearance and motion.

The proposed method was compared with both "handcrafted" feature methods and with two works that applied deep feature learning, reviewed in Section 2.2. With regards to the "handcrafted feature" category the ViF method [9], and the more recent methods [46] and [48], all mentioned in Section 2.2 are considered. Moreover, we consider the LMP method [65] where the best results were obtained using 6 bins. For this "handcrafted" feature category three different state-of-the-art classifiers were used: SVM (linear kernel), AdaBoost (100 classifiers, weighted voting) and RF (50 trees). See Table 6.1.

On the other hand, method [27] was applied on the Hockey dataset, see Table 6.3. The C3D method [28] was applied to both datasets. The method achieved the best results with frames of size 100×120 pixels and a maximum of 2000 iterations. Optimization was done using stochastic gradient descent (base learning rate=0.0001, momentum=0.9, weight decay factor=0.005). Finally, we have tested BRISK descriptor with Hough Forest classifier for both datasets with the same parameters that the proposed method. Averages of 5 repetitions of 10-fold cross-validation are reported. See the results on Table 6.1 with the accuracies on the two datasets considered.

Overall the proposed method gives the best accuracies in both datasets. To put the results in context, Table 6.2 shows other accuracies obtained in the state-of-the-art *Movies* dataset. The proposed method provides the best absolute accuracy with 99$\%$.

Table 6.3 shows other accuracies obtained in the literature on the reference *Hockey* dataset. The best absolute accuracy reported on this

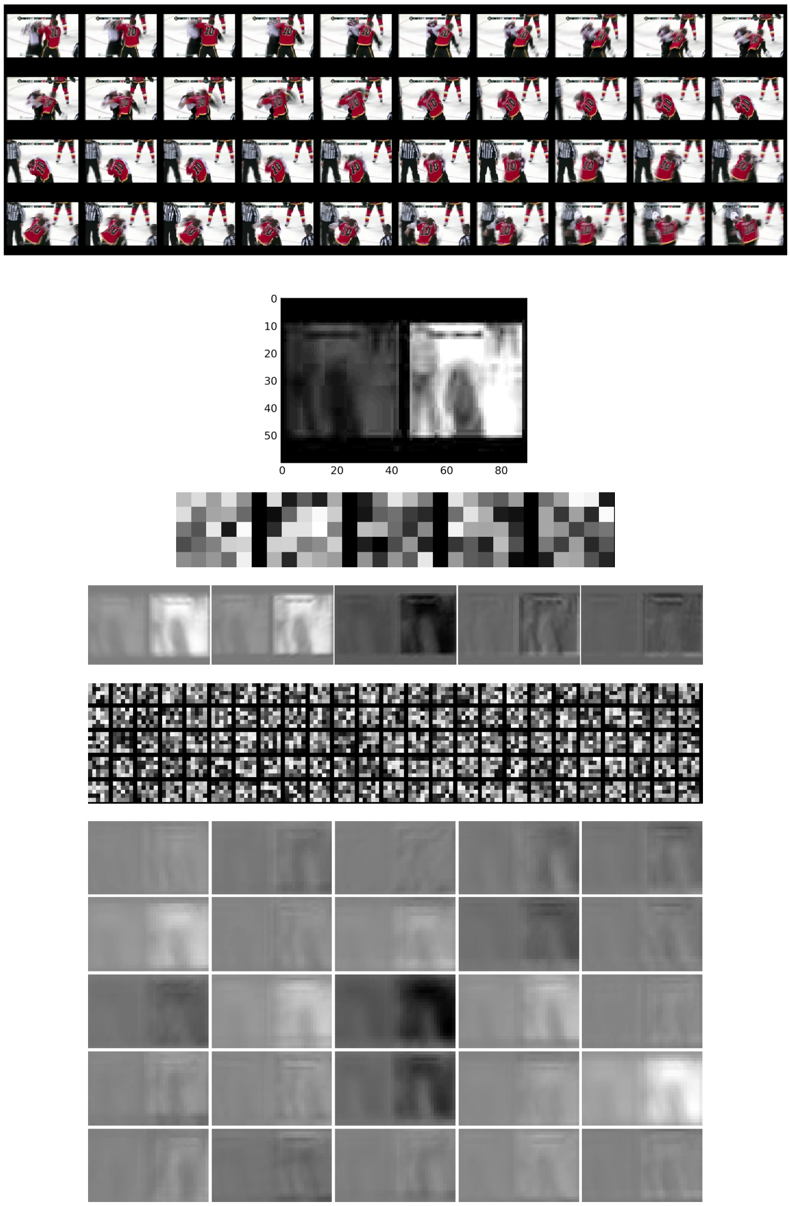

Figure 6.10: Network layers after processing a *Hockey* fight input image (computed as shown in the previous section). From top to bottom: input sequence, input image, convolution layer 1 features, convolution layer 1 outputs, convolution layer 2 features, convolution layer 2 outputs.

Features	Classifier	Movies	Hockey
ViF [9]	SVM	96.7±0.3%	86.3±0.2%
	AdaBoost	92.8±0.4%	84.2±0.3%
	RF	88.9±1.2%	82.4±0.6%
LMP [65]	SVM	84.4±0.8%	75.9±0.3%
	AdaBoost	81.5±2.1%	76.5±0.9%
	RF	92±0.1%	77.7±0.6%
Deniz [46]	SVM	85.4±9.3%	90.1±0%
	AdaBoost	98.9±0.2%	90.1±0%
	RF	90.4±3.1%	61.5±6.8%
Serrano [48]	SVM	87.2±0.7%	72.5±0.5%
	AdaBoost	81.7±0.2%	71.7±0.3%
	RF	97.8±0.4%	82.4±0.6%
C3D [28]		93.6±0.8%	87.4±1.2%
FAST+BRISK	Hough Forests	95.5±0.6%	90.3±0.7%
Proposed method		**99±0.6%**	**94.5±0.7%**

Table 6.1: Accuracies and standard deviation on the two datasets is considered. The best average accuracies obtained on each dataset are shown in bold.

Method	Features/Classifier	Type	Acc.
Bermejo [36]	STIP, BoW, SVM	HC	89.5%
Deniz [46]	histograms of frequency-based motion intensities, AdaBoost	HC	98.9%
Serrano [48]	motion blobs, RF	HC	96.9%
Mohammadi [18]	OF, substantial derivative, BoW, SVM	HC	97.8%

Table 6.2: Results reported in the literature on the *Movies* dataset, sorted by year of publication (oldest at the top). Types: HC=handcrafted features, DL=deep learning.

Method	Features/Classifier	Type	Acc.
Wang [60]	dense optical flow, BoW, SVM	HC	74.1%
Bermejo [36]	STIP, BoW, SVM	HC	91.7%
Huang [55]	variance of optical flow, SVM	HC	86.9%
Ding [27]	3D-CNN	DL	91.0%
Xu [16]	MoSIFT, kernel density estimation, sparse coding	HC	94.3%
Deniz [46]	histograms of frequency-based motion intensities, AdaBoost	HC	90.1%
Serrano [48]	motion blobs, RF	HC	82.4%
Senst [20]	modified SIFT, BoW, SVM	HC	93.3%

Table 6.3: Results reported in the literature on the reference *Hockey* dataset, sorted by year of publication (oldest at the top). Types: HC=handcrafted features, DL=deep learning.

dataset was 94.3%, using MoSIFT as the main descriptor. However the best absolute accuracy, by a slight margin, was obtained with the proposed method.

In Table 6.4 we compare the times required to process an input sequence of the reference *Hockey* dataset with other state-of-the-art methods. This comparative was carried out using only CPU, however the method can be executed in GPU. The proposed method is fast enough for practical application.

Overall the proposed method provides state-of-the-art accuracies, specifically improving upon the previous use of deep learning in this task. Besides, the framework of the proposed convolutional network and its handcrafted inputs is efficient.

6.5 Discussion

The task of fight recognition has attracted interest of many researchers in the last few years. “Handcrafted” features, some used for generic action recognition work, were used for this task before

Method	Time	Conditions
MoSIFT [16]	43.22 s	Time to extract features, this does not include time for feature reduction and/or classification
ViF [9]	16.27 s	-
LMP [65]	5.43 s	-
Deniz [46]	1.51 s	-
Serrano [48]	0.95 s	-
3D-CNN [27]	0.19 s	-
C3D [28]	0.41 s	-
Proposed	0.46 s	Took 151.2, 105.3, 67.4 and 138.2 ms for each step (the first and latter parts the codes were interpreted)

Table 6.4: Time required to process an input sequence of the reference Hockey dataset using an Intel i7 3.4Ghz processor.

the advent of feature learning methods. The latter methods, represented by the powerful convolutional neural networks, automatically learn features for the task at hand. The two compared works in this line used a 3D convolutional neural network that processes the entire video sequence as input. Inspired by psychophysics experiments suggesting that motion features may be more important for this specific task, this chapter proposed a hybrid approach. The rich temporal-voting information from Hough Forests classifier is used, which is fed with BRISK features that capture motion and appearance from a video sequence. Finally, a much simpler 2D convolutional neural network is fed with the "handcrafted" images. The proposed method demonstrated superiority over both handcrafted feature methods and the previous 3D convolutional networks. Besides, the framework is computationally very efficient.

CHAPTER 7

CONCLUSIONS

7.1 Conclusions

Whereas action detection is an important topic in computer vision, the detection of particular events such as violence, aggression or fights, has received less attention. These events may be extremely useful in several video surveillance scenarios such as prisons, psychiatric wards, or even in camera smartphones.

In this PhD thesis, a detailed review of state-of-the-art violence datasets has been carried out. A dataset division is presented and these datasets are compared. Eight public datasets have been reviewed. A dataset ranking according to the estimated number of papers which directly references for each dataset is also shown. On the other hand, a study of state-of-the-art methods for violence detection has been carried out. These methods are clustered into the following approaches: spatial and spatio-temporal descriptors, optical flow, trajectories, deep learning, use of audio and others. Finally, state-of-the-art methods that use public datasets are compared by year and classification accuracy.

In addition, four novel methods for detecting fights in video are introduced.

The first method detects motion blobs and then calculates different features to characterize them. The method makes no assumptions on number of individuals (it can be also used to detect vandalism), body part detection or salient point tracking. Experiments show that the method does not outperform the best methods considered. However, it is much faster while still maintaining useful accuracies ranging from 70% to near 98% depending on the dataset. The method was assessed using five different datasets and compared with five other methods. While other methods tackle a more general problem (such as action recognition) or resort to computationally intensive optical flow computations, the proposed method opens up the possibility of practical implementations.

The second method is based on the observation that kinematic information suffices for human perception of other's actions. The method uses acceleration/deceleration patterns as the main discriminant feature. The method shows promising features for surveillance scenarios and it also performs relatively well when considering challenging actions such as those that occur in sports. The proposed method was assessed using three different datasets and compared with five other methods. Accuracy improvements of up to 8% with respect to state-of-the-art violence detection techniques were reported. It is hypothesized that when motion is sufficient for violence detection, appearance not only takes significant additional computation but it also may confuse the detector. The method makes no assumptions on number of individuals (it can be also used to detect vandalism), body part detection or salient point tracking. Additionally, it is faster than other methods and uses fewer features, which opens up the possibility of practical implementations.

The third method develops a novel descriptor, called Spatio-Temporal Elastic Cuboid (STEC) Trajectories, to model the different parts in motion in fight or non-fight sequences. STEC trajectories capture the relative position, size, orientation and shape of the movements. This descriptor is robust to scale, orientation and position changes due to the use of relative measures between consecutive frames. Furthermore, this descriptor does not need a person detector to find the actors as other *state-of-the-art* methods because the whole sequence is used. Also, the method does not use trackers that are generally computationally expensive and may require manual initialization. The proposed method does not use a BoW method to cluster the trajectories (features), for the adapted Hough Forests classifier is used to keep the temporal order in the action. The method was compared with nine other methods in four datasets. Experiments showed that the method is better than the related methods. While it is not the fastest one, the trade-off between computational time and accuracy is clearly better. This method gives an accuracy between 98% and 99.5% in *UT1*, *UT2* and *Movies* datasets. The accuracy in the *Hockey* dataset is 82.5%.

The last method uses a hybrid approach. "Handcrafted" features are typically used for generic action recognition. These features were used for this task before the advent of feature learning methods. The latter methods, represented by the powerful convolutional neural networks, automatically learn features for the task at hand. The proposed method extracts rich temporal-voting information from a Hough Forests classifier, which is fed with BRISK features that capture motion and appearance from a video sequence. Then, a simple 2D convolutional neural network is fed with the "handcrafted" images. The method was compared with six other methods in two datasets. The proposed method demonstrated superiority over compared handcrafted feature methods and two previous 3D convolutional networks. Besides, the framework is computationally very efficient.

Table 7.1 shows a summary of the introduced methods.

7.2 Scientific Publications

7.2.1 Journals

- M. M. Fernández-Carrobles, **I. Serrano**, G. Bueno and O. Deniz. Bagging Tree Classifier and Texture Features for Tumor Identification in Histological Images. Procedia Computer Science Journal. Volume 90, pp. 99-106. July, 2016.

Method	Datasets	Methods	Accuracy	Efficiency
Method 1, Chapter 3	3	5	✗	✓✓
Method 2, Chapter 4	3	5	✓	✗
Method 3, Chapter 5	3	9	✓	✗
Method 4, Chapter 6	2	6	✓✓	✗

Table 7.1: A comparison of the methods introduced in this PhD Thesis. Column Datasets refers to number of used datasets, Methods refers to number of compared methods, Accuracy and Efficiency indicate when the method improves or decreases these measures.

- **I. Serrano**, O. Deniz, G. Bueno. Survey of Violence Detection Methods. International Journal of Computer Vision. July, 2016. Submitted.
- **I. Serrano**, O. Deniz, J. L. Espinosa-Aranda, G. Bueno, D. Moloney, T. Llewellynn and A. Pagani. Fight Recognition in video using Hough Forests and 2D Convolutional Neural Networks. International Journal of Computer Vision. July, 2016. Submitted.
- **I. Serrano**, O. Deniz, G. Bueno, G. Garcia-Hernando and T.K. Kim. Spatio-Temporal Elastic Cuboid Trajectories for Efficient Fight Recognition using Hough Forests. Machine Vision and Applications. February, 2016. Submitted.
- **I. Serrano**, O. Deniz, G. Bueno and T.K. Kim. Fast fight detection. PLoS ONE, 10(4). April, 2015.

7.2.2 Books

- O. Deniz, **I. Serrano**, J. Salido, N. Vállez, G. Bueno and J. L. Espinosa. Learning image processing with OpenCV. PACKT OpenSource. April, 2015.
- O. Deniz, **I. Serrano**, J. Salido, N. Vállez, J. Patón, G. Bueno and M.M. Fernández-Carrobles. OpenCV Essentials. PACKT OpenSource. August, 2014.

7.2.3 International conferences

- **I. Serrano**, A. Pedraza, M.M. Fernández-Carrobles, M. Garcia-Rojo, L. Gonzalez and G. Bueno. Deep Learning Approach for Automatic Scoring HER2 Whole Slide Images. SPIE Medical Imaging (MI17). Orlando, FL, United States. July, 2016. Submitted.

- M.M. Fernández-Carrobles, **I. Serrano**, A. Pedraza, M. Garcia-Rojo, L. Gonzalez and G. Bueno. Scoring Her2 WSI based on Colour Quantification. SPIE Medical Imaging (MI17). Orlando, FL, United States. July, 2016. Submitted.

- A. Pedraza, **I. Serrano**, M.M. Fernández-Carrobles and G. Bueno. HER2 Contest, Results from VISILAB Group. The Pathological Society of Great Britain and Ireland. University of Nottingham, UK. July, 20016. Winners of the Challenge.

- G. García-Hernando, H. Jin-Chang, **I. Serrano**, O. Deniz and H.K. Kim. WACV 2016: IEEE Winter Conference on Application of Computer Vision. Transition Hough Forests for Trajectory-based Action Recognition. Lake Placid, NY, USA. March, 2016.

- M. M. Fernández-Carrobles, **I. Serrano**, O. Deniz and G. Bueno. XIII International Symposium on Biomedical Imaging. Camelyon Challenge - Results from VISILAB group. Prague, Czech Republic. April, 2016.

- O. Deniz, **I. Serrano**, G. Bueno and T. K. Kim. Fast Violence Detection in Video. In The 9th International Conference on Computer Vision Theory and Applications (VISAPP). Lisbon, Portugal. January, 2014.

- **I. Serrano**, O. Deniz and G. Bueno. VISILAB at MediaEval 2013: Fight Detection. In MediaEval 13. Barcelona, Spain. October, 2013.

7.2.4 National conferences

- J. L. Espinosa, D. Aguado, O. Deniz, J. Salido, G. Bueno, N. Vállez, **I. Serrano** and M. M. Fernández. Sistema automatico para la medida de la talla del pie según parámetro del dispositivo Brannnock. XXXVI Jornadas Nacionales de Automática. Comité Español de Automática (CEA). País Vasco, Spain. September, 2015.

- **I. Serrano**, G. Bueno, J. Salido and O. Deniz. ASM Facial Combinando Imagen Visible y de Profundidad. XXXII Jornadas Nacionales de Automática. Comité Español de Automática (CEA), Sevilla, Spain. September, 2011.

7.2.5 Visiting researcher

- Department of Electrical and Electronic Engineering, Imperial College, London, UK. From February 1^{st} to May 1^{st} 2015. Supervisor: Tae-Kyun Kim, PhD, Lecturer.

BIBLIOGRAPHY

[1] Wang, L.: Information Technology. http://wangli.info/information-technology-it.html [Online; Accessed: 23-June-2016].

[2] Service Representaciones SL.: Service Representaciones. http://www.servicerepresentaciones.com/ [Online; Accessed: 29-June-2016].

[3] NovoSun Technology Limited 2016: NovoSum CyeWeb. http://www.novosun.com/index.php/en/nvr-software [Online; Accessed: 23-June-2016].

[4] Fisher, R., Santos-Victor, J., Crowley, J.: CAVIAR: Context aware vision using image-based active recognition. http://homepages.inf.ed.ac.uk/rbf/CAVIAR/ (2005) [Online; Accessed: 30-June-2016].

[5] Fisher, R.: BEHAVE: Computer-assisted prescreening of video streams for unusual activities. EPSRC project GR S **98146** (2004)

[6] Ryoo, M.S., Aggarwal, J.K.: UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA). http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html (2010) [Online; Accessed: 30-June-2016].

[7] Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)

[8] Demarty, C.H., Penet, C., Gravier, G., Soleymani, M.: The MediaEval 2012 affect task: violent scenes detection. In: Working Notes Proceedings of the MediaEval 2012 Workshop. (2012)

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

[10] Zajdel, W., Krijnders, J.D., Andringa, T., Gavrila, D.M.: CASSANDRA: audio-video sensor fusion for aggression detection. In: Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on, IEEE (2007) 200–205

[11] Gong, Y., Wang, W., Jiang, S., Huang, Q., Gao, W.: Detecting violent scenes in movies by auditory and visual cues. In: Advances in Multimedia Information Processing-PCM 2008, Springer (2008) 317–326

[12] Giannakopoulos, T., Pikrakis, A., Theodoridis, S.: A multimodal approach to violence detection in video sharing sites. In: Pattern Recognition (ICPR), 2010 20th International Conference on, IEEE (2010) 3244–3247

[13] Ionescu, B., Schlüter, J., Mironica, I., Schedl, M.: A naive mid-level concept-based fusion approach to violence detection in Hollywood movies. In: Proceedings of the 3rd ACM conference on International conference on multimedia retrieval, ACM (2013) 215–222

[14] de Souza, F.D., Chávez, G.C., do Valle, E.A., de A Araujo, A.: Violence detection in video using spatio-temporal features. In: Graphics, Patterns and Images (SIBGRAPI), 2010 23rd SIBGRAPI Conference on, IEEE (2010) 224–230

[15] Esen, E., Arabaci, M.A., Soysal, M.: Fight detection in surveillance videos. In: Content-Based Multimedia Indexing (CBMI), 2013 11th International Workshop on, IEEE (2013) 131–135

[16] Xu, L., Gong, C., Yang, J., Wu, Q., Yao, L.: Violent video detection based on MoSIFT feature and sparse coding. In: Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, IEEE (2014) 3538–3542

[17] Ribeiro, P.C., Audigier, R., Pham, Q.C.: RIMOC, a feature to discriminate unstructured motions: Application to violence detection for video-surveillance. Computer Vision and Image Understanding **144** (2016) 121–143

[18] Mohammadi, S., Kiani, H., Perina, A., Murino, V.: Violence detection in crowded scenes using substantial derivative. In: Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on, IEEE (2015) 1–6

[19] Zhang, T., Jia, W., Yang, B., Yang, J., He, X., Zheng, Z.: MoWLD: a robust motion image descriptor for violence detection. Multimedia Tools and Applications (2015) 1–20

[20] Senst, T., Eiselein, V., Sikora, T.: A local feature based on Lagrangian measures for violent video classification. In: Imaging for Crime Prevention and Detection (ICDP-15), 6th International Conference on, IET (2015) 1–6

[21] Zhao, Y., Qiao, Y., Yang, J., Kasabov, N.: Abnormal Activity Detection Using Spatio-Temporal Feature and Laplacian Sparse Representation. In: Neural Information Processing, Springer (2015) 410–418

[22] Zhang, T., Yang, Z., Jia, W., Yang, B., Yang, J., He, X.: A new method for violence detection in surveillance scenes. Multimedia Tools and Applications (2015) 1–23

[23] Gao, Y., Liu, H., Sun, X., Wang, C., Liu, Y.: Violence detection using Oriented Violent Flows. Image and Vision Computing **48** (2016) 37–41

[24] Zulkifley, M.A., Samanu, N.S., Zulkepeli, N.A.A.N., Kadim, Z., Woon, H.H.: Kalman filter-based aggressive behaviour detection for indoor environment. In: Information Science and Applications (ICISA) 2016, Springer (2016) 829–837

[25] Wang, K., Zhang, Z., Wang, L.: Violence video detection by discriminative slow feature analysis. In: Chinese Conference on Pattern Recognition, Springer (2012) 137–144

[26] Khokher, M.R., Bouzerdoum, A., Phung, S.L.: Violent Scene Detection Using a Super Descriptor Tensor Decomposition. In: Digital Image Computing: Techniques and Applications (DICTA), 2015 International Conference on, IEEE (2015) 1–8

[27] Ding, C., Fan, S., Zhu, M., Feng, W., Jia, B.: Violence Detection in Video by Using 3D Convolutional Neural Networks. In: Advances in Visual Computing, Springer (2014) 551–558

[28] Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: 2015 IEEE International Conference on Computer Vision (ICCV), IEEE (2015) 4489–4497

[29] Dai, Q., Zhao, R.W., Wu, Z., Wang, X., Gu, Z., Wu, W., Jiang, Y.G.: Fudan-Huawei at MediaEval 2015: Detecting Violent Scenes and Affective Impact in Movies with Deep Learning. In: MediaEval Workshop. (2015)

[30] Karpathy, A.: CS231n Convolutional Neural Networks for Visual Recognition. `http://cs231n.github.io/convolutional-networks/` [Online; Accessed: 23-June-2016].

[31] Hajian-Tilaki, K.: Receiver operating characteristic (ROC) curve analysis for medical diagnostic test evaluation. Caspian Journal of Internal Medicine **4**(2) (2013) 627

[32] Organization, W.H., et al.: Violence: a public health priority. WHO global consultation on violence and health. Geneva: World Health Organization (1996)

[33] Krug, E.G., Mercy, J.A., Dahlberg, L.L., Zwi, A.B.: The world report on violence and health. The Lancet **360**(9339) (2002) 1083–1088

[34] Fisher, R.B.: The PETS04 surveillance ground-truth data sets. In: Proc. 6th IEEE international workshop on performance evaluation of tracking and surveillance. (2004) 1–5

[35] Ryoo, M.S., Aggarwal, J.K.: Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In: 2009 IEEE 12th international conference on computer vision, IEEE (2009) 1593–1600

[36] Bermejo, E., Deniz, O., Bueno, G., Sukthankar, R.: Violence detection in video using computer vision techniques. In: Computer Analysis of Images and Patterns, Springer (2011) 332–339

[37] Demarty, C.H., Penet, C., Soleymani, M., Gravier, G.: VSD, a public dataset for the detection of violent scenes in movies: design, annotation, analysis and evaluation. Multimedia Tools and Applications **74**(17) (2015) 7379–7404

[38] Nam, J., Alghoniemy, M., Tewfik, A.H.: Audio-visual content-based violent scene characterization. In: Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on, IEEE (1998) 353–357

[39] Clarin, C., Dionisio, J., Echavez, M., Naval, P.: DOVE: Detection of movie violence using motion intensity analysis on skin and blood. Technical report, University of the Philippines (2005)

[40] Chen, D., Chen, M.y., Wactlar, H., Gao, C., Bharucha, A.: Video measurement of resident-on-resident physical aggression in nursing homes. In: Proceedings of the 1st ACM workshop on Vision networks for behavior analysis, ACM (2008) 61–68

[41] Giannakopoulos, T., Makris, A., Kosmopoulos, D., Perantonis, S., Theodoridis, S.: Audio-visual fusion for detecting violent scenes in videos. In: Artificial Intelligence: Theories, Models and Applications, Springer (2010) 91–100

[42] Gninkoun, G., Soleymani, M.: Automatic violence scenes detection: A multi-modal approach. In: Benchmark Workshop, MediaEval Workshop. (2011)

[43] Rasheed, Z.: Video categorization using semantics and semiotics. PhD thesis, University of Engineering and Technology, Karachi. (2003)

[44] Ionescu, B., Buzuloiu, V., Lambert, P., Coquin, D.: Improved cut detection for the segmentation of animation movies. In: Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on. Volume 2., IEEE (2006) II–II

[45] Serrano, I., Deniz, O., Bueno, G.: VISILAB at MediaEval 2013: Fight Detection. In: MediaEval. Volume 1043. (2013)

[46] Deniz, O., Serrano, I., Bueno, G., Kim, T.K.: Fast violence detection in video. In: Computer Vision Theory and Applications (VISAPP), 2014 International Conference on, IEEE (2014) 478–485

[47] Alazrai, R., Mowafi, Y., Lee, C.G.: Anatomical-plane-based representation for human–human interactions analysis. Pattern Recognition **48**(8) (2015) 2346–2363

[48] Serrano, I., Deniz, O., Bueno, G., Kim, T.K.: Fast fight detection. PLoS ONE **10**(4) (2015) e0120448

[49] Lloyd, K., Marshall, D., Moore, S.C., Rosin, P.L.: Detecting Violent Crowds using Temporal Analysis of GLCM Texture. ArXiv preprint arXiv:1605.05106 (2016)

[50] Mecocci, A., Micheli, F.: Real-time automatic detection of violent-acts by low-level colour visual cues. In: Image Processing, 2007. ICIP 2007. IEEE International Conference on. Volume 1., IEEE (2007) I–345

[51] Chen, D., Wactlar, H., Chen, D., Gao, C., Bharucha, A., Hauptmann, A.: Recognition of aggressive human behavior using binary local motion descriptors. In: Engineering in Medicine and

Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE, IEEE (2008) 5238–5241

[52] Lin, J., Wang, W.: Weakly-supervised violence detection in movies with audio and video based co-training. In: Advances in Multimedia Information Processing-PCM 2009, Springer (2009) 930–935

[53] Laptev, I.: On Space-Time Interest Points. International Journal of Computer Vision **64**(2-3) (2005) 107–123

[54] Chen, Y., Zhang, L., Lin, B., Xu, Y., Ren, X.: Fighting detection based on optical flow context histogram. In: Innovations in Bio-inspired Computing and Applications (IBICA), 2011 Second International Conference on, IEEE (2011) 95–98

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

[56] Zizi, T.K.T., Ramli, S., Ibrahim, N., Zainudin, N.M., Abdullah, L.N., Hasbullah, N.A.: Implementing Low Level Features for Human Aggressive Movement Detection. In: Advances in Visual Informatics, Springer (2015) 296–302

[57] Fu, E.Y., Leong, H.V., Ngai, G., Chan, S.: Automatic Fight Detection Based on Motion Analysis. In: 2015 IEEE International Symposium on Multimedia (ISM), IEEE (2015) 57–60

[58] Datta, A., Shah, M., Lobo, N.D.V.: Person-on-person violence detection in video data. In: Pattern Recognition, 2002. Proceedings. 16th International Conference on, IEEE (2002) 433–438

[59] Rota, P., Conci, N., Sebe, N., Rehg, J.M.: Real-life violent social interaction detection. In: Image Processing (ICIP), 2015 IEEE International Conference on, IEEE (2015) 3456–3460

[60] Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: Computer Vision and Pattern

Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 3169–3176

[61] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105

[62] Giannakopoulos, T., Kosmopoulos, D., Aristidou, A., Theodoridis, S.: Violence content classification using audio features. In: Advances in Artificial Intelligence, Springer (2006) 502–507

[63] Giannakopoulos, T., Pikrakis, A., Theodoridis, S.: A multi-class audio classification method with respect to violent content in movies using bayesian networks. In: Multimedia Signal Processing, 2007. MMSP 2007. IEEE 9th Workshop on, IEEE (2007) 90–93

[64] Patnaik, S., Yang, Y.M.: Soft Computing Techniques in Vision Science. Volume 395. Springer Science & Business Media (2012)

[65] Guha, T., Ward, R.K.: Learning sparse representations for human action recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on **34**(8) (2012) 1576–1588

[66] Barlow, H.B., Olshausen, B.A.: Convergent evidence for the visual analysis of optic flow through anisotropic attenuation of high spatial frequencies. Journal of Vision **4**(6) (2004) 415–426

[67] De Castro, E., Morandi, C.: Registration of translated and rotated images using finite Fourier transforms. IEEE Transactions on Pattern Analysis & Machine Intelligence (5) (1987) 700–703

[68] Jiang, Y., Liu, J., Zamir, A.R., Toderici, G., Laptev, I., Shah, M., Sukthankar, R.: Thumos challenge: Action recognition with a large number of classes. In: ECCV Workshop (`http://crcv.ucf.edu/THUMOS14`). (2014)

[69] Cho, H., Lee, H., Jiang, Z.: Evaluation of LC-KSVD on UCF101 Action Dataset. In: THUMOS: ICCV Workshop on

Action Recognition with a Large Number of Classes. Volume 7. (2013)

[70] Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems. (2014) 568–576

[71] Hu, K., Yin, L.: Multi-scale Topological Features for Hand Posture Representation and Analysis. In: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE (2013) 1928–1935

[72] Dollár, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on, IEEE (2005) 65–72

[73] Le, Q.V., Zou, W.Y., Yeung, S.Y., Ng, A.Y.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 3361–3368

[74] Taylor, G.W., Fergus, R., LeCun, Y., Bregler, C.: Convolutional learning of spatio-temporal features. In: Computer Vision–ECCV 2010, Springer (2010) 140–153

[75] Willems, G., Tuytelaars, T., Van Gool, L.: An efficient dense and scale-invariant spatio-temporal interest point detector. In: Computer Vision–ECCV 2008, Springer (2008) 650–663

[76] Yao, A., Gall, J., Van Gool, L.: A Hough transform-based voting framework for action recognition. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE (2010) 2061–2068

[77] Waltisberg, D., Yao, A., Gall, J., Van Gool, L.: Variations of a Hough-voting action recognition system. In: Recognizing Patterns in Signals, Speech, Images and Videos, Springer (2010) 306–312

[78] Gall, J., Yao, A., Razavi, N., Van Gool, L., Lempitsky, V.: Hough forests for object detection, tracking, and action recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on **33**(11) (2011) 2188–2202

[79] Niebles, J.C., Wang, H., Fei-Fei, L.: Unsupervised learning of human action categories using spatial-temporal words. International Journal of Computer Vision **79**(3) (2008) 299–318

[80] Breiman, L.: Random forests. Machine learning **45**(1) (2001) 5–32

[81] Vahdat, A., Gao, B., Ranjbar, M., Mori, G.: A discriminative key pose sequence model for recognizing human interactions. In: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, IEEE (2011) 1729–1736

[82] Nebel, J.C., Lewandowski, M., Thévenon, J., Martínez, F., Velastin, S.: Are current monocular computer vision systems for human action recognition suitable for visual surveillance applications? In: Advances in Visual Computing, Springer (2011) 290–299

[83] Mukherjee, S., Biswas, S.K., Mukherjee, D.P.: Recognizing interaction between human performers using: key pose doublet. In: ACM Multimedia Conference and Co-Located Workshops. (2011) 1329–1332

[84] Zhang, Y., Liu, X., Chang, M.C., Ge, W., Chen, T.: Spatio-temporal phrases for activity recognition. In: Computer Vision–ECCV 2012, Springer (2012) 707–721

[85] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11) (1998) 2278–2324

[86] Matsugu, M., Mori, K., Mitari, Y., Kaneda, Y.: Subject independent facial expression recognition with robust face detection using a convolutional neural network. Neural Networks **16**(5) (2003) 555–559

[87] Gulcehre, C.: Convolutional Neural Networks (LeNet). DeepLearning 0.1 Documentation. LISA Lab. `http://deeplearning.net/tutorial/lenet.html` [Online; Accessed: 30-June-2016].

[88] Davis, J., Bobick, A.: The Representation and Recognition of Action Using Temporal Templates. IEEE Transactions on Pattern Analysis and Machine Intelligence **23**(3) (2001) 928–934

[89] Leutenegger, S., Chli, M., Siegwart, R.Y.: BRISK: Binary robust invariant scalable keypoints. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 2548–2555

[90] Rosten, E., Drummond, T.: Fusing points and lines for high performance tracking. In: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, IEEE (2005) 1508–1515

[91] Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: International conference on artificial intelligence and statistics. (2010) 249–256

APPENDIX

APPENDIX A

Classification Result Interpretation

During test, some non-fight actions can be classified by error as fight actions and vice versa. These cases are denominated as False Positive (FP) and False Negative (FN), respectively. When a fight action is classified as fight action, it is called True Positive (TP). However, when a non-fight action is classified as a non-fight action, this is called True Negative (TN).

An important measure is classification accuracy (or simply accuracy) which is the fraction of all instances that are correctly categorized. In other words, it is the ratio of the number of correct classifications to the total number of correct or incorrect classifications:

$$Accuracy = \frac{TP + TN}{TotalPopulation} = \frac{TP + TN}{TP + TN + FP + FN} \tag{A.1}$$

A ROC curve, see an example in Figure A.1, is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied. The curve is created by plotting the True Positive Ratio (TPR) or sensitivity against the False Positive Ratio (FPR) or specificity at various threshold settings. These measures are defined as:

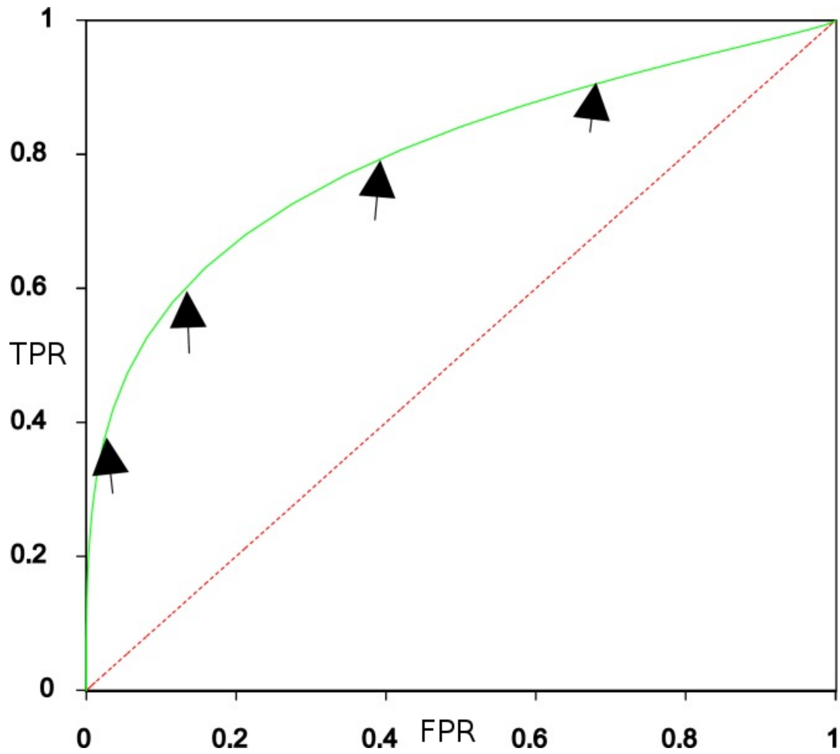

Figure A.1: A ROC curve example. The dotted line represents a random classification (taken from [31]).

- The *sensitivity*, recall or TPR (Equation A.2) represents the likelihood of classifying the fight actions as non-fight actions. Then, this coefficient indicates the capability of the selected classifier to detect non-fight actions in video.

$$Sensitivity = Recall = TPR = \frac{TP}{TP + FN} \tag{A.2}$$

- The *specificity* or FPR (Equation A.3) represents the likelihood of classifying the fight actions as truth fight actions. Therefore, in contrast to the sensitivity, the specificity indicates the capability of the classifier to detect fight actions in the video.

$$Specificity = FPR = \frac{TN}{TN + FP} \tag{A.3}$$

Classification accuracy or simply accuracy is defined as “the fraction of all instances that are correctly categorized. It is the ratio of the number of correct classifications to the total number of correct or incorrect classifications”.

Finally, the F1 score (also F-score or F-measure) is another measure of performance. It considers both the precision p and the recall r of the test: p is the number of correct positive results divided by the number of all positive results, and r is the number of correct positive results divided by the number of positive results that should have been returned. The F1 score can be interpreted as a weighted average of the precision and recall. The F1 score reaches its best value at 1 and worst at 0. Precision (Prec.) is defined as the number of positives detected divided by the total number of examples. So, in this case, precision means ”how useful the search results are”, and recall is “how complete the results are”.