MOTION-BASED REPRESENTATIONS FOR ACTIVITY RECOGNITION

CARLOS ANTÔNIO CAETANO JÚNIOR

MOTION-BASED REPRESENTATIONS FOR ACTIVITY RECOGNITION

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

Orientador: William Robson Schwartz
Coorientador: Jefersson Alex dos Santos

Belo Horizonte

January 2020

Caetano Júnior, Carlos Antônio
C128m Motion-Based Representations for Activity Recognition / Carlos Antônio Caetano Júnior. — Belo Horizonte, 2020
xxvii, 112 f. : il. ; 29cm

Tese (doutorado) — Universidade Federal de Minas Gerais
Orientador: William Robson Schwartz
Coorientador: Jefersson Alex dos Santos

1. Computação – Teses. 2. Redes neurais convolucionais – Teses. 3. Reconhecimento de atividades humanas – Teses. 4. informação espaço-temporal – Teses. I. Orientador. II. Coorientador. III. Título.

CDU 519.6*82.10(043)

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

FOLHA DE APROVAÇÃO

MOTION-BASED REPRESENTATIONS FOR ACTIVITY RECOGNITION

CARLOS ANTÔNIO CAETANO JÚNIOR

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

William Robson Schwartz

Prof. William Robson Schwartz - Orientador
Departamento de Ciência da Computação - UFMG

Prof. Jefersson Alex dos Santos - Coorientador
Departamento de Ciência da Computação - UFMG

Prof. Erickson Rangel do Nascimento
Departamento de Ciência da Computação - UFMG

João Paulo Papa

Prof. João Paulo Papa
Departamento de Computação - Unesp

Prof. David Menotti Gomes
Departamento de Informática - UFPR

Prof. Anderson de Rezende Rocha
Instituto de Computação - UNICAMP

Belo Horizonte, 27 de Janeiro de 2020.

Acknowledgements

Firstly, I would like to thank my supervisors Prof. William Robson Schwartz and Prof. Jefersson Alex dos Santos for their continued support, guidance and for providing this opportunity to work with them. I would also like to thank Dr. François Brémond for his guidance during my doctoral studies as visiting Ph.D. student at the Centre de Recherche INRIA Sophia Antipolis, Méditerranée, France.

Secondly, I would like to thank my family, friends, and the Smart Sense colleagues. Thanks for your patience and understanding. Last but not least, I would like to thank my girlfriend, Ana Paula, who always encourage me to trust myself. Thanks for your lovely support.

Finally, I would like to thank the National Council for Scientific and Technological Development – CNPq (Grants 438629/2018-3, 204952/2017-4 and 311053/2016-5), the Minas Gerais Research Foundation – FAPEMIG (Grants APQ-00567-14 and PPM-00540-17) and the Coordination for the Improvement of Higher Education Personnel – CAPES (DeepEyes Project). This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior - Brasil (CAPES) - Finance Code 001. Part of the results presented in this dissertation were obtained through research on a project titled "HAR-HEALTH: Reconhecimento de Atividades Humanas associadas a Doenças Crônicas ", sponsored by Samsung Eletrônica da Amazônia Ltda. under the terms of Brazilian federal law No. 8.248/91.

"Computer science is no more about computers than astronomy is about telescopes, biology is about microscopes or chemistry is about beakers and test tubes. Science is not about tools, it is about how we use them and what we find out when we do."
(Edsger Dijkstra)

Resumo

Nesta tese, quatro representações distintas baseadas em informações de movimento são propostas para o reconhecimento de atividades. A primeira é um descritor de características espaço-temporal que extrai um conjunto robusto de medidas estatísticas para descrever padrões de movimento medindo propriedades significativas em matrizes de co-ocorrência e capturando características espaço-temporais do movimento através da magnitude e orientação do fluxo ótico. A segunda é uma nova representação intermediária (*mid-level*) compacta baseada em matrizes de co-ocorrência de palavras visuais. Essa representação expressa a distribuição das características em um dado deslocamento utilizando um dicionário visual pré-calculado, codificando assim estruturas globais de várias características baseadas em regiões locais. A terceira representação é a proposta de um novo fluxo temporal para redes convolucionais de dois fluxos (*two-stream*) baseado em imagens calculadas a partir da magnitude e orientação do fluxo ótico. O método aplica transformações não lineares nos componentes vertical e horizontal do fluxo ótico para gerar imagens de entrada para o fluxo temporal. Por fim, a quarta é uma representação de esqueleto para ser usada como entrada para redes convolucionais. A abordagem codifica a dinâmica temporal calculando de forma explícita os valores de magnitude e orientação das articulações do esqueleto. Além disso, a representação tem a vantagem de combinar o uso de juntas de referência e um algoritmo de árvore de esqueleto, incorporando assim diferentes relações espaciais entre as juntas e preservando importantes relações espaciais. Os experimentos realizados em bases de dados desafiadoras e bastante conhecidas sobre reconhecimento de atividades (KTH, UCF Sports, HMDB51, UCF101 NTU RGB+D 60 e NTU RGB+D 120) demonstram que as representações propostas obtiveram resultados melhores ou similares em comparação ao estado da arte, indicando a adequação das abordagens para serem usadas como representações de vídeo.

Palavras-chave: Reconhecimento de atividades, redes neurais convolucionais, informação espaço temporal, fluxo ótico, componente temporal.

Abstract

In this dissertation we propose four different representations based on motion information for activity recognition. The first is a spatiotemporal local feature descriptor that extracts a robust set of statistical measures to describe motion patterns. This descriptor measures meaningful properties of co-occurrence matrices and captures local space-time characteristics of the motion through the neighboring optical flow magnitude and orientation. The second, is the proposal of a compact novel mid-level representation based on co-occurrence matrices of codewords. This representation expresses the distribution of the features at a given offset over feature codewords from a pre-computed codebook and encodes global structures in various local region-based features. The third representation, is the proposal of a novel temporal stream for two-stream convolutional networks that employs images computed from the optical flow magnitude and orientation to learn the motion in a better and richer manner. The method applies simple non-linear transformations on the vertical and horizontal components of the optical flow to generate input images for the temporal stream. Finally, the forth is a novel skeleton image representation to be used as input of Convolutional Neural Networks (CNNs). The proposed approach encodes the temporal dynamics by explicitly computing the magnitude and orientation values of the skeleton joints. Moreover, the representation has the advantage of combining the use of reference joints and a tree structure skeleton, incorporating different spatial relationships between the joints and preserving important spatial relations. The experimental evaluations carried out on challenging well-known activity recognition datasets (KTH, UCF Sports, HMDB51, UCF101, NTU RGB+D 60 and NTU RGB+D 120) demonstrated that the proposed representations achieved better or similar accuracy results in comparison to the state of the art, indicating the suitability of our approaches as video representations.

Palavras-chave: Activity recognition, convolutional neural networks (CNNs), spatiotemporal information, optical flow, temporal stream.

List of Figures

List of Tables

Acronym List

Acc	Accuracy
BoW	Bag-of-Words
CCW	Co-occurrence of Codewords
CNN	Convolutional Neural Network
CNNs	Convolutional Neural Networks
CRF	Conditional Random Field
CS	Co-occurrence Space
DT	Dense Trajectories
DTW	Dynamic Time Warping
FC	Fully-Connected
FV	Fisher Vector
FPS	Frames Per Second
FTP	Fourier Temporal Pyramid
GBH	Gradient Boundary Histograms
GLCM	Gray Level Co-occurrence Matrix
HMM	Hidden Markov Model
HOF	Histogram of Optical Flow
HOFM	Optical Flow Orientation and Magnitude
HOG	Histogram of Oriented Gradients

HOG3D	Histogram of Oriented Gradients 3D
IDT	Improved Dense Trajectories
JTM	Joint Trajectory Maps
LSA	Latent Semantic Analysis
LSTM	Long-Short Term Memory
MBH	Motion Boundary Histogram
MOS	Magnitude-Orientation Stream
MOS+D	Depth Weighting Magnitude-Orientation Stream
MST	Minimum Spanning Tree
NGLD	Normalized Google-Like Distance
NN	Neural Network
OFCM	Optical Flow Co-occurrence Matrices
RBF	Radial Basis Function
RGB	red, green, blue
RNN	Recurrent Neural Networks
SIFT	Scale Invariant Feature Transform
SSS	Spatial Segment Stream
STIP	Spatio Temporal Interest Points
ST-NGLDC	Spatio Temporal Normalized Google-Like Distance Correlogram
SURF	Speeded Up Robust Feature
SVM	Support Vector Machines
TSA	Temporal Scale Aggregation
TSN	Temporal Segment Networks
TSS	Temporal Segment Stream

TSRJI	Tree Structure Reference Joints Image
TSSI	Tree Structure Skeleton Image
VD2S	Very Deep Two-Stream network
VDSS	Very Deep Spatial Stream
VDTS	Very Deep Temporal Stream

Contents

Chapter 1

Introduction

Human activity recognition has been used in many real-world applications. In environments that require a higher level of security, surveillance systems can be used to detect and prevent abnormal or suspicious activities such as robberies and kidnappings. In addition, human activity recognition can be employed in video retrieval systems, so that a user is able to search for videos containing specific activities. It can also be employed in health care - e.g., monitoring systems can be used to monitor elderly people on their daily living activities.

Considering only surveillance applications, such systems have traditionally relied on network cameras monitored by a human operator that must be aware of the activities carried out by people who are in the cameras' field of view. With the recent growth in the number of cameras to be analyzed, the efficiency and accuracy of human operators has reached its limit [Keval, 2006]. Therefore, security agencies have attempted computer vision-based solutions to replace or assist the human operator. As a result, automatic recognition of activities is a problem that has attracted the attention of researchers in the area [Danafar & Gheissari, 2007; Reddy et al., 2011; Wiliem et al., 2012; Wang & Xu, 2016].

According to Aggarwal & Ryoo [2011], human activities can be categorized into four different levels: (i) gestures, which are atomic components that describe the motion of a person - i.e., body part movements; (ii) actions, composed of multiple gestures organized temporally, such as 'walking' or 'punching'; (iii) interactions, which are activities involving more than one subject or activities involving object manipulation - e.g., 'two people hugging' or 'typing on a keyboard'; and (iv) group activities, performed by groups composed of multiple persons and/or objects, such as 'military parade'.

To achieve automatic activity recognition, one can employ machine learning to generate statistical models that relate video content to activities being performed.

However, since the use of raw pixels provides no clues regarding video semantic concepts [Smeulders et al., 2000], there is a need for designing algorithms that create representations of video content based on characteristics, e.g., motion, shape, texture and skeleton pose information.

Over the last decade, a significant portion of the progress on the activity recognition task has been achieved with the design of discriminative representations known as handcrafted feature descriptors. In general, such representations are based on global features or local feature descriptors (2D or spatiotemporal) employed for the image and video domains, respectively [Krig, 2014], followed by encoding schemes using mid-level representations. Such information is based on appearance or motion analysis and representing the video in a more discriminative space is essential, allowing the improvement of activity recognition.

The feature extraction process is crucial since it allows the image/video content to be represented in a more discriminative and compact space, when compared to the direct use of pixels. These representations must be rich enough to allow proper recognition. Typically, extracted features tend to be invariant to transformations such as rotation, scaling or illumination changes. To that end, the most common approach is to extract 2D local feature descriptors of the image domain by using methods such as Histogram of Oriented Gradients (HOG) [Dalal & Triggs, 2005].

The most employed local spatiotemporal feature descriptors are either generalizations of the aforementioned descriptors or based on motion analysis using optical flow [Poppe, 2010]. There exists a large number of works on local spatiotemporal feature descriptors based on optical flow [Dalal et al., 2006; Laptev & Lindeberg, 2006; Wang et al., 2011]. However, according to Oliva & Torralba [2007], these descriptors perform well below expectations in extreme cases. Moreover, the flow information is not fully exploited due to dimensionality reduction and histogram binning [Dollar et al., 2005]. Consequently, these approaches do not necessarily capture interesting interactions from a surveillance point of view, discarding important information concerning spatial relations among the optical flow field [Ryan et al., 2011].

A smaller number of works focus on the design of global spatiotemporal feature descriptors [Mota et al., 2014; Shao et al., 2014]. A drawback of global features is that they can be influenced by motions of multiple objects and variations in the background [Schuldt et al., 2004].

Significant progress on the activity recognition task has been achieved due to the design of the aforementioned discriminative representations based on RGB frames. However, due to the development of cost-effective RGB-D sensors (e.g., Kinect), it became possible to employ different types of data such as depth information as well as human skeleton joints to perform 3D activity recognition. Compared to RGB or depth information, skeleton based methods have demonstrated impressive results by modeling temporal dynamics in videos. These approaches have the advantage of being computationally efficient due to a smaller data size and being robust to illumination changes, background noise and invariance to camera views [Han et al., 2017].

In the last decade, many works that employed skeleton information as feature representation model temporal dynamics in videos by employing Dynamic Time Warping (DTW), FTP or Hidden Markov Model (HMM) in conjunction with skeleton handcrafted feature descriptors, such as covariance matrices of joint trajectories, relative positions of joints, or rotations and translations between body parts [Wang et al., 2012; Yang & Tian, 2012; Hussein et al., 2013; Zanfir et al., 2013; Gowayyed et al., 2013; Vemulapalli et al., 2014; Wang et al., 2014; Devanne et al., 2015].

Regarding mid-level image representations, the BoW model [Sivic & Zisserman, 2003] or Fisher vector (FV) [Sánchez et al., 2013] are the most common approaches for encoding the feature descriptors. Such models can be understood as a quantization of the local features to create a representation for the whole image or video.

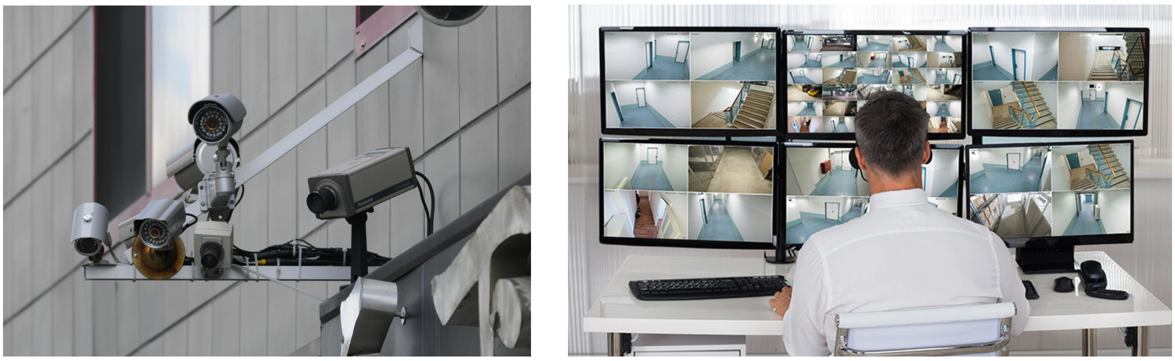
Nowadays, large efforts have been dedicated to the employment of deep Convolutional Neural Networks (CNNs) as representation learning. These approaches learn hierarchical layers of representations to perform pattern recognition and have achieved effective results on the activity recognition task [Simonyan & Zisserman, 2014; Du et al., 2015; Wang et al., 2016a; Carreira & Zisserman, 2017; Yang et al., 2018].

1.1 Motivation

According to Herath et al. [2017], analyzing human motion or activities has a long history and are attractive to various disciplines such as psychology, biology and computer science. There are motivations focused both on applications as well as on scientific purposes to cope with the activity recognition task.

Regarding its applications, activity recognition plays a key role in a broad range of potential areas, such as the following:

Security: In environments that require a higher level of security, such as banks and airports, surveillance systems can be used to detect and prevent abnormal or suspicious activities such as robberies and kidnappings. Currently, such systems are operated by humans so the efficiency could drastically be affected by the large number of screens that need to be analyzed (see Figure 1.1). Consequently,



(a) Surveillance cameras

(b) Surveillance operator monitoring several cameras

Figure 1.1. Example of multiple surveillance cameras and a human operator monitoring several screens from a surveillance system.

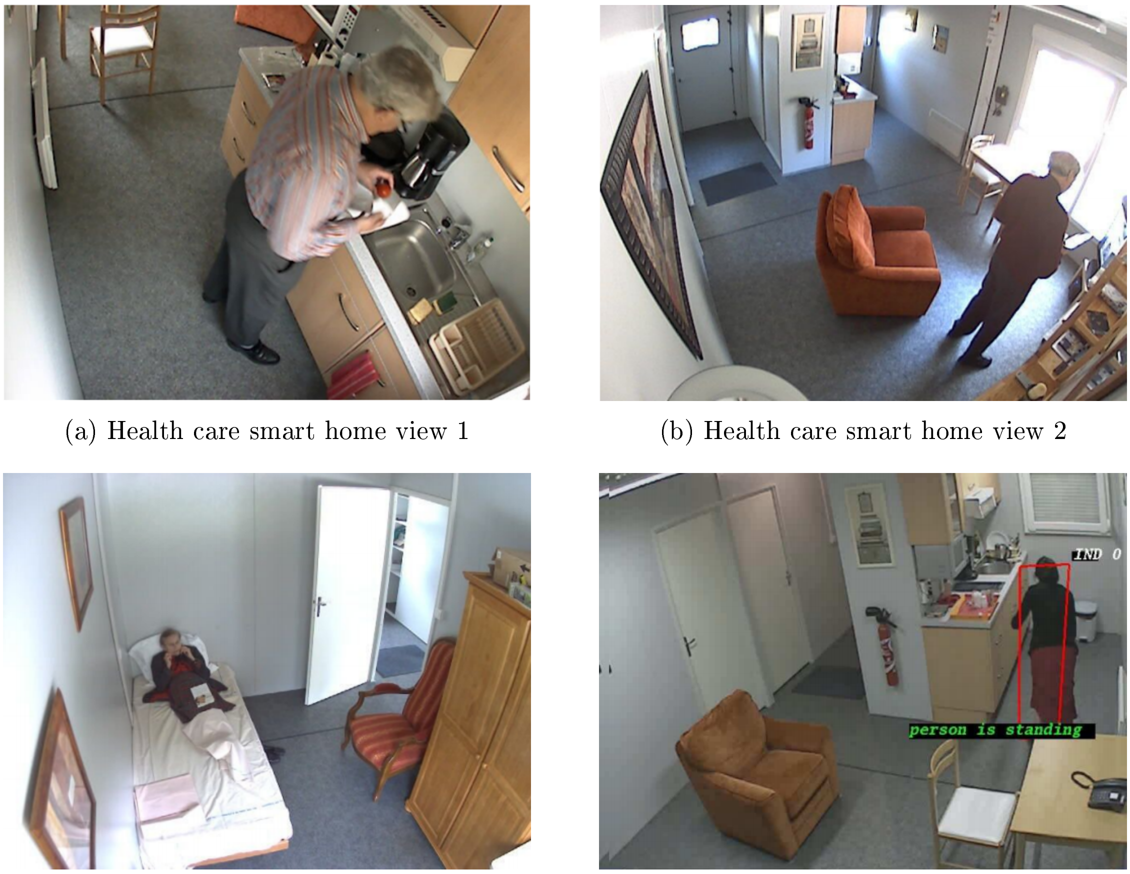
automated surveillance systems could be used to generate detection and tracking of people executing suspicious activities in the surveilled scene.

Web search: During the last century, a huge amount of visual data has arisen on the Internet, e.g., videos. The reason for such growth lies behind the availability of digital cameras - i.e., the accessibility of cellphones with built-in cameras. According to the Youtube website, the number of videos being uploaded have tremendously increased. Youtube has more than a billion users, which accounts for almost one third of Internet users. Every day, these people watch billions of hours of video, generating billions of views. Moreover, since March 2015, content creators who filmed their videos on Youtube Spaces have produced more than ten thousand videos, generating one billion views and more than 70 million hours of viewing[1]. As the activity recognition task becomes more accurate, the user should be able to search by visual content containing specific activities. Thus, efficient activity representations can be used to support such application.

Health care: Activity recognition has been also used in health care systems that consist of monitoring diseased people. Such systems can be used to detect symptoms and also to recognize the progression of certain diseases on patients with physical and mental health problems based on the activities performed by the person. Furthermore, health care systems are commonly used to monitor elderly people in smart homes by recognizing activities of daily living guaranteeing that they live healthier and safer. Figure 1.2 illustrates an example from a smart home adapted with a health care system[2].

[1] https://www.youtube.com/yt/about/press/

[2] http://www-sop.inria.fr/members/Francois.Bremond/topicsText/gerhomeProject.html



(a) Health care smart home view 1

(b) Health care smart home view 2

(c) Health care smart home view 3

(d) Activity recognition on Health care smart home

Figure 1.2. Example of a health care smart home. Images extracted from the GERHOME dataset [Zouba et al., 2007].

Research on activity recognition has also become a hot topic in the scientific community nowadays. Although significant improvements have been achieved, activity recognition still lacks on performance when using representations. Many works [Feichtenhofer et al., 2016; Park et al., 2016; Diba et al., 2016] point that the potential reason behind such gap falls in two cases: (i) current datasets do not have enough videos for training and are too noisy; and (ii) current CNN architectures are still not able to handle temporal information (or to take full advantage of it), consequently letting spatial (appearance) information prevail. Exploring such representations is the subject of this dissertation.

1.2 Challenges

Although several works present promising results, activity recognition still remains a challenging task that must cope with some problems, such as occlusion, viewpoint, camera movement and class similarities.

Occlusion: It is a common condition that prevents us from observing the occurrence of the activity. Some parts of the body of the person performing the activity may be occluded by objects present in the scene or the object used to perform the activity could also be occluded. Figure 1.3(a) illustrates that challenge.

Viewpoint: Activities can be captured by different cameras from distinct angles or even by multiple cameras, changing the appearance on the video. As a consequence, the direction of motions captured during the recording appear considerably differently according to viewpoint. Figure 1.3(b) illustrates that challenge.

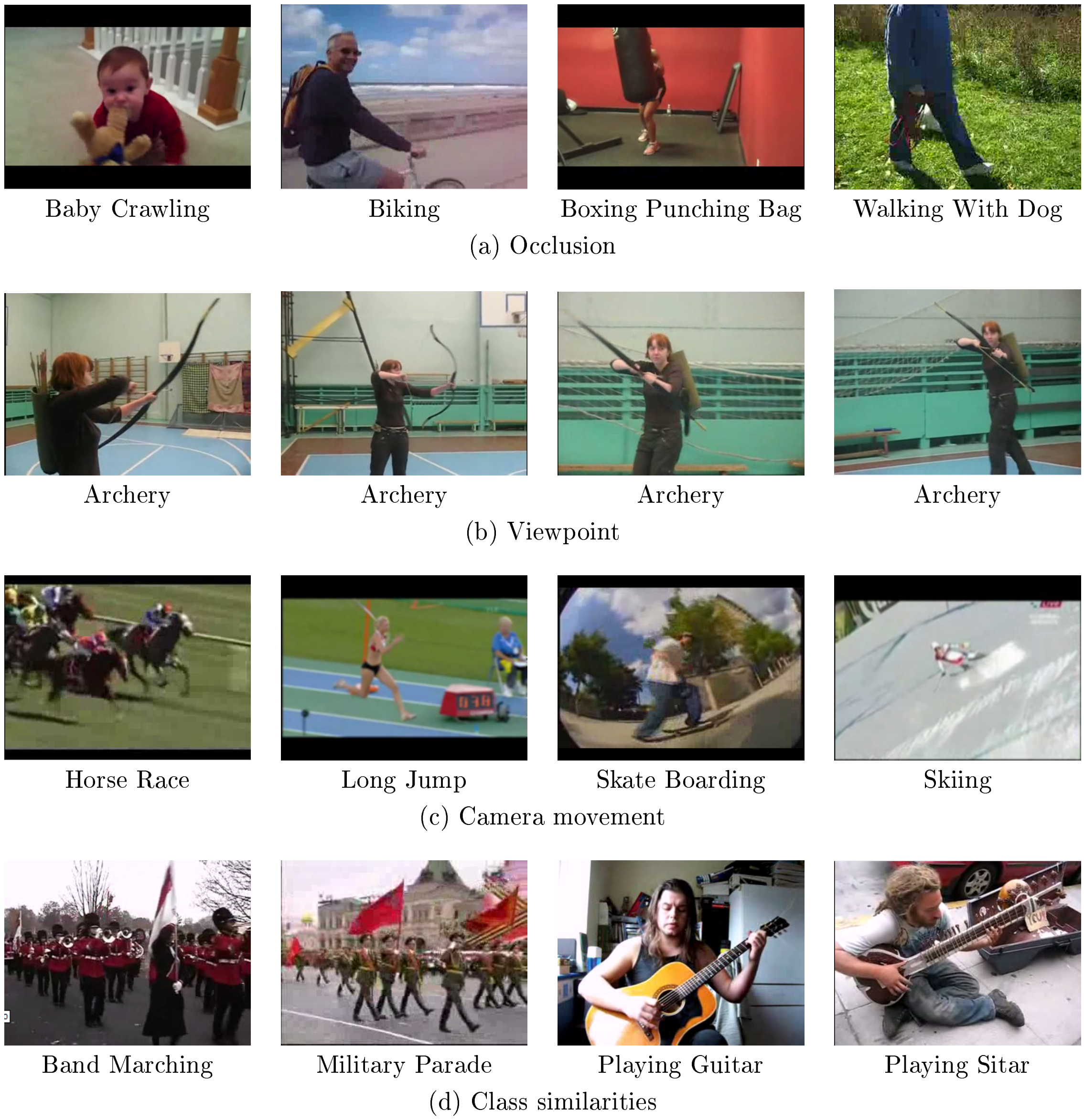
Camera movement: Moving the camera while filming brings two main issues to the activity recognition task: blur and unwanted motion information. The blurring can affect the efficiency of activity recognition by suppressing the visual information of the person performing the activity in the scene or the object used to perform it. Moreover, the unwanted motion information generated due to camera movement can somehow confuse the representations that make use of the motion feature. Figure 1.3(c) illustrates that challenge.

Class similarities: An activity category might look similar to other categories due to similar appearance or even due to similar motions. For instance, the activity *playing guitar* may be very similar in terms of the movements performed in contrast to *playing sitar* category, on the other side *band marching* and *military parade* presents very similar appearance and motion information. Figure 1.3(d) illustrates that challenge.

1.3 Hypothesis

The temporal component of videos provides an important clue for activity recognition, as a number of activities can be reliably recognized based on motion information. Hence, a significant portion of the progress on the activity recognition task has been achieved with the design of discriminative representations exploring temporal information that is based on motion analysis. It is essential to represent the video in a more discriminative space, allowing the improvement of activity recognition.

The most employed representation in the literature for encoding motion information is based on optical flow analysis. Optical flow provides displacement fields in horizontal and vertical axes that encodes the pixel level motions in a video clip. Thus,



(a) Occlusion

(b) Viewpoint

(c) Camera movement

(d) Class similarities

Figure 1.3. Example of challenges faced in the activity recognition task. Images extracted from UCF101 dataset [Soomro et al., 2012].

a typical hypothesis found in the literature is that the motion information of an activity can be represented by its optical flow encoding. Concerning that hypothesis, we design our representations based on the assumption that the motion information on a video sequence can be described by the spatial relationship contained on the local neighborhood of magnitude and orientation extracted from the optical flow or skeleton information. More specifically, we assume that the motion information is adequately specified by fields of magnitude and orientation.

A secondary hypothesis considered in this dissertation is that the motion information on a video sequence can be described by the spatial relationship contained on the local neighborhood of a set of quantized local feature descriptors.

A third hypothesis considered in this dissertation is that the motion information on a pose skeleton data can be described by the spatial relationship contained on the local neighborhood of a set of joints.

In a formal way, the problem statement that this dissertation deals with can be formulated as follows:

> *Given a video dataset of human activity categories, how to represent their motion content information for classifying the activity being performed in the video sequence?*

1.4 Contributions

The contributions of this dissertation are the development of four different representations based on motion information for activity recognition: (i) a local feature descriptor; (ii) a mid-level representation; (iii) optical flow deep learning based approaches; and (iv) motion skeleton deep learning based approaches.

By analyzing the classical local features descriptors in the literature, we noticed some possible improvements that could be made. Thus, we propose the Optical Flow Co-occurrence Matrices (OFCM) local feature descriptor [Caetano et al., 2016], which extracts a robust set of statistical measures from the spatial relationship of the local motion patterns. Thus, OFCM is based on a common technique found in the literature [Watanabe et al., 2008; Kobayashi & Otsu, 2008, 2012; Maki et al., 2011] used to extract information related to global structures in various local region known as co-occurrence matrices. Moreover, the proposed feature descriptor is based on motion analyses by employing optical flow.

Regarding mid-level representation, we propose the Co-occurrence of Codewords (CCW) [Caetano et al., 2018] which is also based on co-occurrence matrices, however here we use sampled features to build the representation. The method captures local relationships among the sampled local features, through the computation of a set of statistical measures. Co-occurrence of features is also commonly used in the literature [Banerjee & Nevatia, 2011; Zhang et al., 2012; Sun & Liu, 2013] to represent features as mid-level representation.

Another contribution is the employment of deep learning technique to encode motion information. We propose a novel feeding scheme for Convolutional Neural Network (CNN)-based on images computed from the optical flow, named Magnitude-

Orientation Stream (MOS) [Caetano et al., 2017], to learn the motion in a better and richer manner. Our method applies simple nonlinear transformations (magnitude and orientation) on the vertical and horizontal components of the optical flow to generate the input images. We also extend the approach to compensate for the distance of the subjects performing the activity to the camera with a depth weighting scheme, which we call Depth Weighting Magnitude-Orientation Stream (MOS+D) [Caetano et al., 2019a].

Regarding skeleton information, we propose the SkeleMotion representation [Caetano et al., 2019c]. The proposed approach encodes temporal dynamics by explicitly using motion information computing the magnitude and orientation values of the skeleton joints. To that end, different temporal scales are used to aggregate more temporal dynamics to the representation making it able to capture long-range joint interactions involved in activities.

We also introduce the Tree Structure Reference Joints Image (TSRJI) representation [Caetano et al., 2019b] which is also based on skeleton information, however here we improve the representation of skeleton joints by combining the use of reference joints [Ke et al., 2017] and a tree structure skeleton [Yang et al., 2018]. The method takes advantage of a structural organization of joints that preserves spatial relations of more relevant joint pairs and incorporates different spatial relationships between the joints.

Finally, we present an empirical comparison of our proposed representations against the state-of-the-art methods in the activity recognition task. The experiments are conducted on various challenging datasets, including KTH [Schuldt et al., 2004], UCF Sports [Rodriguez et al., 2008], HMDB51 [Kuehne et al., 2011], UCF101 [Soomro et al., 2012], NTU RGB+D 60 [Shahroudy et al., 2016] and NTU RGB+D 120 [Liu et al., 2019], and have shown the advantage of the proposed representations when compared to traditional techniques (see Chapter 5).

All the aforementioned representations are employed considering the four different levels of human activity categorized by Aggarwal & Ryoo [2011] - i.e., gestures, actions, interactions, and group activities.

This dissertation has led to five refereed international conference papers and one refereed journal paper (see Chapter 6).

1.5 Outline

The remainder of the text is organized as follows.

Chapter 2 - Literature Review and Theoretical Concepts We give a detailed description of the classical representations for human activity recognition: (i) local feature descriptors, for image/video and skeleton domains; (ii) mid-level representations; and (iii) CNN-based approaches, also for image and skeleton domains. Moreover we provide theoretical concepts for a better understanding of the dissertation.

Chapter 3 - Proposed Representations We introduce the proposed human activity recognition representations of this dissertation: (i) the local feature descriptor, OFCM; (ii) the mid-level representation, CCW; (iii) the optical flow CNN-based approaches, MOS and MOS+D; and (iv) the motion skeleton deep learning based approaches, SkeleMotion and TSRJI.

Chapter 4 - Challenges and Benchmarks Addressed We introduce the human activity recognition datasets used in this dissertation, providing a discussion about their characteristics and how they differ from each other.

Chapter 5 - Experimental Analysis We provide our empirical results regarding the proposed representations on contrast with state-of-the-art methods on many activity recognitions datasets. We also present a discussion by showing a detailed comparison of when our methods prevail and also where they fail.

Chapter 6 - Conclusions and Next Steps We present our concluding remarks and register our thoughts regarding the next steps of this dissertation.

Chapter 2

Literature Review and Theoretical Concepts

In the literature, we can clearly find two types of approaches for the human activity recognition task: (i) handcrafted feature based methods and (ii) Convolutional Neural Network (CNN)-based approaches.

Poppe [2010] describes a handcrafted feature based method as a larger method that can be considered a combination of two main processes: the video representation creation and recognition and classification of the video representations. The video representation creation process is crucial since it allows the content to be represented in a more discriminative and compact space by employing local feature descriptors extraction. These representations must be rich enough to allow proper recognition. Typically, extracted features tend to be invariant to transformations such as rotation, scaling or illumination changes. Thus, the most common procedure is to extract spatiotemporal local feature descriptors based on the image appearance (spatial), pose information (skeleton) or based on motion analysis (temporal) using optical flow. For instance, one common technique found in the literature is known as co-occurrence matrices, which are used to extract information related to global structures in various local regions.

Commonly, after the feature extraction process, a codification is applied by intermediate representations, known as mid-level representations. The main purpose of such representations is to combine the local features by a quantization step followed by a combination step to summarize the quantized local features into a single vector representing the video sequence. Hence, the Bag-of-Words (BoW) model [Sivic & Zisserman, 2003] is the most common approach for encoding feature descriptors. Finally, the mid-level representation is sent to a classification step to learn a function able to assign discrete labels to the video segments. Thus, the majority of human activity recognition works make use of machine learning techniques such as probabilistic graphical models.

We find in the literature a second family of methods involving large efforts on developing CNN approaches as representation learning. Such methods are based on architectures that learn hierarchical layers of representations to perform pattern recognition and have demonstrated impressive results on vision classification problems including activity recognition.

This chapter reviews the aforementioned techniques as well as both families of approaches for human activity recognition. First, we introduce the concepts of co-occurrence and its application as local feature descriptor (Section 2.1). After, on Section 2.2, we survey the literature on the most employed local features descriptors for activity recognition on image/video and skeleton domains. Then, we present the most used mid-level representations, in particular the classical BoW model and the methods based on co-occurrence (Section 2.3). Finally, on Section 2.4, we review methods based on CNN, also on image/video and skeleton domains.

2.1 Co-occurrence

The first use of co-occurrence as local feature was introduced by Haralick et al. [1973] to describe textural information on gray level images. Known as Gray Level Co-occurrence Matrix (GLCM), the method estimates the joint distribution of pixel intensity given a distance and an orientation. The co-occurrence matrix Σ is defined over an $n \times m$ image I, at a specified offset (Δ_x, Δ_y), as

$$\Sigma_{\Delta_x,\Delta_y}(i,j) = \sum_{r=1}^{n}\sum_{q=1}^{m} \begin{cases} 1, & \text{if } I(r,q) = i \text{ and} \\ & \quad I(r+\Delta_x, q+\Delta_y) = j, \\ & \\ 0, & \text{otherwise} \end{cases}$$

where i and j are the image intensity values of pixels separated by a distance d, r and q are the spatial positions in the image I and the offset (Δ_x, Δ_y) depends on the angle π used. Usually, π is expressed as angles $0\,^\circ$ $(0, d)$, $45\,^\circ$ $(-d, d)$, $90\,^\circ$ $(-d, 0)$ and $135\,^\circ$ $(-d, -d)$. Figure 2.1 illustrates possible offset configurations.

After the computation of four co-occurrence matrices ($\Sigma_{0\,^\circ}$, $\Sigma_{45\,^\circ}$, $\Sigma_{90\,^\circ}$ and $\Sigma_{135\,^\circ}$), a set of statistical measures is extracted from the four matrices and then used as feature descriptors. Aiming at depicting the meaningful properties contained in the co-occurrence matrices, Haralick et al. [1973] introduced 14 statistical measures that

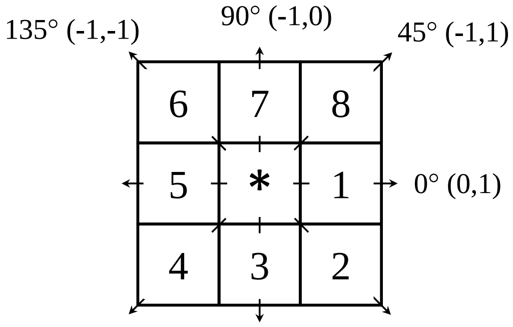


Figure 2.1. Offset configurations with $d = 1$. Cells 1 and 5 are the 0° (horizontal) nearest neighbors to cell *; cells 2 and 6 are the 135° nearest neighbors; cells 3 and 7 are the 90° nearest neighbors; and cells 4 and 8 are the 45° nearest neighbors to *. Note that this information is purely spatial, and has nothing to do with pixel intensity values Haralick et al. [1973].

became known in the literature as Haralick textural features. Such measures are extracted from the computed matrices and are: angular second moment, contrast, correlation, sum of squares, inverse difference moment, sum average, sum variance, sum entropy, entropy, difference variance, difference entropy, two information measures of correlation, and maximal correlation coefficient (the definitions of such features are given in Appendix B).

In the literature, co-occurrence is often used to extract information related to global structures in various local features [Kobayashi & Otsu, 2008; Watanabe et al., 2008; Maki et al., 2011; Kobayashi & Otsu, 2012] and also in mid-level representation [Banerjee & Nevatia, 2011; Zhang et al., 2012; Sun & Liu, 2013]. Consequently, we employ it in this dissertation as part of the process for a local feature descriptor as well as a mid-level representation. In the former, we use co-occurrence to extract the spatial relationship of local motion patterns, while in the latter we use it to capture local relationships among sampled local features.

2.2 Handcrafted Local Feature Descriptors

2.2.1 Image and Video domain

Over the last decade, a significant portion of the progress in visual recognition tasks has been achieved with the design of discriminative local feature descriptors. In general, such representations are based on 2D spatial feature descriptors employed for the image domain. Thus, the most common approach is the Histogram of Oriented Gra-

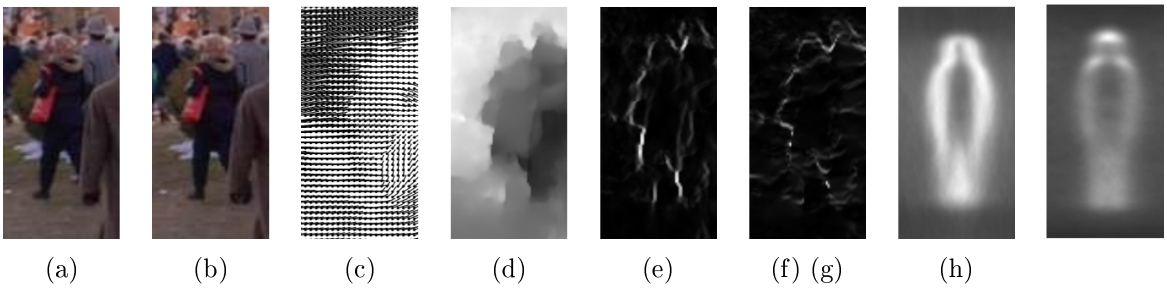
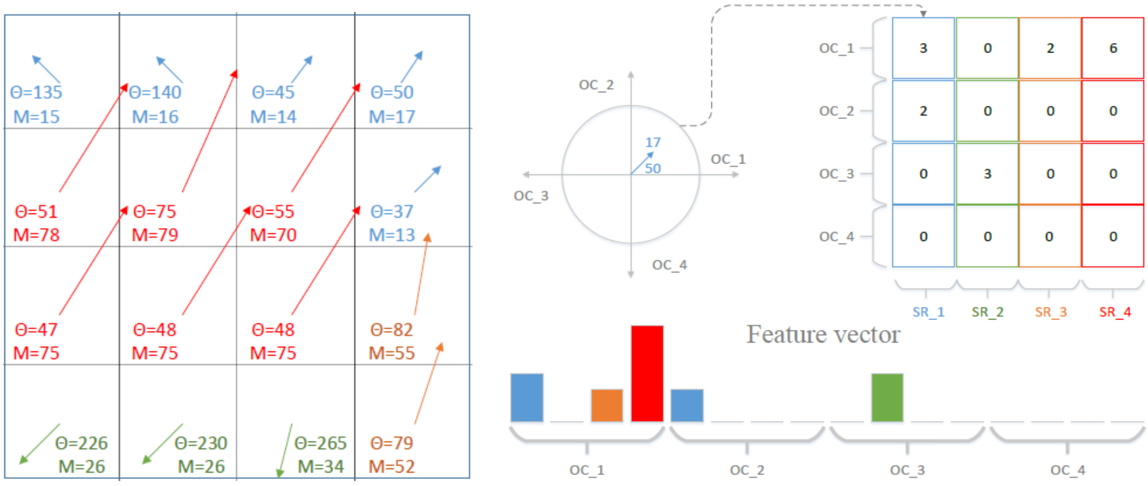


Figure 2.2. Illustration of the MBH descriptor. (a) and (b) show consecutive frames. (c) and (d) show the computed optical flow, and flow magnitude showing motion boundaries. (e) and (f) present the gradient magnitude of horizontal and vertical components for image pair (a) and (b). (g) and (h) illustrates the average MBH descriptor over all training images for flow field horizontal and vertical components. Figure extracted from Dalal et al. [2006].

dients (HOG) [Dalal & Triggs, 2005]. Spatial derivatives are computed for the local regions of the image and orientation information is quantized into histograms. With the aim of extending it to videos, Kläser et al. [2008] developed the Histogram of Oriented Gradients 3D (HOG3D). It is based on histograms of 3D gradient orientations computed using an integral video representation. The gradients are binned into regular polyhedrons in a multi-scale fashion in space and time. Therefore, HOG3D combines appearance and motion information.

To characterize motion and appearance of local features, Laptev et al. [2008] computed histogram descriptors of space-time volumes in the neighborhood of detected points. Each volume is subdivided into a grid of cuboids and, for each cuboid, they compute HOG [Dalal & Triggs, 2005] and Histogram of Optical Flow (HOF). The HOG descriptor is computed by dividing the cuboid into regions and accumulating a histogram binned by gradient directions over the pixels, while HOF is binned according to the flow orientations and weighted by its magnitude. Then, normalized histograms are concatenated and named HOG-HOF.

Dalal et al. [2006] introduced the Motion Boundary Histogram (MBH). First applied to human detection, the motion boundary coding scheme captures local orientations of optical flow borders based on HOG feature descriptors [Dalal & Triggs, 2005]. Handling the horizontal and vertical components (channels) of the optical flow as independent “images”, the authors take their local gradients separately, find the corresponding magnitudes and orientations and use these as weighted votes to the local orientation histograms. Figure 2.2 illustrates the steps for obtaining the MBH descriptor. Later on, the MBH was used on some works to describe motion information for activity recognition [Laptev et al., 2008; Wang et al., 2011; Wang & Schmid, 2013].



	SR_1	SR_2	SR_3	SR_4
OC_1	3	0	2	6
OC_2	2	0	0	0
OC_3	0	3	0	0
OC_4	0	0	0	0

(a) Optical Flow for a cuboid.

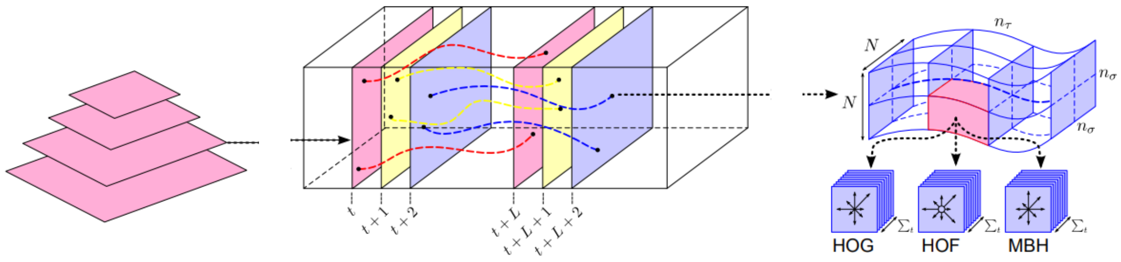
(b) Feature vector for cuboid the cuboid.

Figure 2.3. Example of Feature vector extraction using HOFM. (a) illustrates the resultant matrix of optical flow from a cuboid. (b) shows a matrix presenting four magnitude ranges: $\{(0,20],(20,40],(40,60],(60,\infty)\}$, named SR_1, SR_2, SR_3, SR_4. All magnitude are represented by colors blue, green, orange and red, respectively. It also presents four orientation ranges: $\{(0,90],(90,180],(180,270],(270,360]\}$, named as OC_1, OC_2, OC_3, OC_4. Figure from Colque et al. [2015].

Aiming at encoding both local static appearance and motion information, as in the HOG3D, but avoiding high dimensionality and a relatively expensive quantization cost, Shi et al. [2015] proposed the Gradient Boundary Histograms (GBH). Instead of using image gradients, the authors use time-derivatives of image gradients to emphasize moving edge boundaries. For each frame, they compute image gradients and apply temporal filtering over two consecutive gradient images. Then, they compute the magnitude and orientation for each pixel that are used to build a histogram of orientation as in HOG.

Colque et al. [2015] developed a descriptor called Histograms of Optical Flow Orientation and Magnitude (HOFM). In contrast to HOF that only encodes orientation information, HOFM captures the orientation and the magnitude of flow vectors providing information regarding the velocity of the moving objects. They build a 3D matrix based on the orientation and magnitude information provided by the optical flow field, where each line corresponds to a given orientation range and each column to the magnitude ranges. The authors then extended it to capture information regarding appearance and density of regions by encoding the entropy of the orientation flow [Colque et al., 2017]. Figure 2.3 illustrates a brief example of HOFM construction.

A major breakthrough on local feature-based approaches was achieved by Wang



(a) Dense sampling in each spatial scale.

(b) Tracking in each spatial scale separately.

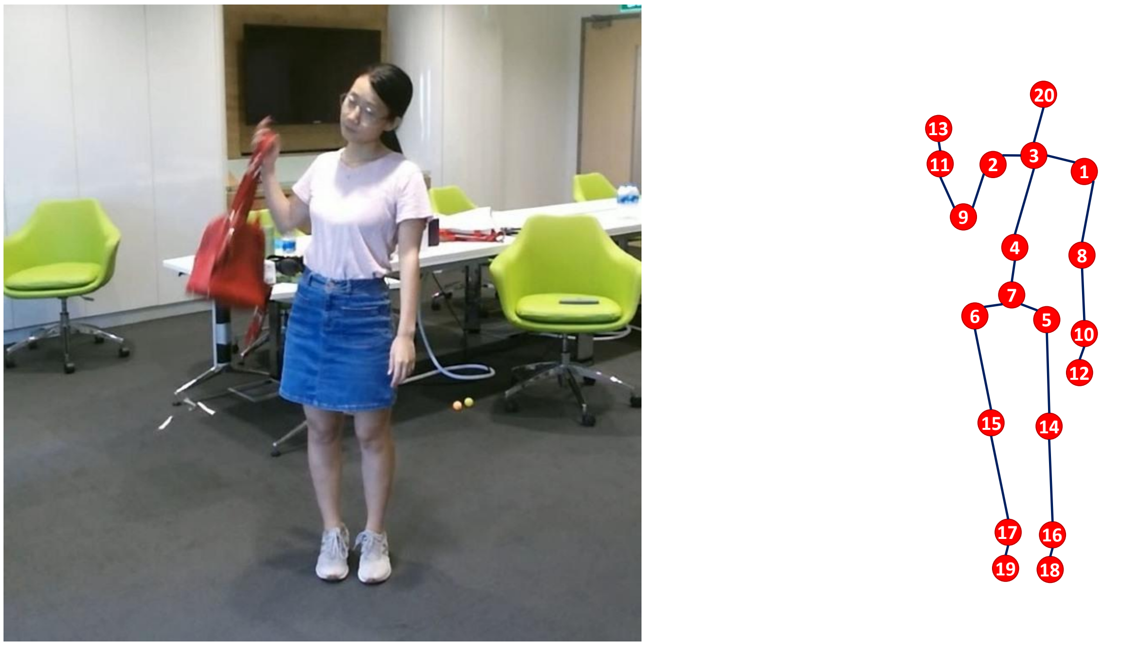
(c) Trajectory description.

Figure 2.4. Illustration of Dense Trajectory description. (a) densely sample interest points for multiple spatial scales. (b) tracking is performed in the corresponding spatial scale over frames. (c) trajectory descriptors are based on its shape, appearance and motion information along the trajectory. Figure from Wang et al. [2011].

et al. [2011] who proposed an method to describe videos by dense trajectories. Trajectory shapes encode local motion information by tracking spatial interest points over time. To generate the trajectories, they densely sample interest points for multiple spatial scales and track them based on displacement information using an efficient dense optical flow algorithm. The HOG, HOF and MBH feature descriptors are then used to describe the trajectories. Afterwards, the authors improved it to the Improved Dense Trajectories (IDT) [Wang & Schmid, 2013] using the homography between consecutive frames to estimate camera motion and Fisher vector encoding. Figure 2.4 illustrates the dense trajectories steps.

2.2.2 Skeleton domain

As mentioned before, the last decade was marked by significant progress on activity recognition thanks to the design of discriminative representations employed to the image and video domains on RGB data or optical flow. However, due to the development of low-cost RGB-D sensors (e.g., Kinect), it became possible to employ depth information as well as human skeleton joints to perform 3D activity recognition. Regarding the skeleton information, the human body can be represented as a graph-based data in which the nodes are the skeleton joints composed of x, y and z coordinates, and the edges are the bones that connect such joints (see Figure 2.5). Such graph can be seen as an articulated system of rigid segments connected by joints and human motion can be considered as a continuous evolution of the spatial configuration of these rigid segments [Zatsiorsky, 1998]. Compared to RGB and optical flow, skeleton data has the advantages of being computationally efficient since the data size is smaller. More-



(a) RGB frame (b) Skeleton information

Figure 2.5. Illustration of skeleton information. Nodes are the skeleton joints and the edges are the bones that connect the joints.

over, skeleton data are robust to illumination changes, robust to background noise and invariant to camera views [Han et al., 2017].

Hussein et al. [2013] introduced a feature descriptor for the human activity recognition task employing skeleton information based on covariance matrices. First, the all joint coordinates are normalized from 0 to 1 in all dimensions to make the feature scale invariant. To compute a covariance matrix, the authors make use of coordinates (x, y, z) from pose skeleton joints sampled over time provided by the Kinect sensor. Moreover, they employed multiple covariance matrices to encode the temporal dependency of joint locations. Hence, each matrix covers a sub-sequence of the input sequence. After that, the produced descriptors are normalized by L2 norm. Finally, the features are used as input to Support Vector Machines (SVM) classifier for prediction.

In the work of Wang et al. [2012], 3D joint coordinates were used to model the motion of the human body. Their feature represents the activity motion as the pairwise relative positions of the joints. First, the joint coordinates are normalized. According to the authors, such normalization step makes the descriptor invariant to the absolute body position, the initial body orientation and the body size of the subjects. After that, they extract pairwise relative position features by computing the difference between a joint i and that of each other joint j. Moreover, Wang et al. [2012] also proposed a temporal pattern representation called FTP to represent the temporal structure of an individual joint. To that end, they partition the activity into a pyramid style and employ the short time Fourier transform for all the segments. Finally, the Fourier coefficients from all the segments are concatenated and used as final feature representation. Figure 2.6 illustrates the FTP computation steps.

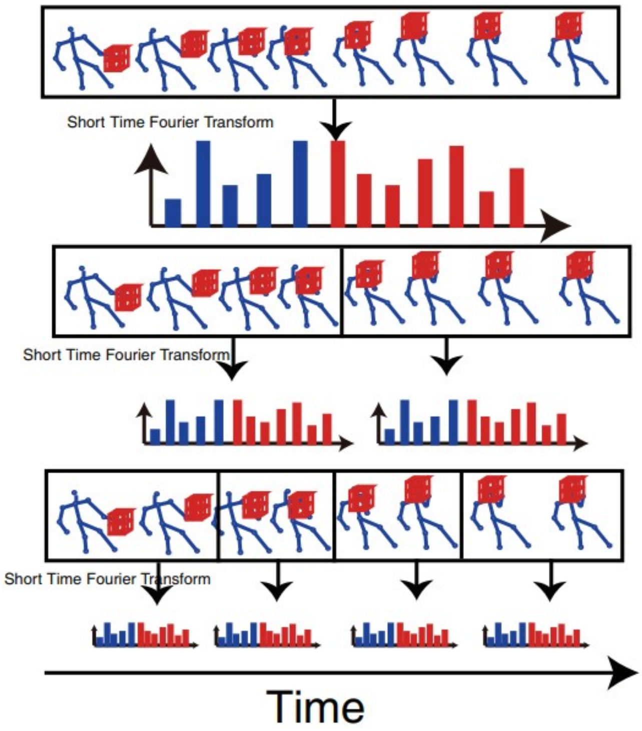

Short Time Fourier Transform
Short Time Fourier Transform
Short Time Fourier Transform
Time

Figure 2.6. A Illustration of the FTP computation steps. Figure from Wang et al. [2012].

Claiming that the relative geometry between body parts provides much more meaningful description power than their absolute locations, Vemulapalli et al. [2014] introduced a new body part-based skeletal representation. They explicitly model the relative 3D geometry between different body parts. To that end, given two rigid body parts, their relative geometry is described by employing the rotation and translation required to take one body part to the position and orientation of the other. The relative geometry between a pair of body parts is represented as a point in the special Euclidean group, and the entire human skeleton as a point in the Lie group. Finally, the proposed representation is modeled as curves in the Lie group and activity recognition can be performed by classifying these curves.

2.3 Mid-level Representations

Following the approach mentioned at the beginning of this chapter, after the local feature extraction, it is necessary to encode them someway to have a global representation of the video based on the local characteristics. This coding scheme is referred in the literature as an intermediate representation or mid-level representation.

The most widely used mid-level representation is known as Bag-of-Words (BoW), initially proposed by Sivic & Zisserman [2003]. Basically, the BoW model can be understood as the application of two critical steps: *coding* and *pooling*. The coding step quantifies the local features extracted from the video sequence according to a visual dictionary, also known as *codebook*, associating the extracted local features with the element closest to this visual dictionary. The visual dictionary is usually constructed by applying a clustering algorithm, such as *k-means* [Lloyd, 2006], to a set of sampled extracted local features, where each visual word (codeword) corresponds to the centroid obtained from each cluster. The pooling step summarizes the visual words obtained in a single feature vector to represent the entire video sequence.

In the literature, features originated from local feature descriptors are known as 1^{st} order features and features that encode spatial relationship between a set of patches as higher order features. In this way, to improve the BoW model, Liu et al. [2008] proposed an approach by calculating spatial histograms where the co-occurrences of local features are considered to encode spatial relationship as 2^{nd} order features. Thus, instead of assigning a local feature descriptor to a single codeword, one can assign it to the top-N closest codewords.

Banerjee & Nevatia [2011] introduced a Minimum Spanning Tree (MST)-based distance function to count the co-occurrence of Spatio Temporal Interest Points (STIP) [Laptev, 2005] codewords. Their approach models the neighborhood relationships in terms of a function that measures the pairwise co-occurrence frequency of the codewords. Their transformation represents a function that counts edge connectivity of latent variables of a undirected graphical model. The model is a conditional probability distribution of the unobserved variables conditioned over the observed variables. In this way, they explicitly learn the co-occurrence statistics as a part of its maximum likelihood objective function.

According to Zhang et al. [2012], the codeword space generated by the BoW model is too sparse and too diverse to result in an informative and compact representation of a video sequence. They also claim that ambiguities would be inevitably introduced by the assignment errors and uncertainty of the codebook's size. Thus, to reduce the ambiguities and obtain a more compact representation, Zhang et al. [2012] introduced a method based on text processing approaches, which employs Latent Semantic Analysis (LSA) to find a proper transform and map the codeword space into a co-occurrence space.

In [Sun & Liu, 2013], the authors model the semantic relationship (spatial and temporal) of codewords in terms of Normalized Google-Like Distance (NGLD), which measures the co-occurrence frequency of each pairwise codewords in the video. First, they use an auxiliary human detector to obtain body regions and then the NGLD correlogram is mapped to the temporal domain with a huge amount of primitives to cover the whole body region. To the spatial domain, they apply the same mapping scheme, however with codewords from the same frame. After that, the spatial and temporal correlograms are combined to aggregate bi-domain weights on the co-occurrence semantics of pairwise words. Finally, their mid-level representation is incorporated with the classical BoW.

2.4 Convolutional Neural Network Based Methods

2.4.1 Image and Video domain

Due to the impressive results achieved on image classification by CNNs [Krizhevsky et al., 2012], many works have tried to apply them to learn spatiotemporal information for the activity recognition task. A natural choice, the 3D convolutional network was presented by Ji et al. [2013], where they tried to learn both appearance and motion features with 3D convolution operations. Their method works by stacking consecutive segments of human subjects in videos and applying 3D convolutions over such volume. Tran et al. [2015] also explored 3D CNNs. However, in contrast with Ji et al. [2013], their method takes full video frames as inputs and does not rely on any preprocessing.

Karpathy et al. [2014] also used CNN to learn motion features. The authors investigated different temporal information fusion schemes, learning local motion direction/speed with global information. Although significant gains in accuracy were achieved compared to the works based on local features, only a small improvement was achieved when compared to single-frame CNN models, showing that the current CNN architectures are unable to efficiently learn motion features directly.

A major breakthrough was achieved by Simonyan & Zisserman [2014]. Instead of trying to learn motion information as Karpathy et al. [2014] and Tran et al. [2015], the authors incorporated it by using optical flow. Known as two-stream network, their architecture is composed of two stream of data: a spatial network, which takes as input the raw red, green, blue (RGB) pixels and a temporal network, which takes as input dense optical flow displacements computed across the frames. The final predictions are computed as the average of the output scores from the two streams. Such approach showed significant improvement over other approaches.

By employing the aforementioned two-stream network, Wang et al. [2015] conducted experiments showing the impact on results when changing the network architecture. In addition, they also introduced data augmentation techniques to improve the network training. To that end, the authors used three distinct architectures (ClarifaiNet [Zeiler & Fergus, 2014], GoogLeNet [Szegedy et al., 2015] and VGG-16 [Simonyan & Zisserman, 2015]), the best results were achieved by VGG-16's deeper architecture. Afterwards, the authors improved it to the TSN [Wang et al., 2016a] by studying different types of input modalities to two-stream and employing the inception with batch normalization network architecture [Ioffe & Szegedy, 2015].

Perez et al. [2017] used MPEG motion vectors [Richardson, 2003] as a different input for a two-stream network to explore temporal information. Such vectors are used to perform motion estimation in video compression; pixels are grouped in macroblocks and motion vectors are then computed for each block. They showed that both optical flow and MPEG motion vectors provide equivalent accuracy, but the latter allows a more efficient implementation. Afterwards, Varol et al. [2018] studied the impact of the use of different motion information as input for the networks. They investigated the dependency of activity recognition on the quality of motion estimation by experimenting three types of optical flow inputs: (i) MPEG motion vectors; (ii) Farneback optical flow estimator; and (iii) Brox optical flow. Their experiments confirmed the advantage of motion-based representations and highlight the importance of good quality motion estimation for learning efficient representations for human activity recognition.

As another type of input information, Zhu & Newsam [2016] performed an investigation with depth information for large-scale human activity recognition in video without using any depth sensor, such as Kinect-like devices. To that end, the authors estimated the depth information directly from the video itself by using two state-of-the-art approaches to extract depth from images and experimented them by feeding two different CNNs models.

To make a spatial network learn to relate which parts of the image are moving, Park et al. [2016] proposed a feature amplification technique by using magnitude information of the optical flow on the spatial network. To that end, they extracted features maps of the last convolutional layer of the spatial network, computed optical flow magnitudes and resize it to be the same size of the previously extracted feature maps. Finally, they perform element-wise product to amplify the activations.

To capture temporal dynamics of body parts over time, Zolfaghari et al. [2017]

proposed a combination of networks based on a three-stream architecture. Their method relies on three different inputs: the raw RGB images, optical flow information and human pose. For the later stream, the authors used a network for human body part segmentation that provides body pose information. According to the authors, the advantage of using the pose network is that it yields the spatial localization of the persons, which allows the application of the approach to spatial activity localization in a straightforward manner.

Revisiting 3D convolutions, Carreira & Zisserman [2017] argue that the reason 3D convolutions might be unable to improve over their flat counterparts lies on the dataset size. Consequently, they take state-of-the-art activity recognition architectures, inflate them to 3D convolutions and evaluate them on the novel Kinetics dataset [Kay et al., 2017], which provides a large amount of data. They showed that 3D convolution yields better results, claiming that the previous tries of modeling temporal information with 3D convolution failed due to noisy datasets and/or lack of data. In further experiments, Carreira and Zisserman evaluated both RGB and optical flow using 3D convolutions and showed that optical flow still has a leverage when compared to RGB.

2.4.2 Skeleton domain

Nowadays, large efforts have been directed to the employment of deep neural networks to model skeleton data by using two main approaches: (i) Recurrent Neural Networks (RNN) with Long-Short Term Memory (LSTM) [Veeriah et al., 2015; Shahroudy et al., 2016; Song et al., 2017; Zhang et al., 2017]; and (ii) skeleton image representations used as input to a CNN [Du et al., 2015; Wang et al., 2016b; Liu et al., 2017b; Ke et al., 2017; Li et al., 2017; Wang et al., 2018; Yang et al., 2018; Li et al., 2018; Choutas et al., 2018]. Although the former approach present excellent results in the 3D activity recognition task due to their power of modeling temporal sequences, such structures lack the ability to efficiently learn the spatial relations between the skeleton joints [Yang et al., 2018]. On the other hand, the latter takes the advantage of having the natural ability of learning structural information from 2D arrays and is able to learn spatial relations from the skeleton joints. Here we only review CNN approaches.

As the forerunner of skeleton image representations, Du et al. [2015] take advantage of the spatial relations by employing a hierarchical structure representing the skeleton sequences as a matrix. Each row of such matrix corresponds to a chain of concatenated skeleton joint coordinates from the frame t. Hence, each column of the matrix corresponds to the temporal evolution of the joint j. At this point, the matrix size is $J \times T \times 3$, where J is the number of joints for each skeleton, T is the total frame number of the video sequence and 3 is the number coordinate axes (x, y, z). The values of this matrix are quantified into an image (i.e., linearly rescaled to a $[0, 255]$) and normalized to handle the variable-length problem. In this way, the temporal dynamics of the skeleton sequence is encoded as variations in rows and the spatial structure of each frame is represented as columns. Finally, the authors use their representation as input to a CNN model composed of four convolutional layers and three max-pooling layers. After the feature extraction, a feed-forward neural network with two fully-connected layers is employed for classification. Figure 2.7 illustrates the skeleton image representation.

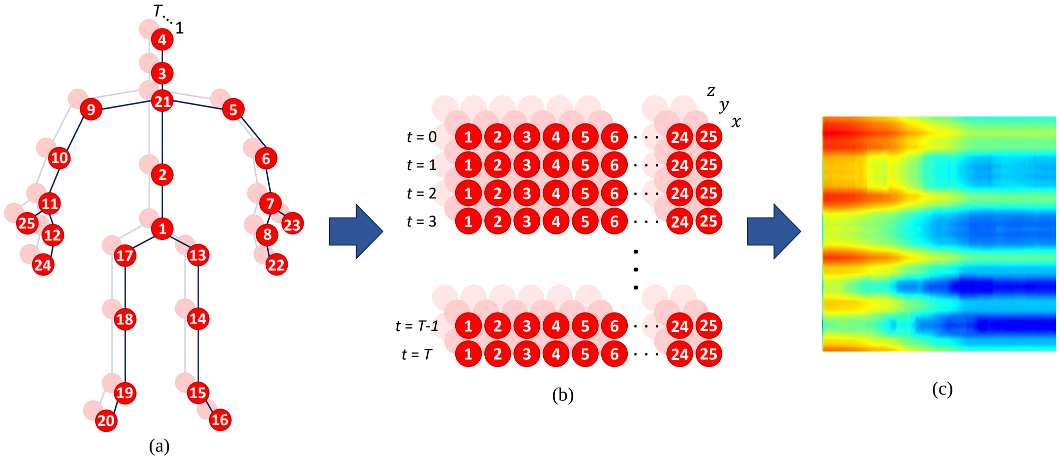


Figure 2.7. A Illustration of the skeleton image representation computation from Du et al. [2015]. (a) skeleton sequences; (b) matrix computation; and (c) final skeleton image representation.

Wang et al. [2016a, 2018] present a skeleton representation to encode both spatial configuration and dynamics of joint trajectories into three texture images through color encoding, named Joint Trajectory Maps (JTM). The authors apply rotations to the skeleton data to mimic multi-views and also for data enlargement to overcome the drawback of CNNs usually being not view invariant. JTMs are generated by projecting the trajectories onto the three orthogonal planes. To encode motion direction in the JTM, they use a hue colormap function to "color" the joint trajectories over the activity period. They also encode the motion magnitude of joints into saturation and brightness claiming that changes in motion results in texture in the JTM. Finally, the authors individually fine-tune three AlexNet [Krizhevsky et al., 2012] CNNs (one for each JTM) to perform classification.

Representations based on heat map to encode spatiotemporal skeleton joints were also proposed by Liu et al. [2017b]. Their approach considers each joint as 5D point space (x, y, z, t, j) and expresses them as a 2D coordinate space on a 3D color space.

Thus, they permute elements of the 5D point space. Nonetheless, such permutation can generate very similar representations which may contain redundant information. To that end, they use ten types of ranking to ensure that each element of the point (x, y, z, t, j) can be assigned to the color 2D coordinate space. After that, the ten skeleton representations are quantified and treated as a color image. Finally, the authors employ a multiple CNN-based model, one for each of the representations. They used the AlexNet [Krizhevsky et al., 2012] architecture and fused the posterior probabilities generated from each CNN for the final class score.

To overcome the problem of the sparse data generated by skeleton sequence video, Ke et al. [2017] represent the temporal dynamics of the skeleton sequence by generating four skeleton representation images. Their approach is closer to Du et al. [2015] method, however they compute the relative positions of the joints to four reference joints by arranging them as a chain and concatenating the joints of each body part to the reference joints resulting in four different skeleton representations. According to the authors, such structure incorporates different spatial relationships between the joints. Finally, the skeleton images are resized and each channel of the four representations is used as input to a VGG19 [Simonyan & Zisserman, 2015] pre-trained architecture for feature extraction.

To encode motion information on skeleton image representation, Li et al. [2017, 2018] proposed the skeleton motion image. Their approach is created similar to Du et al. [2015] skeleton image representation, however each matrix cell is composed of joint difference computation between two consecutive frames. To perform classification, the authors used Du et al. [2015] approach and their proposed representation independently as input for a neural network with a two-stream paradigm. The CNN used was a small seven-layer network consisting of three convolution layers and four fully-connected layers.

Yang et al. [2018] claim that the concatenation process of chaining all joints with a fixed order leads to a lack of semantic meaning and a loss of skeleton structural information. To that end, Yang et al. [2018] proposed a representation named Tree Structure Skeleton Image (TSSI) to preserve spatial relations. Their method is created by traversing a skeleton tree with a depth-first order algorithm with the premise that the fewer edges there are, the more relevant the joint pair is. The generated representation is then quantified into an image and resized before being presented to a ResNet-50 [He et al., 2016] CNN architecture.

2.5 Concluding Remarks

In this chapter, we surveyed the classical representations for human activity recognition: (i) local feature descriptors; (ii) mid-level representations; and (iii) CNN-based approaches. Moreover, we provided the main theoretical concepts of the co-occurrence technique that is applied in this dissertation.

We observed that although there are many approaches based on local feature descriptors and mid-level representations, these works often require a certain amount of engineering (e.g., feature extraction, mid-level representation and classifier training). On the contrary, CNNs are a class of deep learning models that replaces all engineering with a single neural network trained end to end from pixel values to classifier outputs [Karpathy et al., 2014].

The literature presents a large number of works on local feature descriptors based on motion and skeleton information. However, according to Oliva & Torralba [2007], these descriptors perform well below expectations in extreme cases. Moreover, the flow information is not fully exploited due dimensionality reduction and histogram binning [Dollar et al., 2005]. Consequently, these approaches do not necessarily capture interesting interactions from a surveillance point of view, discarding important information concerning spatial relations among the optical flow field [Ryan et al., 2011].

Regarding mid-level representations, many works improved the classical BoW model with co-occurrence with the objective of extracting information related to global structures in various local region-based features. However, even though the reviewed methods used co-occurrence to encode information, they do not compute co-occurrence matrices considering different offset angles on their description process.

As it can be inferred from the image and video domain CNNs reviewed methods, most of them use either convolution operations over raw pixels or optical flow to model temporal information. The former does not decouple spatial and temporal information, letting appearance information prevail [Feichtenhofer et al., 2016], while the latter approaches rely on horizontal and vertical components of the optical flow. Regarding the skeleton domain CNNs reviewed methods, most of them are improved versions of Du et al. [2015] skeleton image representation focusing on spatial structure of joint axes while the temporal dynamics of the sequence is encoded as variations in columns.

In Chapter 3, we introduce our proposed human activity recognition representations. Aiming at capturing more information from the optical flow, we propose a novel spatiotemporal local feature descriptor. The method is based on the extraction of Haralick features from co-occurrence matrices computed using the optical flow information. More specifically, we assume that the motion information is adequately specified by a set of magnitude and orientation co-occurrence matrices computed for various angular relationships at a given offset between neighboring vector pairs on the optical flow. As mid-level representation, we propose an approach based on the assumption that the information on a video sequence can be encoded by the spatial relationship contained on local neighborhoods of the features. The main difference between our approach and others is that we consider the co-occurrence of feature codewords and compute the Haralick features from the co-occurrence matrices as the final mid-level representation. To train the aforementioned approaches, we choose to apply the SVM technique as classifier.

Despite the optical flow-based CNN methods producing promising results, they focus only on displacement information. In view of that, aiming at capturing more information from the optical flow, we introduce a new temporal stream for the two-stream networks to perform activity recognition. The method is based on non-linear transformations on the optical flow components to generate input images for the temporal stream. It captures not only the displacement through the orientation but also the magnitude providing information regarding the velocity of the movement. Moreover, we also employed depth information by weighting the magnitude by the depth to compensate the distance of the subjects performing the activity to the camera.

Regarding the skeleton information representations, they do not explicitly encode rich motion information. Consequently, to capture more motion information, we introduce a new skeleton image representation that directly encodes motion by using orientation and magnitude to provide information regarding the velocity of the movement in different temporal scales. Thus, our approach differs from the literature methods because it captures the temporal dynamics explicitly provided by magnitude and orientation motion information. Moreover, we believe that performance can be improved by explicitly employing joint relationships, which enhances the temporal dynamics encoding. In view of that, we also introduce another skeleton-based approach that takes advantage of combining a structural organization that preserves spatial relations of more relevant joint pairs by using the skeleton tree with a depth-first order algorithm from Yang et al. [2018] and also by incorporating different spatial relationships by using the reference joints technique from Ke et al. [2017].

Chapter 3

Proposed Representations

The human activity recognition research domain has been widely explored in the recent years. The last decade has witnessed important breakthroughs in the area due to the design of discriminative representations. Such representations can be divided into two groups: (i) the development of handcrafted feature based methods that employ spatiotemporal local feature descriptors on image or skeleton domains (e.g., HOG-HOF [Laptev et al., 2008] and rotation and translation of joints [Vemulapalli et al., 2014]) followed by mid-level representations (e.g., BoW [Sivic & Zisserman, 2003]); and (ii) the design of CNN-based approaches, such as the two-stream network [Simonyan & Zisserman, 2014] for the image domain or skeleton image representation [Du et al., 2015] for the skeleton domain.

In the literature, the most employed representations for image and video domains are based on motion analysis using optical flow [Dalal et al., 2006; Laptev et al., 2008; Wang et al., 2011; Simonyan & Zisserman, 2014; Wang et al., 2015; Carreira & Zisserman, 2017]. The reason behind this lies in the fact that the motion in the video provides important clues for activity recognition, as a number of activities can be reliably recognized based on motion information. Thus, motion analysis is critical for the design of efficient activity recognition representations. On the other hand, skeleton image representations used as input for a CNN [Du et al., 2015; Wang et al., 2016b; Liu et al., 2017b; Ke et al., 2017; Li et al., 2017; Wang et al., 2018; Yang et al., 2018; Li et al., 2018; Choutas et al., 2018] takes advantage of having the natural ability of learning structural information from 2D arrays and is able to learn spatial relations and the temporal evolution patterns from the skeleton joints.

In this chapter we introduce representations based on motion analysis. We start by presenting the proposed spatiotemporal local feature descriptor named Optical Flow Co-occurrence Matrices (OFCM) (Section 3.1). Then, in Section 3.2, we introduce

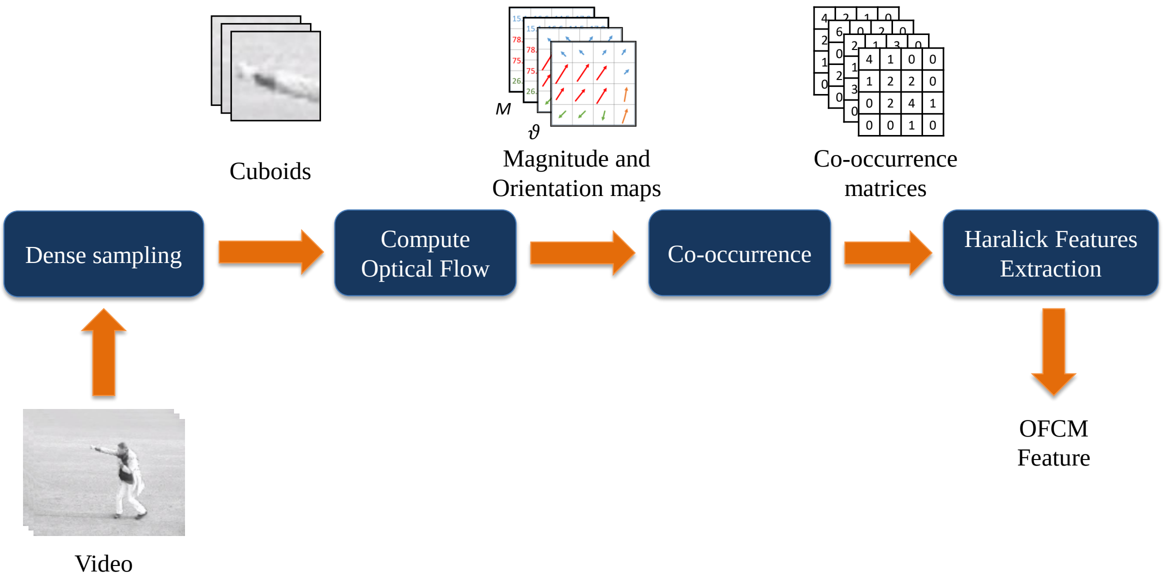


Figure 3.1. Diagram illustrating the pipeline extraction of the proposed spatiotemporal feature descriptor.

the Co-occurrence of Codewords (CCW), which is a mid-level representation. After, we present our CNN-based approaches known as Magnitude-Orientation Stream (MOS) and Depth Weighting Magnitude-Orientation Stream (MOS+D) (Section 3.3). Finally, in Section 3.4, we introduce our skeleton-based approaches named SkeleMotion and Tree Structure Reference Joints Image (TSRJI) which can be classified as skeleton image representations.

3.1 Optical Flow Co-occurrence Matrices

Several spatiotemporal feature descriptors, such as HOG3D [Kläser et al., 2008] and HOF [Laptev et al., 2008], use gradient or optical flow to encode the information extracted from the video. While those works have demonstrated encouraging recognition accuracy, a rich source of information contained in these descriptors, such as spatial relations contained in the optical flow field, have not been fully explored.

To explore the local relations on the optical flow field, we propose a novel spatiotemporal feature descriptor, called OFCM, based on the co-occurrence matrices computed over the optical flow field. These co-occurrence matrices will express the distribution of the magnitude and orientation components at a given offset over the optical flow.

Our hypothesis for designing OFCM is based on the assumption that motion information on a video sequence can be captured by the overall relationship of the vectors in the optical flow. In addition, we believe that such information can be specified by a set of magnitude and orientation dependence matrices computed for various angular relationships and distances between neighboring vector pairs on the video optical flow. Once the co-occurrence matrices have been computed, we use a set of measures known as Haralick textural features [Haralick et al., 1973] to describe the flow patterns.

OFCM is based on the classical textural feature GLCM proposed by Haralick et al. [1973], reviewed in Chapter 2. GLCM estimates the joint distribution of pixel intensity given a distance and an orientation. It is important to emphasize here that in the proposed method, we do not compute the matrices using the image intensity values, but using maps θ^Q e M^Q, which will be discussed in the next paragraphs.

The process of computing the OFCM is illustrated in Figure 3.1 and will be explained in details as follows. First, a dense sampling step is applied to the video dividing it into $n_i \times n_j \times n_t$ regions. These regions are referred to as cuboids and are described by their width (n_i), height (n_j), and length (n_t). Once the cuboids are computed, we build an orientation-magnitude representation derived from the optical flow. Since extracting the optical flow of every pixel is computationally expensive [Ryan et al., 2011], we create a binary mask using absolute difference image between the frame I_t and the frame I_{t+k}, given a threshold h, if the resulting difference is smaller than h, the pixel is discarded; otherwise, the pixel p is set to its corresponding local cuboid C_i. In view of that, we compute the optical flow using the Lucas-Kanade Pyramid [Bouguet, 2000] only for the pixels with the resulting difference higher than h.

As mentioned earlier, OFCM uses optical flow information (orientation and magnitude) to build co-occurrence matrices for each cuboid. To this end, we build two maps for each flow field computed on the cuboid: one based on the orientations of the flow field and the other based on the magnitudes of the flow field. In this way, for each cuboid C_i composed of n_t frames, we compute $n_t - 1$ magnitude maps M and $n_t - 1$ orientation maps θ using

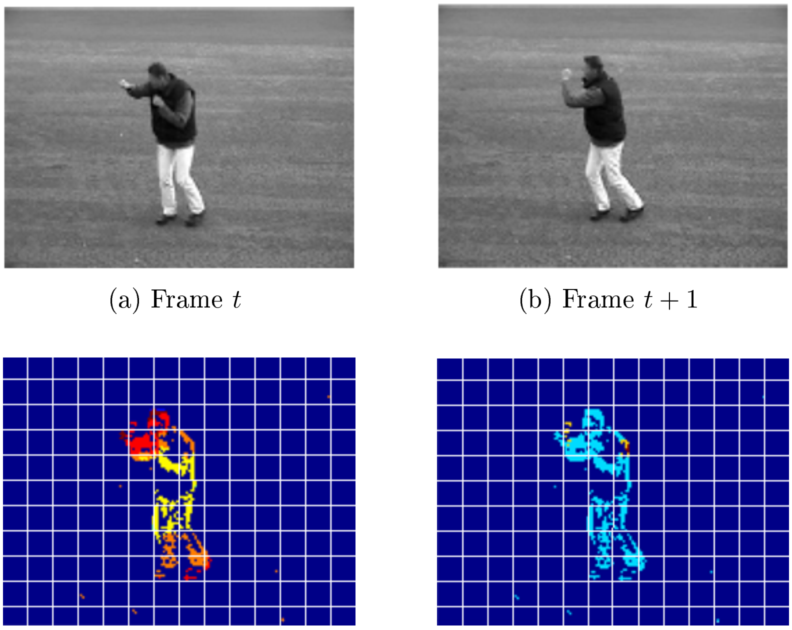
$$M_{i,j} = \sqrt{u_{i,j}^2 + v_{i,j}^2} \tag{3.1}$$

and

$$\theta_{i,j} = tan^{-1}\left(\frac{v_{i,j}}{u_{i,j}}\right), \tag{3.2}$$

where u and v represent the horizontal and vertical components of each flow vector contained in the flow field.

Since the values obtained in M and θ maps are composed of real numbers, a quantization step is applied to compute the co-occurrence matrices of M and θ. The



(a) Frame t

(b) Frame $t+1$

(c) Quantized orientation map θ^Q

(d) Quantized magnitude map M^Q

Figure 3.2. Example of challenges faced in the activity recognition task. Images extracted from UCF101 dataset [Soomro et al., 2012].

magnitude quantization used is based on *dLog* distance proposed by Stehling et al. [2002]. Here, our main goal is to reduce the impact of noisy high magnitude values. The quantization functions are defined as

$$M^Q_{i,j} = \begin{cases} \sigma, & \text{if } m_{i,j} > \sigma \\ 0, & \text{if } m_{i,j} < 0 \\ m_{i,j}, & \text{otherwise} \end{cases} \tag{3.3}$$

$$m_{i,j} = \lfloor \log_2 M_{i,j} \rfloor \tag{3.4}$$

$$\theta^Q_{i,j} = \lfloor \theta_{i,j}/(\Theta/\omega) \rfloor, \tag{3.5}$$

where Θ is the maximum orientation value, σ and ω are the numbers of desired magnitude and orientation bins, respectively. Figure 3.2 illustrates the quantized θ^Q and M^Q maps between two frames. Figure 3.2 (c) illustrates a concatenation of all θ^Q maps extracted from a dense grid of cuboids between a frame t and frame $t+1$. The same is illustrated for the M^Q maps in Figure 3.2 (d).

For each cuboid, we compute α co-occurrence matrices, where α is the number of angles considered ($\pi = 0°$, $45°$, $90°$ and $135°$), from the quantized M^Q and θ^Q maps according to Equation 2.1. After that, we extract f Haralick textural features [Haralick et al., 1973] for each co-occurrence matrix, generating a feature vector with f dimensions per matrix, where f is the number of extracted Haralick features. Finally, all feature vectors are concatenated, providing a final representation with length of $2 \times (\alpha \times f) \times n_t - 1$.

Figure 3.3 shows an example of co-occurrence matrices computed based on the orientations and magnitudes per pixel provided by the optical flow for an horizontal displacement of a single pixel between the frames.

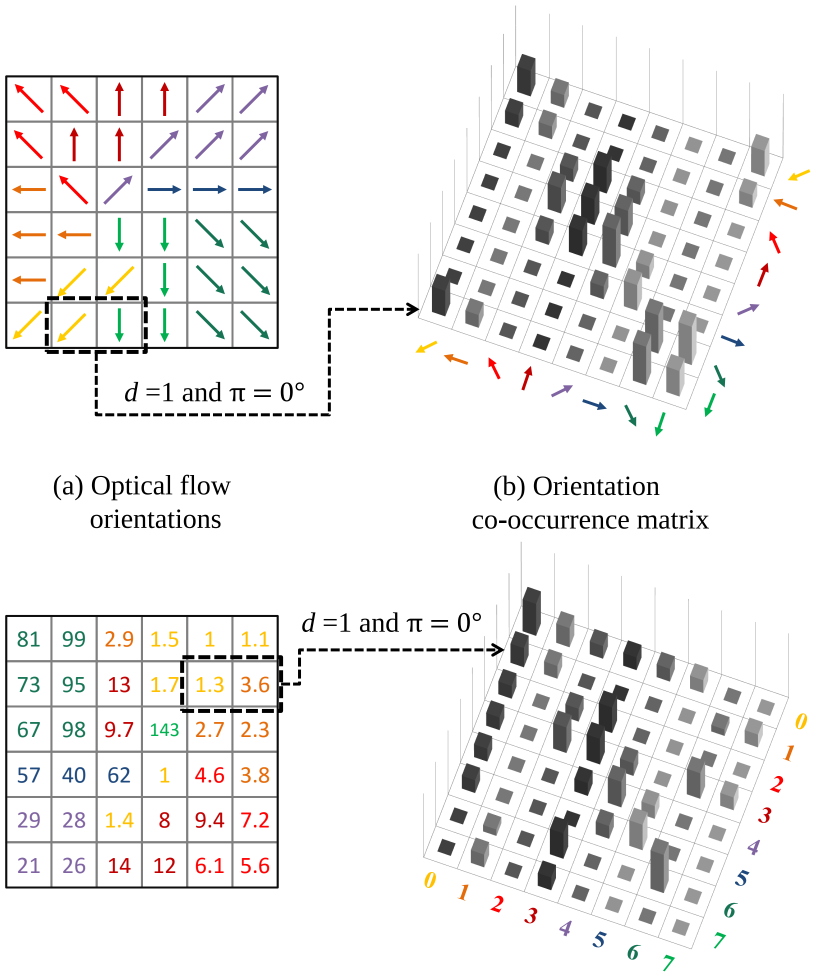
3.2 Co-occurrence of Codewords

Many activity recognition approaches, such as Wang et al. [2011] and Shi et al. [2015], are based on feature extraction followed by a BoW representation to encode the information extracted from the video. Although those works have demonstrated excellent results, a rich source of information contained in mid-level encoding, such as spatial relations contained on the features, has not yet been fully explored.

To explore the local spatial relations contained in the features, we propose a novel spatiotemporal feature representation, called CCW, based on the co-occurrence matrices computed over feature codewords. Such co-occurrence matrices express the distribution of the features at a given offset over feature codewords from a pre-computed codebook. It is important to emphasize the difference between OFCM and CCW, while the former generates co-occurrence matrices from magnitude and orientations extracted from optical flow the later computes the co-occurrence matrices using sampled features (codewords).

Our hypothesis for designing the CCW representation is based on the assumption that the information on a video sequence can be encoded by the overall relationship of the feature codewords from a pre-computed codebook. In addition, we believe that it can be specified by a set of codeword dependence matrices computed for various angular relationships and distances between neighboring codewords on the video. Once the co-occurrence matrices have been computed, we extract the set of measures known as Haralick textural features [Haralick et al., 1973] to describe the patterns. Note that in the proposed method, we do not compute the matrices using the image intensity values, but using quantized features (codewords), as it will be discussed in the next paragraphs.

The process of computing the CCW representation is illustrated in Figure 3.4 and will be explained in details as follows. Similarly to OFCM, a dense sampling step



81	99	2.9	1.5	1	1.1
73	95	13	1.7	1.3	3.6
67	98	9.7	143	2.7	2.3
57	40	62	1	4.6	3.8
29	28	1.4	8	9.4	7.2
21	26	14	12	6.1	5.6

(a) Optical flow orientations

(b) Orientation co-occurrence matrix

(c) Optical flow magnitudes

(d) Magnitude co-occurrence matrix

Figure 3.3. Creation of the co-occurrence matrices from the orientation and magnitude information provided by the optical flow. For this example, the comparison between pairs of pixels is performed considering a 0° (horizontal) neighbor with distance $d = 1$. Maps in (a) and (c) represent a region of a frame considering orientations and magnitudes derived from the flow field, respectively. (b) and (d) represent the co-occurrence matrices computed from the optical flow orientation and magnitude, respectively (the latter considers a logarithm quantization).

is applied to the video dividing it into cuboids described by their width (n_i), height (n_j), and length (n_t). With the cuboids at hand, we apply a spatiotemporal feature descriptor to describe the information of each cuboid.

Since the features obtained from the cuboids are composed of real valued vectors, a quantization step is applied to compute the co-occurrence matrices. The feature quantization used is based on a codebook, or visual dictionary. Let $\mathcal{C}$ be a visual codebook obtained by an unsupervised learning algorithm (e.g., k-means clustering

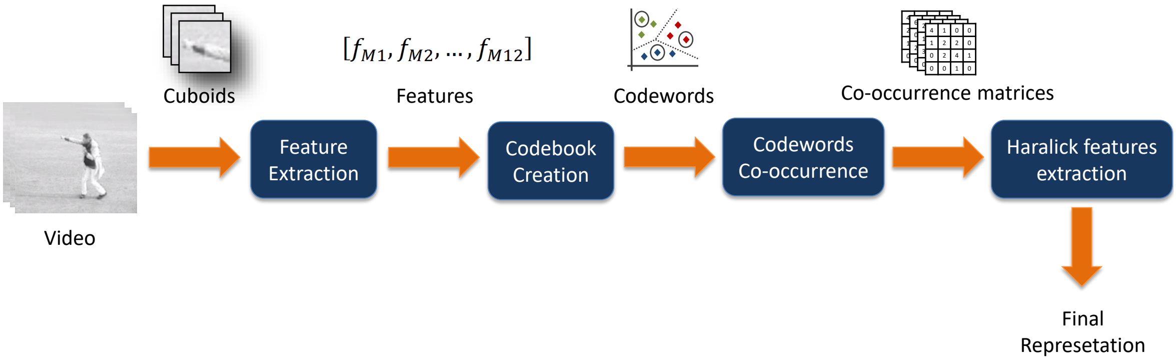


Figure 3.4. Diagram illustrating the pipeline extraction of the proposed spatiotemporal feature representation.

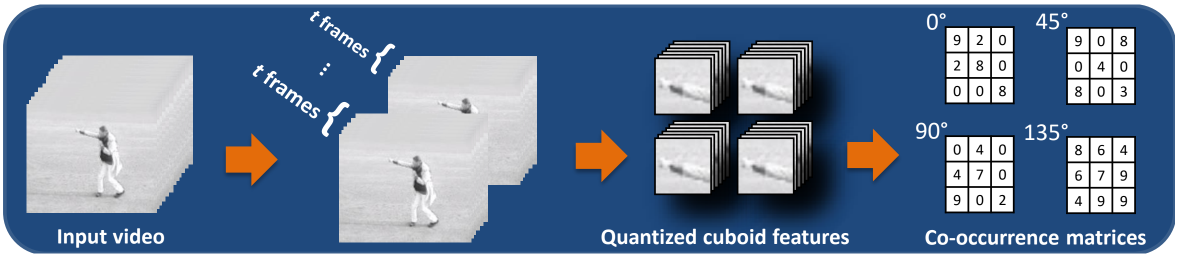


Figure 3.5. Detailed "Codewords Co-occurrence" illustration.

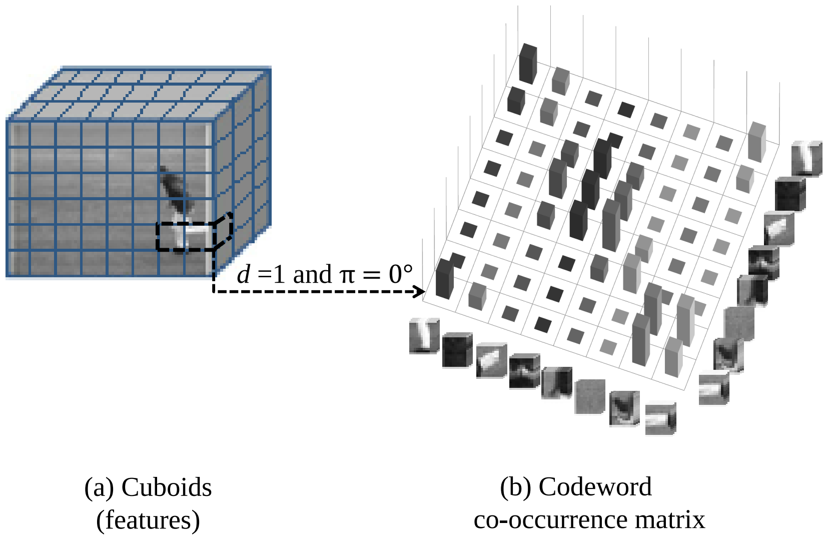
algorithm). $\mathcal{C} = \{\mathbf{c}_k\}$, $k \in \{1, \dots, K\}$, where $\mathbf{c}_k \in \mathbb{R}^D$ is a codeword and K is the number of codewords.

Given a video, we compute α co-occurrence matrices for each t frames from the quantized features (codewords) according to

$$\Sigma_{\Delta_x,\Delta_y}(i,j) = \sum_{r=1}^{n}\sum_{q=1}^{m} \begin{cases} 1, & \text{if } I(r,q) = i \text{ and} \\ & I(r+\Delta_x, q+\Delta_y) = j, \\ & \\ 0, & \text{otherwise} \end{cases}$$

Here, α is the number of angles used ($\pi = 0°$, $45°$, $90°$ and $135°$) and t is related to the temporal length of the cuboids. Then, the co-occurrence matrices are accumulated according to their angles. After that, we extract f Haralick textural features for each co-occurrence matrix, generating a feature vector with f dimensions per matrix, where f is the number of extracted Haralick features. Finally, all feature vectors are concatenated in $X = \{x_i\}, i \in \{1, ...N\}$, followed by a normalization step. The z score norm is applied as

$$z_i = \frac{x_i - \mu_i}{\sigma_i}, \tag{3.6}$$



(a) Cuboids (features)

(b) Codeword co-occurrence matrix

Figure 3.6. Creation of a co-occurrence matrix from feature codewords. For this example, the comparison between pairs of cuboids is performed considering a 0° (horizontal) nearest neighbor with distance $d = 1$. (a) Set of cuboids (features) densely extracted over the video. (b) Co-occurrence matrix computed from the feature codewords.

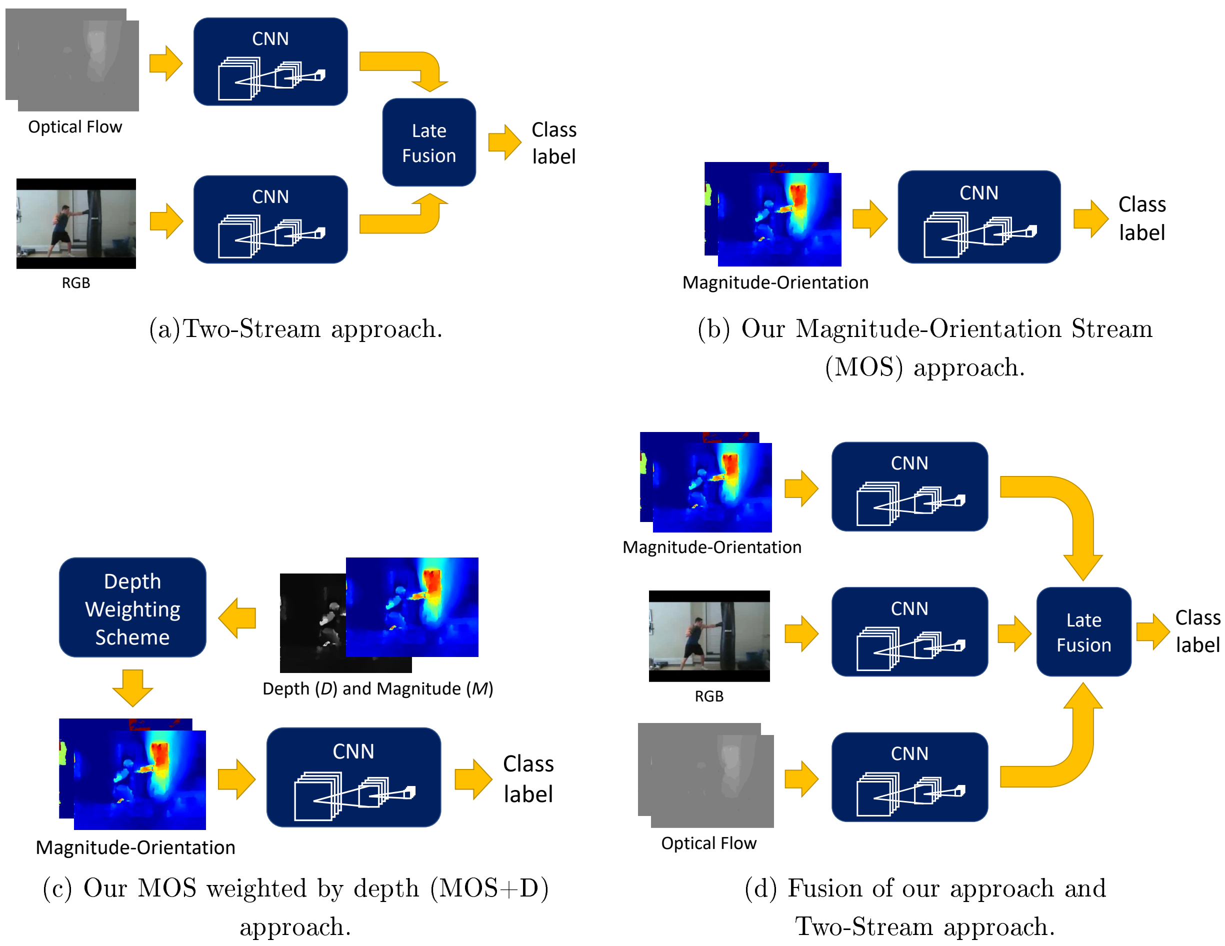
where $x_i \in \mathbb{R}^D$ represents each dimension of the concatenated vectors, N is the length of $\alpha \times f$, μ_i is the mean value of the dimension i and σ_i is standard deviation of each dimension i. Such process is the "Codewords Co-occurrence" block shown in Figure 3.4 and illustrated with more details in Figure 3.5.

We also present two variations of the CCW representation: (i) a simply concatenation of the vectorized co-occurrence matrices with no Haralick feature extraction (inspired by the work from Nosaka et al. [2012]); and (ii) a combination of the Haralick feature extraction and the concatenation of the vectorized co-occurrence matrices. The vectorized matrices can be seen as a histogram in which each bin is composed of a pair of features instead of just a single feature.

Figure 3.6 shows an example of co-occurrence matrices computed based on a pre-computed codebook for an horizontal displacement of codewords between frames.

3.3 Magnitude-Orientation Stream Network

In this section, we present our approaches for performing activity recognition with the proposed MOS and MOS+D. For completeness, we first present the basic concepts of the network architectures that we use to learn data representation. Then, we detail the MOS method showing how to incorporate magnitude and orientation as temporal



(a)Two-Stream approach.

(b) Our Magnitude-Orientation Stream (MOS) approach.

(c) Our MOS weighted by depth (MOS+D) approach.

(d) Fusion of our approach and Two-Stream approach.

Figure 3.7. Architectures considered in this work for extracting spatiotemporal information.

information for the network input. Finally, aiming at compensating for the distance of the subjects performing the activity to the camera, we explain the approach used to estimate the depth information from monocular videos [Godard et al., 2017] and how we employ it as a magnitude weighting scheme, which we call MOS+D.

3.3.1 Employed Architectures

In this section, we present the basic concepts of the Very Deep Two-Stream network (VD2S) [Wang et al., 2015] and Temporal Segment Networks (TSN) [Wang et al., 2016a], which are the baseline network architectures we use to learn the data representation based on the magnitude and orientation.

3.3.1.1 Very Deep Two-Stream

Motivated by the successful results achieved by deep architectures (e.g., VGG-16) in the object recognition task, Wang et al. [2015] improved the two-stream network by adapting it to use the VGG-16 on activity recognition, which they called VD2S. As mentioned in Chapter 2, the two-stream network is composed of two different networks receiving distinct flows of data, spatial and temporal. The spatial stream receives as input the RGB frames while the temporal stream receives an optical flow image as input.

The spatial network is built on a single frame, therefore its architecture is the same as those used for object recognition on the image domain. Thus, at each iteration of the training step, 256 training videos are uniformly sampled across the classes and a single frame is randomly selected. Moreover, to avoid overfitting, the authors employ two data augmentation techniques: (i) cropping and flipping the four corners and center of the frame; and (ii) a multi-scale cropping method that randomly samples the cropping width and height from 256, 224, 192, 168. Finally, they resize the cropped regions to $224 \times 224 \times 3$.

The temporal network receives images of optical flow as input. The process for computing the optical flow is the following. For each frame F at time t, the optical flow O_t is computed considering F_t and F_{t+1}. The resulting optical flow O_t is composed of two channels: (i) $\mathcal{O}_t^x$, denoting an image containing the x (horizontal) displacement field; and (ii) $\mathcal{O}_t^y$, denoting an image containing the y (vertical) displacement field. Moreover, to avoid storing the displacement fields as floats, the horizontal and vertical components of the flow are linearly rescaled to a [0, 255] interval as

$$\mathcal{I}^f_{t_{i,j}} = \begin{cases} 0, & \text{if } \mathcal{O}^f_{t_{i,j}} < l \\ 255, & \text{if } \mathcal{O}^f_{t_{i,j}} > h \\ 255 \times \frac{(O^f_{t_{i,j}} - l)}{(h-l)}, & \text{otherwise} \end{cases}, \tag{3.7}$$

where f represents the image channel (flow component x or y), h is the higher bound maximum optical flow value, l is the lower bound minimum optical flow value and $\mathcal{I}^f$ the optical flow image. The same data augmentation techniques used in spatial network are used in the temporal stream. Finally, the input of the temporal network is generated by stacking 10 randomly images $\mathcal{I}^f$ of optical flow fields ($224 \times 224 \times 20$) [Simonyan & Zisserman, 2014].

To combine the two networks, a late fusion scheme is employed by using a weighted linear combination of their prediction scores. Hence, the weight is set as

2 for the temporal network and 1 for the spatial network to give more importance to the temporal information. Figure 3.7(a) illustrates the Deep Two-Stream network.

3.3.1.2 Temporal Segment Networks

Most CNN frameworks focus their learning methods on short-term motions by working on single stack of frames, thus lacking the capacity to incorporate long-range temporal structure. To learn a video representation that is able to capture such structure, Wang et al. [2016a] developed the Temporal Segment Networks (TSN), which extract short snippets over the video by employing a sparse sampling scheme to capture the long-range temporal structure.

The basic idea of the work proposed by Wang et al. [2016a] is the following. A video is divided into K segments and for each segment, their approach randomly samples T snippets used as inputs for a two-stream network. After the predictions of each snippet, the authors employ a fusion scheme by an aggregation function (averaging, maximum or weighted averaging). Finally, a softmax layer is applied to predict the probability of activity class for the whole video.

Regarding the network architecture, Wang et al. [2016a] adapted the inception with batch normalization (BN-Inception) to the design of two-stream following the same input scheme as the very deep two-stream [Wang et al., 2015]: (i) spatial stream, receives RGB images; and (ii) temporal stream operates on a stack of optical flow images. Moreover, they employed the same data augmentation techniques and late fusion scheme to perform the combination of the two networks as in very deep two-stream [Wang et al., 2015].

3.3.2 Magnitude-Orientation Stream (MOS)

Our MOS follows the same fundamentals as the two-stream networks. However, aiming at extracting more information from the optical flow, MOS captures the displacement information by using the orientation of the optical flow and the velocity of the movement considering the optical flow magnitude. The spatial relationship contained on local neighborhoods of magnitude and orientation captures not only displacement by using orientation, but also magnitude, providing information regarding the velocity of the movement. The method is based on non-linear transformations on the optical flow components to generate input images for the temporal stream. To incorporate such information on the temporal stream, we compute the dense optical flow as Wang et al. [2015]. In this way, for each video composed of n frames, we compute $n-1$ optical flows $\mathcal{O}$. Once the optical flow is available, we compute the magnitude and orientation information as

$$M_{i,j} = \sqrt{(\mathcal{O}^x_{i,j})^2 + (\mathcal{O}^y_{i,j})^2} \tag{3.8}$$

and

$$\theta_{i,j} = tan^{-1}\left(\frac{\mathcal{O}^y_{i,j}}{\mathcal{O}^x_{i,j}}\right), \tag{3.9}$$

where M and θ are magnitude and orientation information, respectively.

Since the values obtained in M and θ are composed of real numbers, they are linearly rescaled to a [0, 255] using Equation 3.7. Moreover, since the orientation values are estimated for every pixel of the optical flow, they can generate noisy values from regions of the image without any movement. Therefore, we perform a filtering on θ based on the values of M as

$$\theta^{'}_{i,j} = \begin{cases} 0, & \text{if } M_{i,j} < m \\ \theta_{i,j}, & \text{otherwise} \end{cases}, \tag{3.10}$$

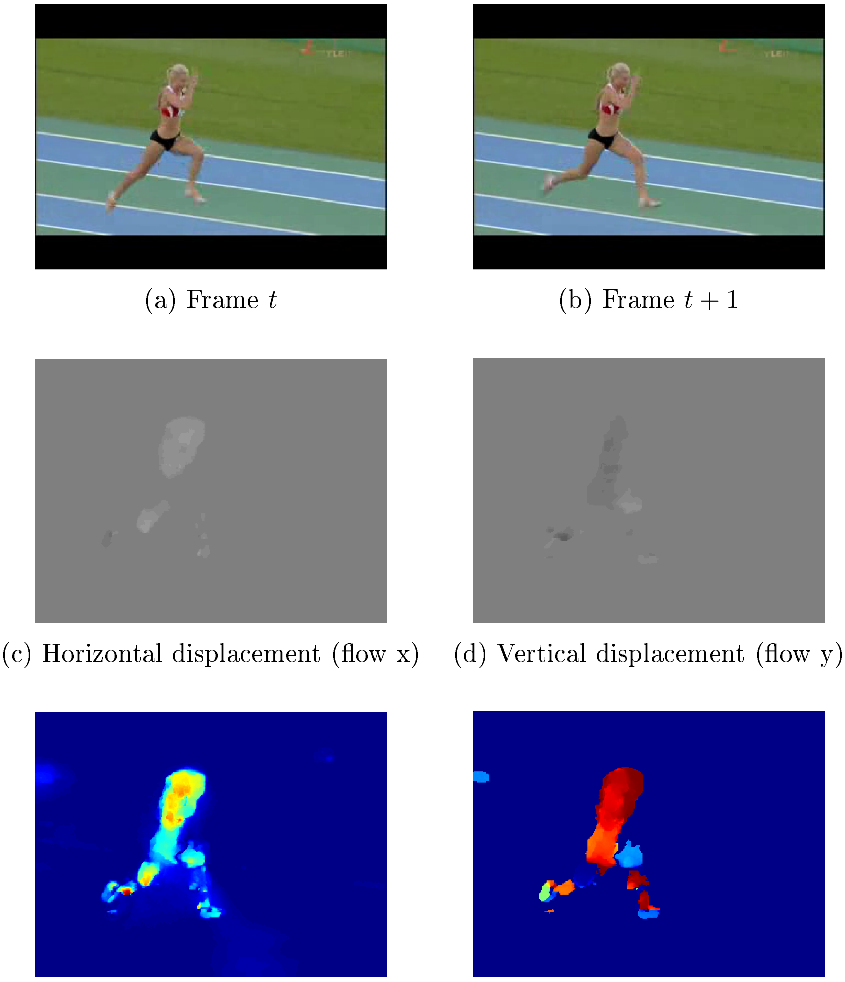
where m is a magnitude threshold value. Figure 3.8 illustrates a comparison between the magnitude and orientation information with the optical flow x and y displacements extracted from two consecutive frames.

With the rescaled magnitude and orientation information, which can be seen as two image channels, we use the same data augmentation techniques as Wang et al. [2015]. Therefore, the input is composed of 10 stacked images (224 × 224 × 20). Figure 3.7(b) illustrates the MOS network stages.

3.3.3 Depth Information Estimation

The use of depth information has shown several advantages in a number of visual recognition tasks including human activity recognition [Wang et al., 2014]. Compared with RGB video sequences, depth information has shown several advantages in the context of activity recognition. For instance, Liang & Zheng [2015] claim that depth data can provide 3D structural information so that the motion information of activities can be more discriminative.

Our main goal on using depth information is to circumvent problems related to activities taken considering their distance in relation to the camera. As an example, the “BandMarching” class in the UCF101 dataset [Soomro et al., 2012]. Figure 3.9(c) shows the magnitude information extracted from a “BandMarching” video. As it can be seen, although every person in the scene should have similar magnitude/velocity information, people closer to the camera present much higher magnitude values than the ones distant from the camera. Such difference happens because the pixel displacement near the



(a) Frame t

(b) Frame $t+1$

(c) Horizontal displacement (flow x)

(d) Vertical displacement (flow y)

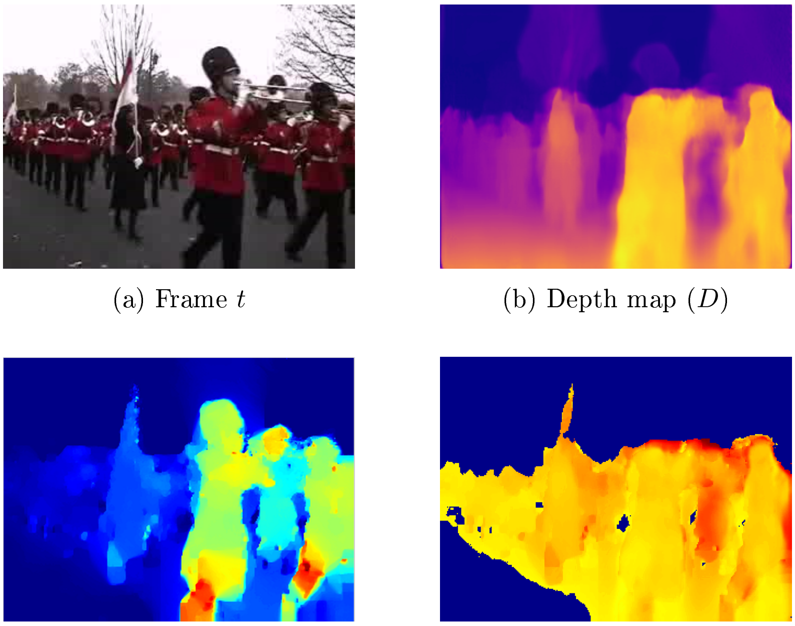
(e) Magnitude (M)

(f) Orientation (θ')

Figure 3.8. Comparison between optical flow displacement information and magnitude and orientation extracted from two consecutive frames (t and $t+1$) of an activity sample extracted from the UCF101 dataset [Soomro et al., 2012].

camera is much higher than the displacement of distant pixels. To circumvent this problem, we apply a normalization scheme to the magnitude information by weighting the magnitude by the depth information.

Since the videos from classic activity recognition datasets, such as UCF101 [Soomro et al., 2012] and HMDB51 [Kuehne et al., 2011] were not recorded using a depth sensor to capture depth information, here we extract it from the RGB data employing a fast state-of-the-art approach [Godard et al., 2017] to estimate depth data from monocular views. In this way, for each video composed of n frames, we compute n depth maps D. Once the depth maps are available, we first apply a Gaussian filter on each depth map with the aim of softening erroneous estimates and then we



(a) Frame t

(b) Depth map (D)

(c) Magnitude (M)

(d) Magnitude weighted by depth ($M^{'}$)

Figure 3.9. Comparison between magnitude information and the weighted magnitude by depth extracted from an activity sample extracted from the UCF101 dataset [Soomro et al., 2012].

weight the magnitude information as

$$M^{'}_{i,j} = \begin{cases} M\{i,j\} \times (D_{i,j} + 1), & \text{if } D_{i,j} < d \\ 0, & \text{otherwise} \end{cases}, \tag{3.11}$$

where d is the depth threshold value used. The intuition for using such threshold lies on the premise that activities of interest being performed in the video usually do not take place in the background. Therefore, we can filter noisy movements that are not of interest. Then, the weighted magnitude values are linearly rescaled to a [0, 255] using Equation 3.7.

Figure 3.8 illustrates the original magnitude information and the weighted magnitude information by depth. As it can be seen, some magnitude information is lost due to erroneous estimations on the depth map (hat and head of the person in front). After that, the weighted magnitude information and orientation are used as input for a CNN. Figure 3.7(c) illustrates the MOS weighted by depth stages, which we call MOS+D.

Finally, to incorporate spatial information to our approach, we employ a late fusion technique with the two-stream network (by employing VD2S [Wang et al., 2015] or TSN [Wang et al., 2016a]), as illustrated in Figure 3.7(d).

3.4 Skeleton-Based Approaches

In this section, we present our skeleton-based approaches for performing activity recognition named SkeleMotion and TSRJI. We first present the basic concepts of the network architecture employed by us to learn data representation. Then, we detail the SkeleMotion method showing how to incorporate magnitude and orientation information from skeleton data. The SkeleMotion approach encodes temporal dynamics by explicitly using motion information by computing the magnitude and orientation values of the skeleton joints. To that end, different temporal scales are used to filter noisy motion values and to aggregate more temporal dynamics to the representation. Such temporal scales enable to capture long-range joint interactions involved in activities. Moreover, the method takes advantage of a structural organization of joints that preserves spatial relations of more relevant joint pairs. Finally, we introduce our TSRJI representation that encodes temporal dynamics by combining the use of reference joints [Ke et al., 2017] and a tree structure skeleton [Yang et al., 2018]. This approach also takes advantage of a structural organization of joints that preserves spatial relations of more relevant joint pairs and also incorporates different spatial relationships between the joints.

3.4.1 Convolutional Neural Network Architecture Employed

We adopted a modified version of the CNN architecture proposed by Li et al. [2017] to learn the features of the generated skeleton image representations. They designed a small convolutional neural network which consists of three convolution layers and four Fully-Connected (FC) layers. However, we modified this network to a smaller version with the original convolutional layers and only two FC layers. All convolutions have a kernel size of 3×3, the first and second convolutional layers with a stride of one and the third one with a stride of two. Max pooling, ReLU neuron and dropout regularization ratio are adopted. We opted for using such architecture since it demonstrated good performance and, according to the authors, it can be easily trained from scratch without any pre-training and is superior because of its compact model size and fast inference speed. Figure 3.10 presents an overview of the employed architecture.

To handle activities involving multi-person interaction (e.g., shaking hands), we apply a common procedure in the literature, which is to stack skeleton image representations of different people as the network input.

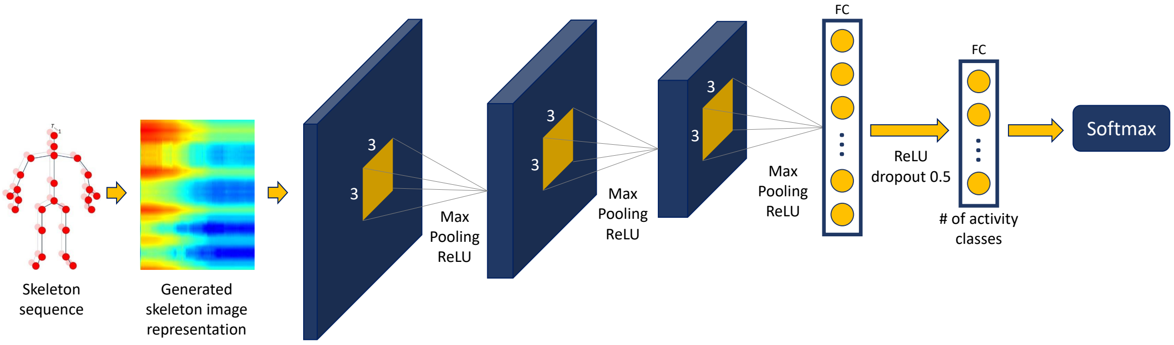


Figure 3.10. Network architecture employed for 3D activity recognition.

3.4.2 SkeleMotion

As reviewed in Section 2.4.2, the majority of works that encode skeleton data as image representations are based on spatial structure encoding of the skeleton joints. According to Li et al. [2018], temporal movements of joints can also be used as crucial cues for activity recognition and although the temporal dynamics of the sequence can be implicitly learned by using a CNN, an explicit modeling can produce better recognition accuracies.

Motivated by our MOS approach, we propose a novel skeleton image representation (named SkeleMotion), based on magnitude and orientation of the joints to explore the temporal dynamics. Our approach expresses the displacement information by using orientation encoding (direction of joints) and magnitude to provide information regarding the velocity of the movement. Furthermore, due to the successful results achieved by the skeleton image representations, our approach follows the same fundamentals by representing the skeleton sequences as matrices. First, we apply the depth-first tree traversal order [Yang et al., 2018] to the skeleton joints to generate a pre-defined chain order C that best preserves the spatial relations between joints in original skeleton structures[1]. Afterwards, we compute a matrix S that corresponds to a chain of concatenated skeleton joint coordinates from the frame t. In view of that, each column of the matrix corresponds to the temporal evolution of the arranged chain joint c. At this point, the size of matrix S is $C \times T \times 3$, where C is the number of joints of the chain, T is the total frame number of the video sequence and 3 is the number joint coordinate axes (x, y, z). Then, we create the motion structure $\mathcal{D}$ as

$$\mathcal{D}_{c,t} = S_{c,t+d} - S_{c}, \tag{3.12}$$

[1] Chain C considering 25 Kinect joints: [2, 21, 3, 4, 3, 21, 5, 6, 7, 8, 22, 23, 22, 8, 7, 6, 5, 21, 9, 10, 11, 12, 24, 25, 24, 12, 11, 10, 9, 21, 2, 1, 13, 14, 15, 16, 15, 14, 13, 1, 17, 18, 19, 20, 19, 18, 17, 1, 2], as defined in [Yang et al., 2018].

where each matrix cell is composed of the temporal difference computation of each joint between two frames of d distance, resulting in a $C \times T - d \times 3$ matrix.

We build two different representations using the proposed motion structure $\mathcal{D}$: one based on the magnitudes of joint motions and another one based the orientations of the joint motion. We compute both representations using

$$M_{c,t} = \sqrt{(\mathcal{D}^x_{c,t})^2 + (\mathcal{D}^y_{c,t})^2 + (\mathcal{D}^z_{c,t})^2} \tag{3.13}$$

and

$$\begin{aligned} \theta_{c,t} &= \text{stack}(\theta^{xy}_{c,t}, \theta^{yz}_{c,t}, \theta^{zx}_{c,t}) \\ \theta^{xy}_{c,t} &= \tan^{-1}\left(\frac{\mathcal{D}^y_{c,t}}{\mathcal{D}^x_{c,t}}\right), \\ \theta^{yz}_{c,t} &= \tan^{-1}\left(\frac{\mathcal{D}^z_{c,t}}{\mathcal{D}^y_{c,t}}\right), \\ \theta^{zx}_{c,t} &= \tan^{-1}\left(\frac{\mathcal{D}^x_{c,t}}{\mathcal{D}^z_{c,t}}\right), \end{aligned} \tag{3.14}$$

where M is the magnitude skeleton representation of size $J \times T - d \times 1$ and θ is the orientation skeleton representation of size $J \times T - d \times 3$ (composed of 3 stacked channels).

Since the orientation values are estimated for every joint, it might generate noisy values for joints without any movement. Therefore, we perform a filtering on θ based on the values of M as

$$\theta^{'}_{c,t} = \begin{cases} 0, & \text{if } M_{c,t} < m \\ \theta_{c,t}, & \text{otherwise} \end{cases}, \tag{3.15}$$

where m is a magnitude threshold value.

Finally, the generated matrices are normalized to [0, 1] and empirically resized to a fixed size of $C \times 100$, since number of frames may vary depending on the skeleton sequence of each video. Figure 3.11 gives an overview of our method for building the SkeleMotion representation.

3.4.2.1 Temporal Scale Aggregation (TSA)

Skeleton image representations in the literature encode joint coordinates as channels. This process may cause a problem that the co-occurrence features are aggregated locally, becoming unable to capture long-range joint interactions involved in activities [Li et al., 2018]. Moreover, one drawback of encoding motion values of joints is the noisy

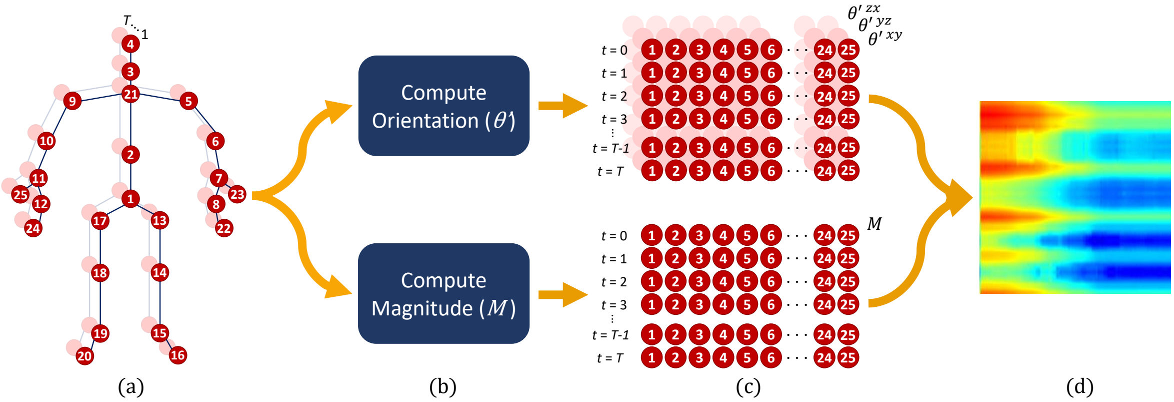


Figure 3.11. SkeleMotion representation. (a) Skeleton data sequence of T frames. (b) Computation of the magnitude and orientation from the joint movement. (c) $\theta^{'}$ and M arrays: each row encodes the spatial information (relation between joint movements) while each column describes the temporal information for each joint movement. (d) Skeleton image after resizing and stacking of each axes.

values that can be introduced to the representation due to the small distance d between two frames. For instance, if the computation is performed considering two consecutive frames, it could add to the representation unnecessary motion of joints that are irrelevant for predicting a specific activity (e.g., motion of the head joint on a handshake activity).

To overcome the aforementioned problems, we propose a variation of our SkeleMotion representation that pre-computes the motion structure $\mathcal{D}$ considering different d distances. For each of the motion structures $\mathcal{D}$, we compute its respective magnitude skeleton representation M and then stack them all into one single representation. The same idea is applied to compute the orientation skeleton representation θ, however a weighting scheme is applied during the filtering process aforementioned, as

$$\theta^{'}_{c,t} = \begin{cases} 0, & \text{if } M_{c,t} < m \times d \\ \theta_{c,t}, & \text{otherwise} \end{cases} . \tag{3.16}$$

Such technique adds more temporal dynamics to the representation by explicitly showing temporal scales to the network. In this way, the network can learn which movements are relevant for learning the activity and also being able to capture long-range joint interactions.

3.4.3 Tree Structure Reference Joints Image (TSRJI)

As reviewed in Section 2.4.2, a crucial step to achieve good performance using skeleton image representations is the definition of how to build the structural organization of the representation preserving the spatial relations of relevant joint pairs. In view of that, and due to the successful results achieved by the skeleton image representations, our approach follows the same fundamentals and represents the skeleton sequences as a matrix. Furthermore, our method is based on two premises of successful representations of the literature: (i) the fewer edges there are, the more relevant the joint pair is [Yang et al., 2018]; and (ii) different spatial relationships between the joints leads to less sparse data [Ke et al., 2017].

To address the first premise, we apply the depth-first tree traversal order [Yang et al., 2018] to each skeleton data from frame t to generate a pre-defined chain order C^t that best preserves the spatial relations between joints in original skeleton structures (see Figure 3.12). The assumption here is that spatially related joints in original skeletons have direct graph links between them [Yang et al., 2018]. The less edges required to connect a pair of joints, the more related is the pair. In view of that, with the C^t chain order, the neighboring columns in skeleton images are spatially related in original skeleton structures.

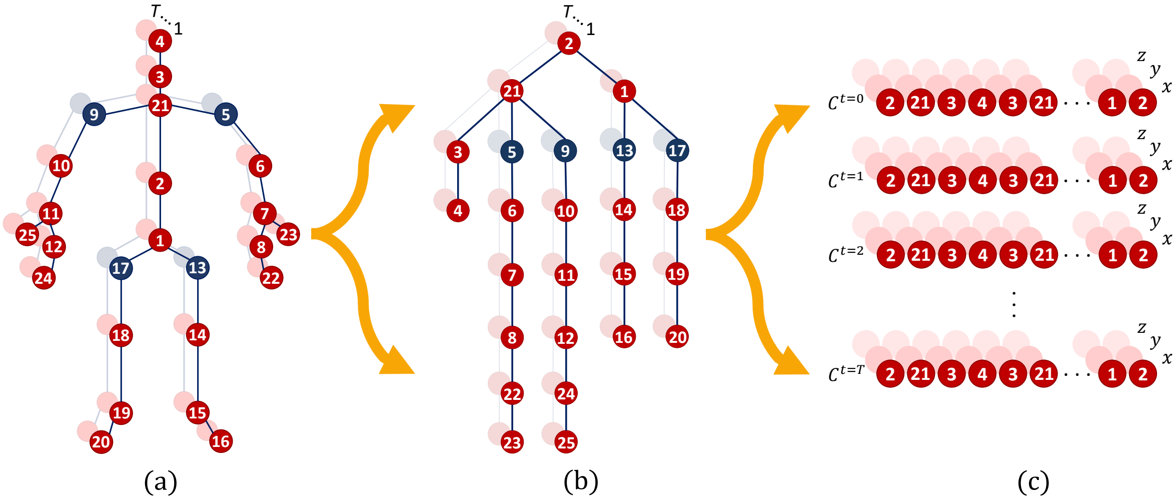


Figure 3.12. Depth-first tree traversal order applied to skeleton data. (a) Skeleton data sequence of T frames. (b) Tree used to apply the depth-first tree traversal algorithm. (c) Generated chains C^t considering 25 Kinect joints: [2, 21, 3, 4, 3, 21, 5, 6, 7, 8, 22, 23, 22, 8, 7, 6, 5, 21, 9, 10, 11, 12, 24, 25, 24, 12, 11, 10, 9, 21, 2, 1, 13, 14, 15, 16, 15, 14, 13, 1, 17, 18, 19, 20, 19, 18, 17, 1, 2], as defined by Yang et al. [2018].

To address the second premise, we apply the reference joints technique [Ke et al.,

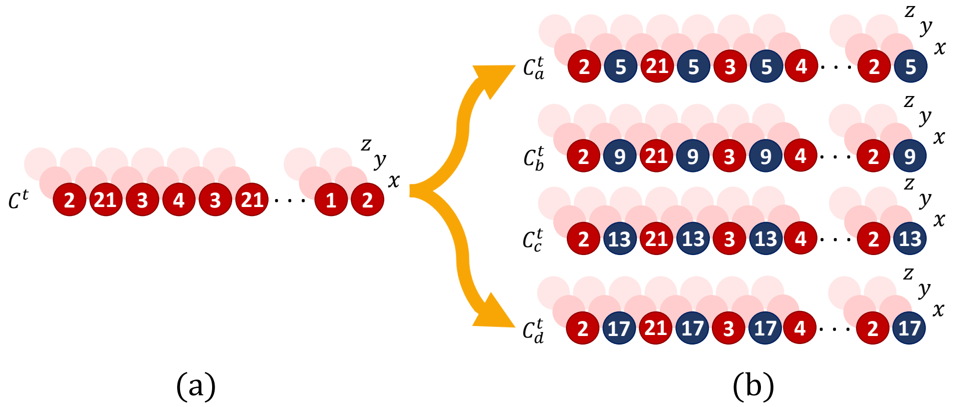


Figure 3.13. Reference joints technique applied to skeleton data. (a) Chain C^t considering 25 Kinect joints. (b) Generated chains C_a^t, C_b^t, C_c^t, C_d^t considering the reference joints (dark painted joints).

2017] to each generated C^t chain. To that end, four reference joints are respectively used to compute relative positions of the other joints: (a) the left shoulder; (b) the right shoulder; (c) the left hip; and (d) the right hip. Thus, at this point, we have four C chains for each skeleton of each frame (i.e., C_a^t, C_b^t, C_c^t, C_d^t). The hypothesis here, introduced by Ke et al. [2017], is that relative positions between joints provide more useful information than their absolute locations. According to Ke et al. [2017], these four joints are selected as reference joints due to the fact that they are stable in most actions, thus reflecting the motions of the other joints. Figure 3.13 illustrates the reference joints technique computation.

After dealing with the aforementioned premises, we compute four matrices S (one for each reference joint) that correspond to the concatenation of the chains C^t from a video (i.e., S_a, S_b, S_c, S_d), where each column of each matrix denotes the temporal evolution of the arranged chain joint c. At this point, the size of matrix S is $J \times T \times 3$, where J is the number of joints of the any reference joint chain C^t, T is the total frame number of the video sequence and 3 is the number joint coordinate axes (x, y, z).

Finally, the generated matrices are normalized to [0, 1] and empirically resized to a fixed size of $J \times 100$ that are used as input to CNNs, since number of frames may vary depending on the skeleton sequence of each video. Figure 3.14 gives an overview of our method for building the skeleton image representation.

3.5 Concluding Remarks

In the activity recognition problem, the accurate representations are based on motion analysis. In this chapter, we proposed four different representations built upon motion

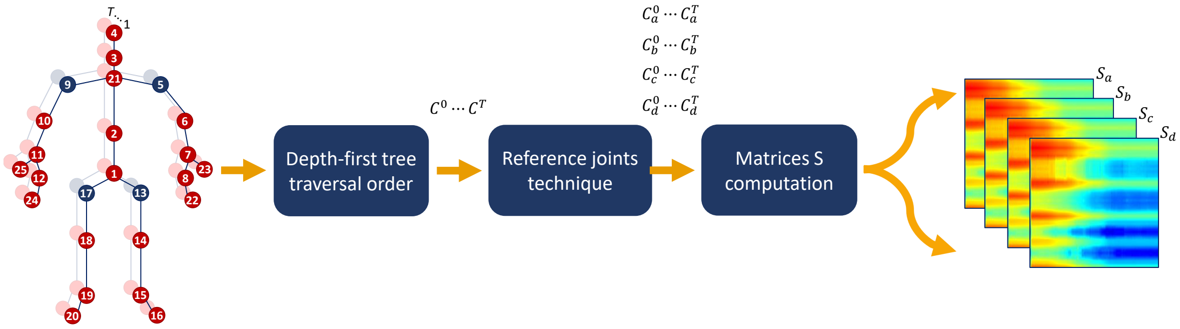


Figure 3.14. Proposed skeleton image representation.

information: (i) the local feature descriptor, OFCM; (ii) a mid-level representation, CCW; (iii) image domain CNN-based approaches, MOS and MOS+D; and (iv) skeleton domain CNN-based approaches, SkeleMotion and TSRJI.

Our proposed OFCM (Section 3.1) and CCW (Section 3.2) representations are based on a common technique found in the literature used to extract information related to global structures in various local region known as co-occurrence matrices. In the former, the co-occurrence matrices are computed over the magnitude and orientation information obtained from the optical flow and describes the distribution of the motion according to velocity and direction. In the latter, the matrices are calculated over feature codewords sampled from a pre-computed codebook and express the distribution of the features at a given offset.

The MOS and MOS+D (Section 3.3) are image domain representations based on deep learning. They use a novel feeding scheme for CNNs that uses images computed from the optical flow. MOS applies simple nonlinear transformations on the vertical and horizontal components of the optical flow to generate the input images based on magnitude and orientation. MOS+D incorporates depth information as a magnitude weighting scheme to compensate the distance of the subjects performing the activity to the camera. As in OFCM, both approaches learn to describe the activities by employing velocity and direction information of the motion.

The SkeleMotion (Section 3.4.2) and TSRJI (Section 3.4.3) are also deep learning representations, however they were constructed for the skeleton domain. Both approaches use skeleton image representations as input of CNNs. SkeleMotion is based on temporal dynamics encoding by explicitly using motion information (magnitude and orientation) of the skeleton joints. We further propose a variation of the magnitude skeleton representation considering different temporal scales to filter noisy motion values as well as aggregating more temporal dynamics to the representation. On the other hand, TSRJI takes advantage of a structural organization of joints that preserves spatial relations of more relevant joint pairs and also incorporates different spatial relationships between the joints.

Chapter 4

Benchmarks Addressed

Thanks to the availability of several benchmark datasets, significant progress has been achieved in the activity recognition task. Such datasets provide standard evaluation protocols with labeled data granting researchers a common way to compare their developed approaches.

This chapter introduces the datasets we used in this dissertation. The chapter presents details regarding each used dataset including information such as number of classes, amount of training and testing data, and the challenges provided by each dataset. Except for the KTH dataset, all other datasets used in this dissertation are composed of the four different levels of human activity categorized by Aggarwal & Ryoo [2011] - i.e., gestures, actions, interactions, and group activities.

4.1 KTH Dataset

KTH [Schuldt et al., 2004] is a well-known and publicly available dataset for activity recognition. The dataset is composed of the first two different levels of human activity categorized by Aggarwal & Ryoo [2011], gestures and actions. It consists of six human activity classes: walking, jogging, running, boxing, waving and clapping. Each action is performed several times by 25 subjects in four different scenarios. In total, the dataset consists of 600 videos with spatial resolution of 160×120 pixels. The dataset has an experimental protocol divided into training, validation and test set with nine people for the test set and eight for each of the remaining sets (Table 4.1 summarizes the number of videos for each activity on each set). The performance is evaluated by the metric average accuracy over all classes. Figure 4.1 illustrates the classes of the KTH dataset.

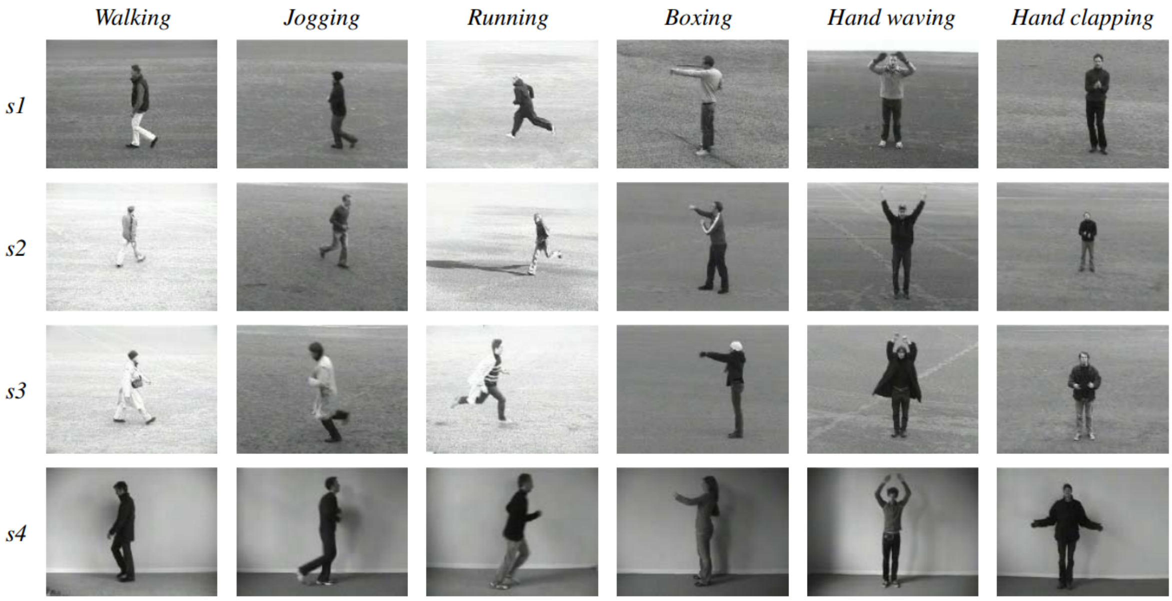


Figure 4.1. Example of activities from different scenarios from KTH dataset [Schuldt et al., 2004]. Figure extracted from Schuldt et al. [2004].

Although the dataset seems to be simple, since it is recorded from four different scenarios, it provides some challenges that include scale variation and illumination variation due to indoor/outdoor scenarios.

Table 4.1. Number of videos for each activity from KTH dataset [Schuldt et al., 2004].

Activity	#Training	#Validation	#Test
1: Boxing	32	32	36
2: Clapping	32	32	36
3: Waving	32	32	36
4: Jogging	32	32	36
5: Running	32	32	36
6: Walking	32	32	36

4.2 UCF Sports Dataset

UCF Sports [Rodriguez et al., 2008] is a realistic dataset that consists of a set of activities collected from various sports. It is composed of ten different types of human activities: swinging (on the pommel horse and on the floor), diving, kicking (a ball), weight-lifting, horse-riding, running, skateboarding, swinging (at the high bar), golf swinging and walking (Figure 4.2 illustrates such activities). It consists of 150 video

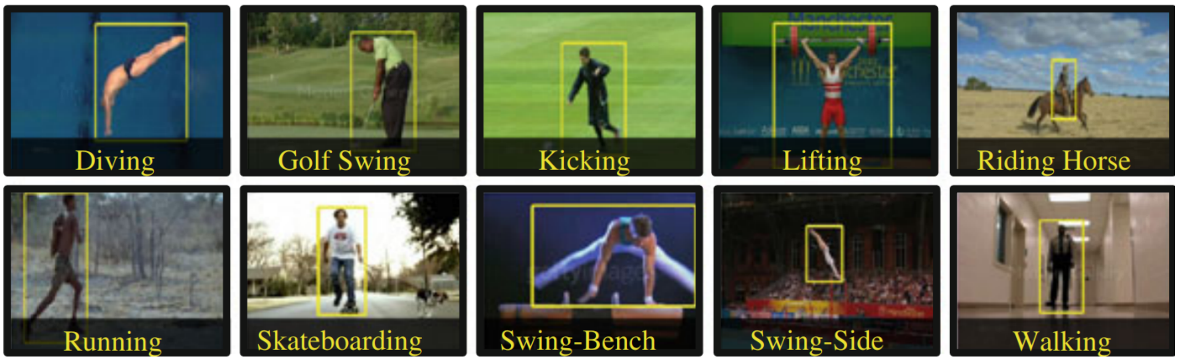


Figure 4.2. Example of activities from UCF Sports dataset [Rodriguez et al., 2008]. Figure extracted from Soomro & Zamir [2014].

samples with a frame rate of 10 fps and spatial resolution of 720 × 480 pixels (Table 4.2 summarizes the number of videos for each activity). In contrast to the KTH dataset, UCF Sports is not divided into sets for training and test. Thus, the protocol used to evaluate the performance of methods is the leave-one-out and the accuracy over all classes metric, as suggested by the authors [Rodriguez et al., 2008].

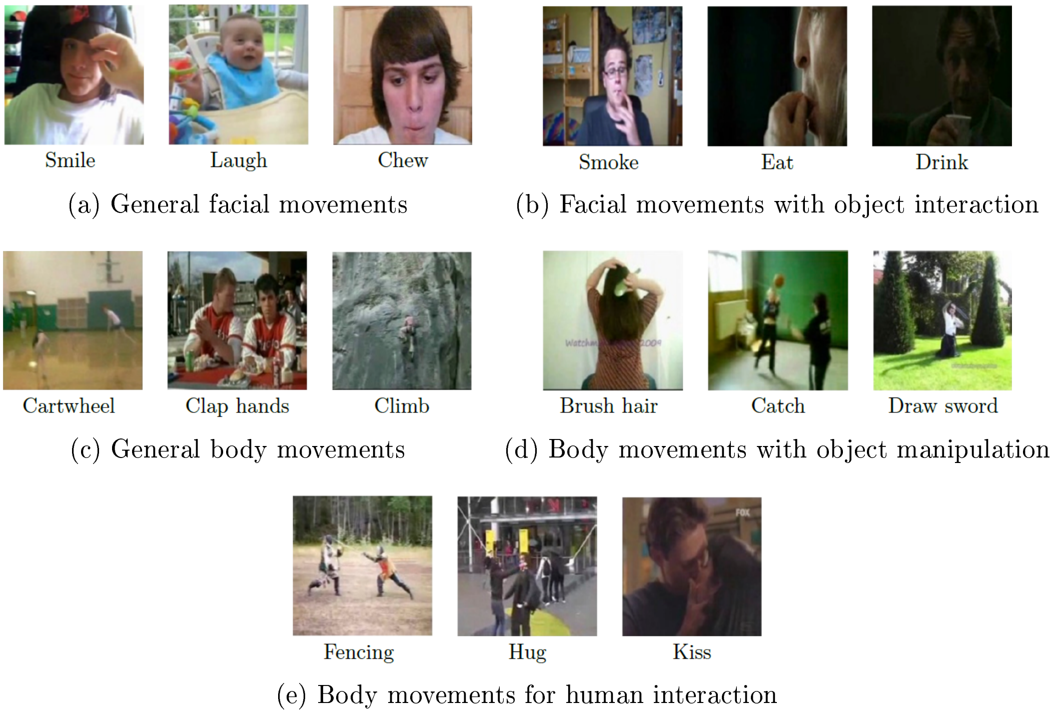
Among the challenges faced in UCF Sports dataset, we can mention the large class variability, occlusions and variations in camera motion.

Table 4.2. Number of videos for each activity from UCF Sports dataset [Rodriguez et al., 2008].

Activity	**#Videos**
1: Diving	14
2: Golf Swing	18
3: Kicking	20
4: Lifting	6
5: Riding Horse	12
6: Running	13
7: SkateBoarding	12
8: Swing-Bench	20
9: Swing-Side	13
10: Walking	22

4.3 HMDB51 Dataset

HMDB51 [Kuehne et al., 2011] is a realistic and challenging activity dataset composed of video clips from movies, the Prelinger archive, Internet, Youtube and Google videos.



(a) General facial movements

(b) Facial movements with object interaction

(c) General body movements

(d) Body movements with object manipulation

(e) Body movements for human interaction

Figure 4.3. Example of activities from the 5 different categories from HMDB51 dataset [Kuehne et al., 2011].

The dataset is comprised of 51 activity categories and consists of 6,766 activity samples with a resolution of 240 pixels in height with preserved aspect ratio.

The activity categories are grouped in five types: (i) general facial movements - e.g.: smile, laugh, chew, talk; (ii) facial movements with object interaction - e.g.: smoke, eat, drink; (iii) general body movements - eg.: cartwheel, clap hands, climb, climb stairs, dive, fall on the floor, backhand flip, handstand, jump, pull up, push up, run, sit down, sit up, somersault, stand up, turn, walk, wave; (iv) body movements with object manipulation - e.g.: brush hair, catch, draw sword, dribble, golf, hit something, kick ball, pick, pour, push something, ride bike, ride horse, shoot ball, shoot bow, shoot gun, swing baseball bat, sword exercise, throw; and (v) body movements for human interaction: fencing, hug, kick someone, kiss, punch, shake hands, sword fight. Sample images for each category are illustrated in Figure 4.3.

The protocol used to evaluate the dataset is based on three distinct training and testing splits. According to the authors, the sets were built to ensure that clips from the same video were not used for both training and testing. For each activity category, there are 70 training and 30 testing videos in each split. Table 4.3 summarizes the number of videos for each activity in each split. The performance is evaluated by computing the metric average accuracy across all classes over the three splits.

Table 4.3. Number of videos for each activity from HMDB51 dataset [Kuehne et al., 2011].

Activity	#Total videos	Activity	#Total videos
1: Brush hair	107	27: Pullup	104
2: Cartwheel	107	28: Punch	126
3: Catch	102	29: Pushup	103
4: Chew	109	30: Push	116
5: Clap	130	31: Ride bike	103
6: Climb stairs	112	32: Ride horse	116
7: Climb	108	33: Run	232
8: Dive	127	34: Shake hands	162
9: Draw sword	103	35: Shoot ball	131
10: Dribble	145	36: Shoot bow	112
11: Drink	164	37: Shoot gun	103
12: Eat	108	38: Situp	105
13: Fall floor	136	39: Sit	142
14: Fencing	116	40: Smile	102
15: Flic flac	107	41: Smoke	109
16: Golf	105	42: Somersault	140
17: Handstand	113	43: Stand	154
18: Hit	127	44: Swing baseball	143
19: Hug	118	45: Sword exercise	127
20: Jump	151	46: Sword	127
21: Kick ball	128	47: Talk	120
22: Kick	130	48: Throw	102
23: Kiss	102	49: Turn	240
24: Laugh	128	50: Walk	548
25: Pick	106	51: Wave	104
26: Pour	106		

The HMDB51 dataset displays different levels of difficulty, including camera motion, viewpoint variation, low quality videos and body part occlusion. It is worth noting that to overcome such difficulties, each video of the dataset is annotated with meta-labels describing some properties of the clip, such as which body parts are visible in the video, if the camera of the video is static or with motion, which viewpoint the video was recorded, information regarding the quality of the video (good, medium or bad), and the number of people involved in the activity. However, the majority of works in the literature do not use such information.

4.4 UCF101 Dataset

UCF101 [Soomro et al., 2012] is a realistic activity recognition dataset composed of videos collected from Youtube. It is constituted by 13,320 videos with 101 activity categories. The activities are grouped into 25 groups consisting of 4-7 videos of each activity. Such groups may share some common features, such as similar background or similar viewpoint.

Similarly to HMDB51 dataset, the UCF101 activity categories can be divided into five different types: (i) human-object interaction; (ii) body-motion only; (iii) human-human interactions; (iv) playing different musical instruments; and (v) sport activities. Sample images for each activity category are illustrated in Figure 4.4.

The authors recommend to use the three split evaluation protocol to keep consistency of the reported experiments in the dataset. Such protocol is based on three distinct training and testing splits that were designed to keep separate clips from the same group in training and testing, since clips within a group were obtained from a single long video. Each training split is composed of 18 different groups and the remaining seven groups are used for testing.

Tables summarizing the number of videos for each activity in each split are given in Appendix A (A.1 and A.2). The performance metric used for evaluation is the average recognition accuracy across all classes over the three splits.

The main challenges of this dataset are the large diversity of activities and the presence of large variations in camera motion. Moreover, since the dataset is composed of videos from Youtube, it also presents other challenges brought during the camera recording process such as object viewpoint, appearance, pose and scale, cluttered background, and illumination conditions.

4.5 NTU RGB+D 60 Dataset

NTU RGB+D 60 [Shahroudy et al., 2016] is a publicly available 3D activity recognition dataset. The videos were collected using three Kinect cameras providing four different data information: (i) RGB frames; (ii) depth maps; (iii) infrared sequences; and (iv) skeleton joints. The dataset provides high resolution RGB videos (1920×1080) and depth maps and IR videos in 512×424 resolution. Moreover, the 3D skeletal data are constituted by the three dimensional locations of 25 major body joints per frame.

It consists of 56,880 videos from 60 activity categories, which are performed by 40 distinct subjects, resulting in 948 videos for each activity category. The activity categories are divided into three broad groups: (i) daily activities - e.g.: drink water,

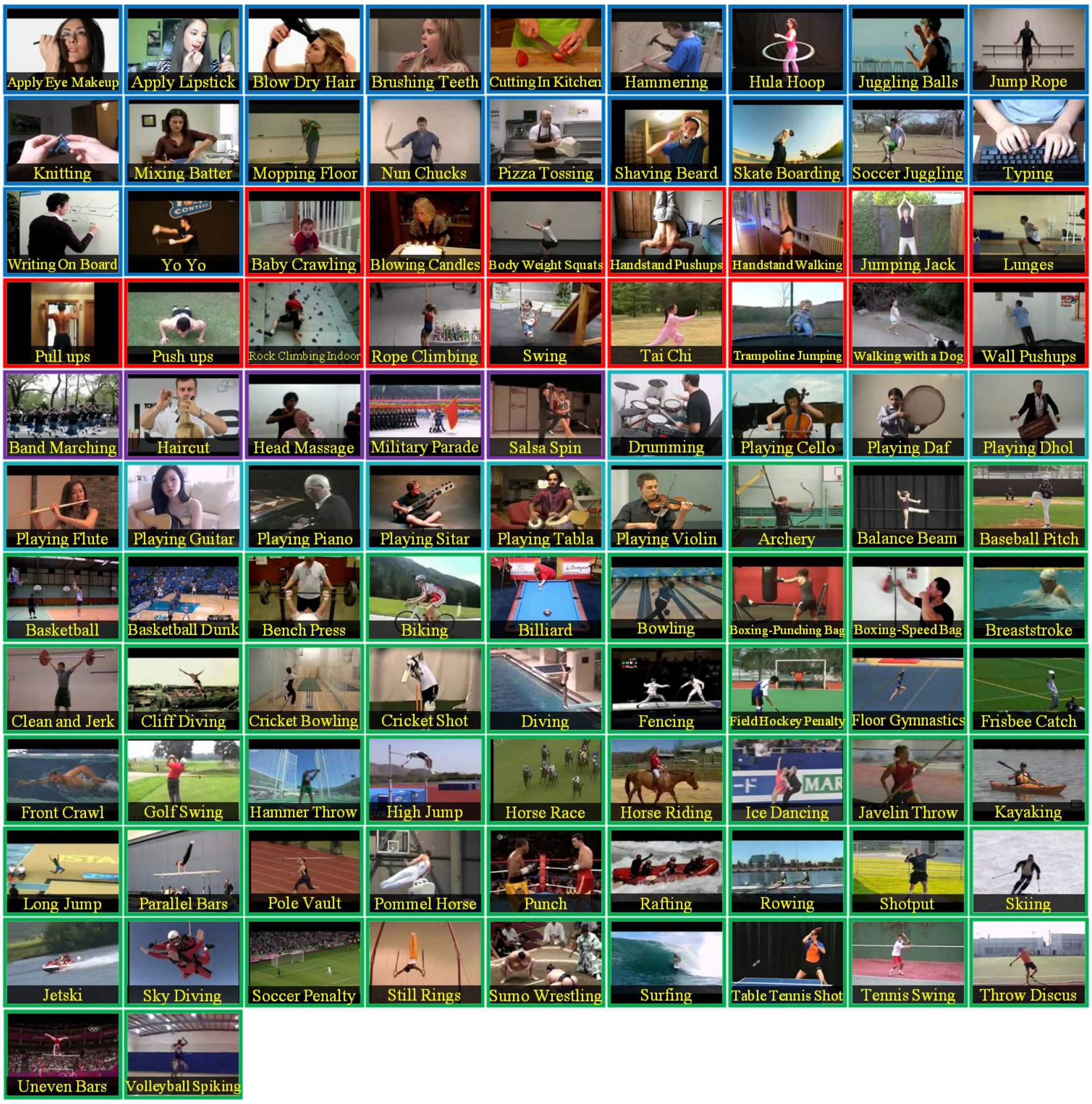
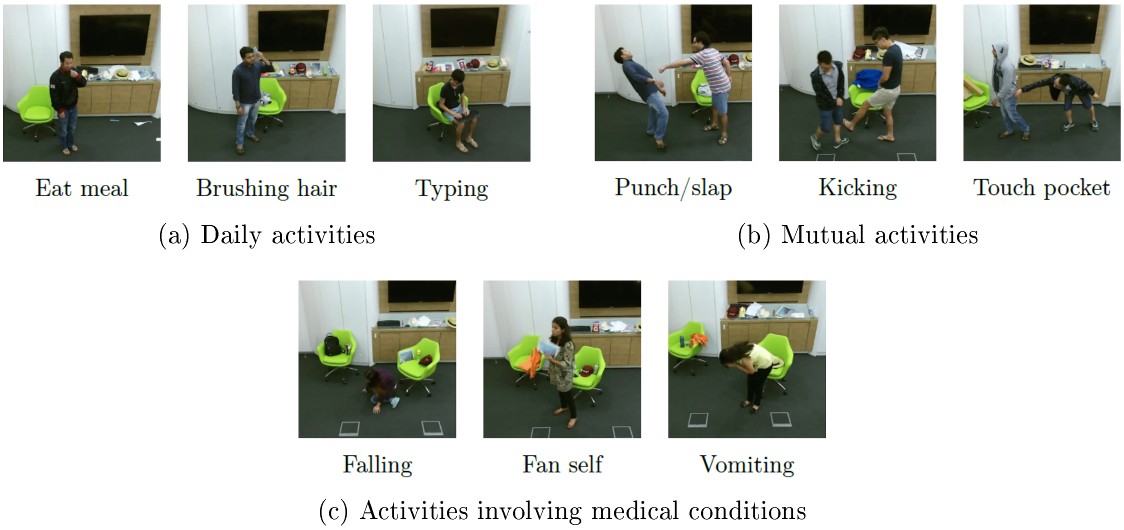


Figure 4.4. Example of activities categories from UCF101 dataset [Soomro et al., 2012]. The activity categories can be divided into five different types: (i) human-object interaction (blue framed); (ii) body-motion only (red framed); (iii) human-human interactions (purple framed); (iv) playing different musical instruments (light blue framed); and (v) sport activities (green framed). Figure extracted from Soomro et al. [2012].

reading, take off a hat/cap, pointing to something, eat meal, writing, cheer up, taking selfie, brushing teeth, tear up paper, hand waving, check time (from watch), brushing hair, wear jacket, kicking something, rub two hands, drop, take of jacket, reach into pocket, nod head/bow, pickup, wear a shoe, hopping, shake head, throw, take off shoe, jump up, wipe face, sitting down, wear on glasses, phone call, salute, standing up,



Eat meal Brushing hair Typing

(a) Daily activities

Punch/slap Kicking Touch pocket

(b) Mutual activities

Falling Fan self Vomiting

(c) Activities involving medical conditions

Figure 4.5. Example of activities from the 3 different groups from NTU RGB+D 60 dataset [Shahroudy et al., 2016].

take off glasses, playing with phone, put the palms together, clapping, put on hat/cap, typing, cross hands in front; (ii) mutual activities - e.g.: punch/slap, pat on the back, giving something, walking towards, kicking, point finger, touch pocket, walking apart, punching, hugging, handshaking; and (iii) activities involving medical conditions - e.g.: sneeze/cough, headache, neck pain, staggering, chest pain, vomiting, falling, back pain, fan self. Sample images for each group are illustrated in Figure 4.5.

To have standard evaluations for all the reported results on the dataset, the authors defined two evaluation protocols. In the cross-subject evaluation, the 40 subjects are split in half into training and test sets, resulting in 276 videos of each activity for training and 672 for test. The cross-view evaluation uses samples from one camera for testing and the other two for training, resulting in 316 videos of each activity for training and 632 for test. A table summarizing the total number of videos for each activity is given in Appendix A (A.3). The performance is evaluated by computing the average recognition across all classes.

According to Shahroudy et al. [2016], the constitution of human activities depends on many characteristics, such as the age, gender, culture and even physical conditions of the subjects. In view of that, the main challenge in the NTU RGB+D 60 dataset is the high variation of human subjects with much more intra-class variations (e.g., poses, environmental conditions, interacted objects and ages of actors).

4.6 NTU RGB+D 120 Dataset

NTU RGB+D 120 [Liu et al., 2019] is a large-scale 3D activity recognition dataset captured under various environmental conditions. This dataset is an extended version of the aforementioned NTU RGB+D 60 [Shahroudy et al., 2016]. The videos were also collected using three Kinect cameras providing 3D skeletal data composed of the three dimensional locations of 25 major body joints per frame.

It consists of 114,480 RGB+D video samples captured using the Microsoft Kinect sensor. As in NTU RGB+D 60, the dataset provides RGB frames, depth maps, infrared sequences and skeleton joints. It is 120 activity categories performed by 106 distinct subjects in a wide range of age distribution.

The NTU RGB+D 120 dataset contains many more activity classes, such as fine-grained hand and finger motions (e.g., make ok sign and snapping fingers), fine-grained object-related individual activities (e.g., counting money and play magic cube), object-related mutual activities (e.g., wield knife towards other person and hit other person with object), similar posture patters but with different motion speeds (e.g., grab other person's stuff), activities with similar body motions but with different objects involved (e.g., put on bag/backpack) and activities with similar objects involved but with different body motions (e.g., put on bag/backpack and take something out of a bag/backpack).

There are two different evaluation protocols: cross-subject, which split the 106 subjects into training and testing; and cross-setup, which divides samples with even setup IDs for training (16 setups) and odd setup IDs for testing (16 setups). The performance is evaluated by computing the average recognition across all classes.

As in NTU RGB+D 60, the main challenge in the NTU RGB+D 120 dataset is the high variation of human subjects with much more intra-class variations (e.g., poses, environmental conditions, interacted objects and ages of actors). Furthermore, fine-grained hand and finger motions and activities with similar objects involved but with different body motions make this dataset more challenging than its previous version.

4.7 Concluding Remarks

The development of datasets is responsible for the advances achieved in the activity recognition research field because they provide standard evaluation data for researchers to compare their proposed approaches.

As reviewed in this chapter, all the datasets used in this dissertation are somehow different from each other and offer different type of challenges. UCF101 may be the more challenging dataset due to its realistic videos extracted from Youtube providing large variations in camera motion, object viewpoint, appearance, pose and scale, cluttered background, and illumination conditions. On the other hand, the KTH dataset is the easiest one with little activity classes in a each controlled scenario. Although the NTU RGB+D 60 and NTU RGB+D 120 datasets were also recorded in controlled scenarios, they provide a high level of difficulty due to the high amount of data and large intra-class variability. To summarize, statistics of all datasets reviewed in this chapter are listed in Table 4.4.

Table 4.4. Summary of all activity recognition datasets used in this dissertation.

Dataset	**#Samples**	**#Classes**	**Protocol**	**Year**
KTH [Schuldt et al., 2004]	600	6	Train/Val/Test	2004
UCF Sports [Rodriguez et al., 2008]	150	10	Train/Val/Test	2008
HMDB51 [Kuehne et al., 2011]	6,766	51	3 Splits (Train/Test)	2011
UCF101 [Soomro et al., 2012]	13,320	101	3 Splits (Train/Test)	2012
NTU RGB+D 60 [Shahroudy et al., 2016]	56,880	60	Cross-subject/view	2016
NTU RGB+D 120 [Liu et al., 2019]	114,480	120	Cross-subject/setup	2019

Chapter 5

Experimental Analysis

In this chapter we present the experimental results obtained with our proposed representations. To that end, we used the six activity recognition datasets introduced in Chapter 4.

The results will be presented in four different groups, each one regarding one proposed representation and its experimental setup. First, we present the empirical results achieved by the proposed spatiotemporal local feature descriptor OFCM (Section 5.1). Then, we evaluate the proposed mid-level representation CCW (Section 5.2). After, in Section 5.3, we present the evaluation of the image domain CNN-based approaches, MOS and MOS+D. Finally, we present the empirical results achieved by the skeleton domain CNN-based approaches, SkeleMotion and TSRJI (Section 5.4). In each group, we perform a comparison with state-of-the-art methods. To make a fair comparison, we carefully follow the standard evaluation protocol of each dataset.

5.1 Spatiotemporal Local Feature Evaluation

In this section we describe the experimental results obtained with the OFCM and compare it to other feature descriptors in the literature. Besides the well-known spatial HOG [Dalal & Triggs, 2005] descriptor, five widely employed spatiotemporal feature descriptors used by state-of-the-art approaches were chosen to be compared with our proposed feature: HOF [Laptev et al., 2008], HOG-HOF [Laptev et al., 2008], HOG3D [Kläser et al., 2008], MBH [Dalal et al., 2006] and GBH [Shi et al., 2015]. We also compare the proposed OFCM with the Dense Trajectories (DT) method [Wang et al., 2011], which employs a combination of three descriptors (HOG [Dalal & Triggs, 2005], HOF [Laptev et al., 2008], and MBH [Dalal et al., 2006]) in conjunction with BoW and the SVM classifier.

To isolate only the contributions brought by the feature descriptors to the activity recognition problem, all descriptors were tested on the same datasets with the same evaluation protocol and using the same classification method.

5.1.1 Experimental Setup

Aiming for a fair comparison, we apply the same evaluation pipeline for every feature descriptor as in Wang et al. [2011, 2009]: a classical visual recognition pipeline involving training and test steps.

In the training phase, we first apply dense sampling extraction of spatiotemporal feature descriptors in video blocks at regular positions in space and time. To that end, there are 3 dimensions to sample from: $n_i \times n_j \times n_t$. In our experiments, the default size of a block is 18×18 pixels with 10 frames and a 50% of overlapping on both spatial and temporal sampling. Then, following the visual recognition strategy, the local features are encoded into a mid-level representation to be used for the classification task. However, a visual codebook must be created before the encoding. Therefore, we randomly sample V training features. This is very fast and, according to Kläser et al. [2008], the results are very close to those obtained using vocabularies built with k-means. We set the number of visual words V to 4000, which, according to Wang et al. [2009], has shown to achieve good results for a wide range of datasets.

For each video sequence, we compute a BoW feature vector. Spatiotemporal features are first quantized into visual words and a video is then represented as the frequency histogram over the visual words. Euclidean distance is applied as the distance metric between the features and the closest vocabulary word. Finally, an one-against-all classification is performed by a non-linear SVM with a Radial Basis Function (RBF)-kernel.

In the test phase, a test video sequence is classified by applying the trained classifier obtained during the training phase. Therefore, for a test video sequence, spatiotemporal feature descriptors are extracted with dense sampling. Then, the BoW feature vector is generated using the visual codebook previously created. Finally, the generated feature vector is given as input to the trained classifier to predict the class label of the test video sequence.

5.1.2 Parameter Setting

This section presents experiments regarding parameter setting for the OFCM focusing on the optimization of the number of bins for optical flow magnitude and orientation,

	Approach	Acc. (%)
Magnitude bins variation	OFCM ($\omega = 8, \sigma = 2$)	95.52
	OFCM ($\omega = 8, \sigma = 4$)	**95.83**
	OFCM ($\omega = 8, \sigma = 8$)	95.68
	OFCM ($\omega = 8, \sigma = 16$)	95.68
Orientation bins variation	OFCM ($\omega = 2, \sigma = 8$)	95.37
	OFCM ($\omega = 4, \sigma = 8$)	95.52
	OFCM ($\omega = 8, \sigma = 8$)	95.68
	OFCM ($\omega = 16, \sigma = 8$)	95.52

Table 5.1. Activity recognition accuracy (%) results of OFCM. ω and σ variation on the KTH dataset [Schuldt et al., 2004].

the offset distance d, used to create the co-occurrence matrix, and the spatial size of the cuboid.

Tables 5.1 and 5.2 show the results of our experiments varying the number of magnitude and orientation bins (σ and ω), on KTH and UCF Sports datasets, respectively. We empirically set the offset distance $d = 1$. On the KTH dataset, the best result (95.83%) was achieved with both $\omega = 8$ and $\sigma = 4$. On the other hand, the best result (92.00%) for UCF Sports dataset was achieved with $\omega = 4$ and $\sigma = 8$. On the UCF Sports dataset, the best result was obtained with a higher number of magnitude bins (σ) than on the KTH dataset. We believe this is partly because UCF Sports videos are composed of a lower frame rate (10 Frames Per Second (FPS)) presenting higher magnitude values.

We also present experiments varying the offset distance d. For this purpose, we fixed σ and ω with the best parameters obtained on the previous experiments (Tables 5.1 and 5.2). In our test, we varied the offset to a maximum of $d = 3$. According to the results shown in Table 5.3., the best result was achieved with d=1 for both datasets.

Table 5.4 reports the performance of different spatial size of the cuboid. As in Wang et al. [2009], we fixed the temporal length n_t to 10 frames and an overlapping rate of 50%. On the KTH dataset, the best result (96.30%) was achieved at a spatial size of 36×36 pixels and the best result for the UCF Sports dataset was achieved with a spatial size of 18×18 pixels (92.00%). We believe this is partly because KTH videos are composed of well-controlled activities in well-controlled scenarios, thus the spatial size of the cuboid does not need to be so fine-grained (detailed) to cover particular patterns of movements. Differently, UCF Sports videos are collected from realistic videos from the web.

	Approach	**Acc. (%)**
Magnitude bins variation	OFCM ($\omega = 8, \sigma = 2$)	89.47
	OFCM ($\omega = 8, \sigma = 4$)	87.33
	OFCM ($\omega = 8, \sigma = 8$)	91.07
	OFCM ($\omega = 8, \sigma = 16$)	91.07
Orientation bins variation	OFCM ($\omega = 2, \sigma = 8$)	90.13
	OFCM ($\omega = 4, \sigma = 8$)	**92.00**
	OFCM ($\omega = 8, \sigma = 8$)	91.07
	OFCM ($\omega = 16, \sigma = 8$)	90.27

Table 5.2. Activity recognition accuracy (%) results of OFCM. ω and σ variation on the UCF Sports dataset [Rodriguez et al., 2008].

		KTH	**UCF Sports**
	Approach	**Acc. (%)**	**Acc. (%)**
Distance *d* variation	OFCM ($d = 1$)	**95.83**	**92.00**
	OFCM ($d = 2$)	94.29	91.87
	OFCM ($d = 3$)	94.44	91.60

Table 5.3. Activity recognition accuracy (%) results of OFCM. Distance d variation on KTH [Schuldt et al., 2004] and the UCF Sports [Rodriguez et al., 2008] datasets .

		KTH	**UCF Sports**
	Approach	**Acc. (%)**	**Acc. (%)**
Cuboid spatial size variation	OFCM ($n_i = 18, n_j = 18, n_t = 10$)	95.83	**92.00**
	OFCM ($n_i = 24, n_j = 24, n_t = 10$)	95.99	87.73
	OFCM ($n_i = 36, n_j = 36, n_t = 10$)	**96.30**	90.00
	OFCM ($n_i = 48, n_j = 48, n_t = 10$)	95.83	89.73
	OFCM ($n_i = 72, n_j = 72, n_t = 10$)	95.52	90.27

Table 5.4. Activity recognition accuracy (%) results of OFCM. Cuboid spatial size variation on KTH [Schuldt et al., 2004] and the UCF Sports [Rodriguez et al., 2008] datasets.

5.1.3 Results and Comparisons

Now, we compare our OFCM approach with several classic local spatiotemporal features of the literature. According to Table 5.5, a considerable improvement was obtained with OFCM, reaching 96.30% of accuracy on the KTH dataset and 92.80% on UCF Sports. There is an improvement of 2.10 percentage points (p.p.) on the KTH dataset and 3.80 p.p. on the UCF Sports dataset achieved by OFCM when compared to Wang et al.'s DT method [Wang et al., 2011]. Furthermore, it is worth noting that

	Approach	KTH Acc. (%)	UCF Sports Acc. (%)	HMDB51 Acc. (%)
Published results	HOG [Dalal & Triggs, 2005]	79.00	77.40	28.40
	HOF [Laptev et al., 2008]	88.00	84.00	35.50
	HOG-HOF [Laptev et al., 2008]	86.10	81.60	43.60
	HOG3D [Kläser et al., 2008]	85.30	85.60	36.20
	MBH [Dalal et al., 2006]	89.04	90.53	51.50
	GBH [Shi et al., 2015]	92.70	-	38.80
	DT [Wang et al., 2011]	94.20	88.20	46.60
Our results	OFCM	**96.30**	**92.80**	**56.91**

Table 5.5. Activity recognition accuracy (%) results of OFCM and classic spatiotemporal features of the literature on KTH Schuldt et al. [2004] and the UCF Sports Rodriguez et al. [2008] datasets. Results for HOG, HOF, HOG-HOF and HOG3D were obtained from Wang et al. [2009].

their approach uses a combination of three different feature descriptors (HOG, HOF and MBH) while we only employed our proposed OFCM feature. Therefore, such results can be considered remarkably good and confirm the advantages introduced by our spatiotemporal feature descriptor.

For the experiments on the HMDB51 dataset, we used the parameters learned using the KTH dataset, as they turned out to be universal enough to obtain accurate results [Kläser et al., 2008]. For this dataset OFCM also achieves the best results, reaching 56.91% of accuracy, an improvement of 5.41 p.p. when compared to the MBH feature descriptor [Dalal et al., 2006]. These results demonstrate the generalization ability of OFCM once its parameters were estimated using data from the KTH dataset.

5.2 Mid-level Representation Evaluation

In this section, we describe the experimental results obtained with CCW for the activity recognition problem and compare it to other mid-level representations in the literature. We evaluate the proposed representation and compare the approach to a BoW based activity recognition approach and to methods in the literature that also employ co-occurrence to encode information, such as Banerjee & Nevatia [2011], Zhang et al. [2012] and Sun & Liu [2013].

5.2.1 Experimental Setup

In the interest of a fair comparison, we apply the same evaluation pipeline as Wang et al. [2009]. It is a classical visual recognition pipeline that comprises two phases: training and testing.

In the training phase, we first densely extract OFCM spatiotemporal feature descriptors[1]. Dense sampling extracts video blocks at regular positions in space and time. There are three dimensions to sample from: $n_i \times n_j \times n_t$. In our experiments, the minimum size of a block is 18 × 18 pixels and 10 frames. Spatial and temporal sampling are performed with 50% of overlapping. Next, following the visual recognition strategy, the local features must be encoded into a mid-level representation to be used for the classification task. However, a visual codebook must be created before the encoding. Thus, we randomly sample K training features. This is very fast and, according to Kläser et al. [2008], the results are very close to those obtained using vocabularies built with k-means. After, for each video sequence, we extract a Co-occurrence of Codewords (CCW) representation. Spatiotemporal features are first quantized into the codewords according to the codebook previously created and a video is then represented as four co-occurrence matrices ($\pi = 0°$, 45°, 90° and 135°). Here, we extract $f = 12$ Haralick textural features: angular second moment, contrast, correlation, sum of squares, inverse difference moment, sum average, sum variance, sum entropy, entropy, difference variance, difference entropy and maximal correlation coefficient. Then, the feature vectors are assigned to their closest codeword using Euclidean distance. Finally, one-against-all classification is performed by a non-linear SVM with a RBF-kernel.

In the testing phase, a new test video sequence is classified by using the classifier learned during the training phase. Thus, for a test video sequence, OFCM spatiotemporal feature descriptors are extracted with dense sampling. Next, the CCW representation is generated using the visual codebook previously created. Then, that feature vector is presented as input to the trained classifier to predict the class label of the test video sequence.

It is important to emphasize that, in the experiments, we change only the mid-level representation used in the pipeline since our main goal is to compare the real contribution of our proposed feature representation, i.e., for every experiment, the pipeline is the same, only the feature representation is changed.

[1] Although we used OFCM, it is important to emphasize that any feature descriptors can be used with CCW.

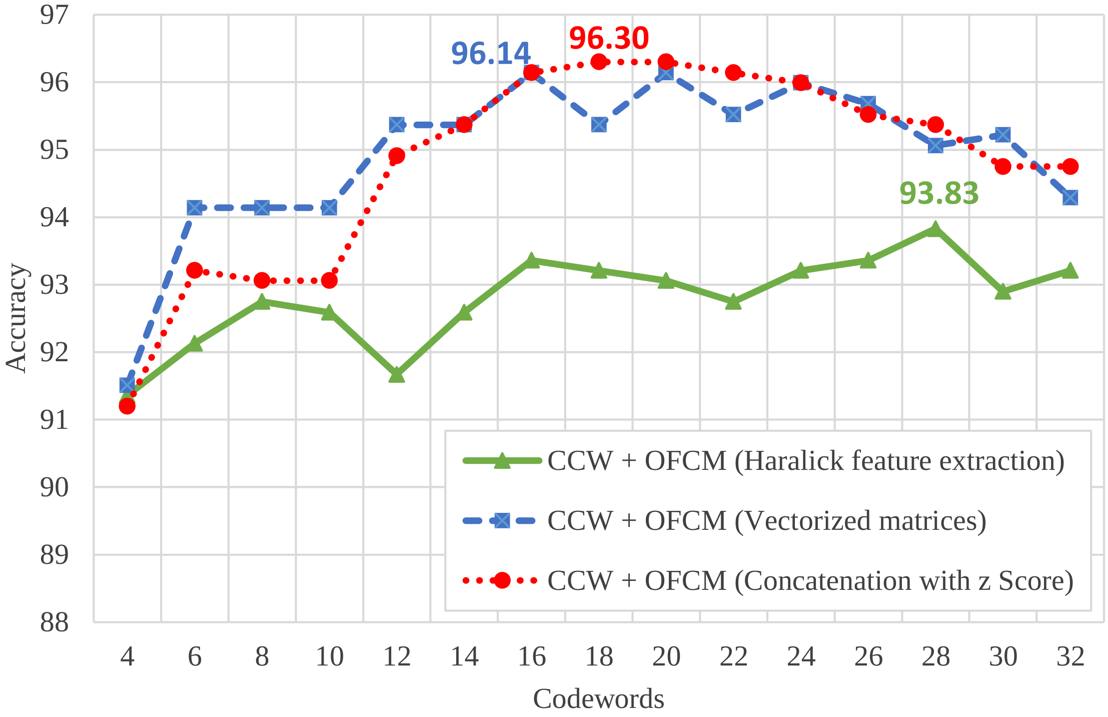


Figure 5.1. Accuracy of codebook size variation on the KTH dataset Schuldt et al. [2004].

5.2.2 Parameter Setting

In this section, we present experiments for parameter optimization and report a comparison of our proposed feature representation. To that end, we used the KTH dataset to perform parameter setting. We focused on the optimization of the number of codewords K used to quantize the features and also to create the co-occurrence matrices. We used the optimized parameters of the OFCM feature descriptor for KTH presented in Section 5.1.

Figure 5.1 shows the variation of the K parameter (number of codewords) for three variations of our proposed feature representation. For the first one, CCW + OFCM (Haralick feature extraction), $K = 28$ presents the best accuracy value reaching 93.83% with final feature vector length of 48 (four co-occurrence matrices with 12 Haralick features each). Second, CCW + OFCM (vectorized matrices), reaches an accuracy of 96.14% with $K = 16$ and a final feature vector length of 1024 (four co-occurrence matrices with size 16×16 each). We also consider a third variation of the CCW representation which, is a concatenation of the two aforementioned variations. Here, we simply concatenated the last two presented approaches (Haralick features and vectorized matrices). For that purpose, we empirically tested four different combination

Approach	**Raw concatenation Acc. (%)**	**L_1 norm Acc. (%)**	**L_2 norm Acc. (%)**	**z score norm Acc. (%)**
Vectorized CCW + OFCM ($k = 4$)	72.22	72.22	72.22	91.20
Vectorized CCW + OFCM ($k = 8$)	88.58	78.55	78.55	93.06
Vectorized CCW + OFCM ($k = 18$)	88.43	81.02	81.02	**96.30**
Vectorized CCW + OFCM ($k = 32$)	87.50	78.55	78.55	94.75

Table 5.6. Video classification Acc. (%) results of the concatenation variations of the CCW representation on the KTH dataset Schuldt et al. [2004].

	Approach	**Codewords k**	**Feature length**
Published results	BoW + OFCM	4000	4000
Our results	CCW + OFCM (Haralick features)	**28**	**48**
	CCW + OFCM (vectorized matrices)	16	1024
	CCW + OFCM (concatenation)	18	1344

Table 5.7. A summary of the proposed feature vector lengths of the CCW representation and OFCM + BoW model fine tuned on the KTH dataset Schuldt et al. [2004].

strategies: a raw concatenation; concatenation followed by L_1 norm; concatenation followed by L_2 norm; and concatenation followed by z score norm. For z score norm, we learn the means μ_i and standard deviations σ_i of each feature dimension (x_i) during the training phase.

Table 5.6 shows the results of the four concatenation strategies with the best result of 96.30% being achieved by the setup CCW representation followed by z score norm with $K = 18$. Figure 5.1 also illustrates the codebook size variation for CCW + OFCM (concatenation with z score).

Table 5.7 summarizes the feature vector lengths for the proposed CCW and for the BoW representation length used in the literature. We can see that CCW presents a more compact representation using fewer codewords and a smaller final feature vector length. For instance, CCW + OFCM (Combined z score) achieves the same accuracy as the BoW + OFCM using a more compact representation with only $k = 18$ codewords and a final feature vector length of 1344, instead of $k = 4000$ used by the OFCM + BoW.

[1] The results presented in Shi et al. [2015] are using Fisher Vector (FV). However, since we directly compare with BoW, here we apply the same mid-level representation (BoW).

5.2.3 Results and Comparisons

In this section, we compare our approach with several classic local spatiotemporal features + BoW model of the literature. Thus, we used the KTH tuned parameters from Section 5.2.2 on UCF Sports and the HMDB51 dataset experiments.

According to Table 5.8, CCW + OFCM (Combined) achieved the highest accuracy value, which is the same as the BoW + OFCM, reaching 96.30% on the KTH dataset. Moreover, we note an improvement of 2.10 p.p. on the KTH dataset, achieved by our representation when compared to the DT method [Wang et al., 2011]. Furthermore, it is important to point out that their approach uses a combination of three different feature descriptors (HOG, HOF and MBH), while we only employ the OFCM feature. Although our CCW representation did not achieve better results on UCF Sports and HMDB51 datasets, it still presents comparable results being only 1.47 p.p. behind the best result on UCF Sports and 1.66 p.p. on HMDB51. Furthermore, as already mentioned, it is worth emphasizing that the CCW has a more compact representation (smaller feature vector length), as shown on Table 5.7.

A comparison with methods of the literature that also employed co-occurrence to encode information on activity recognition tasks is shown on the second part of Table 5.8. On the KTH dataset, our CCW representation (Vectorized and Combined) achieves an improvement of 2.32 p.p. when compared to the Banerjee & Nevatia [2011] Conditional Random Field (CRF) method. Moreover, on UCF Sports, we outperform Zhang et al. [2012] Spatio Temporal Normalized Google-Like Distance Correlogram (ST-NGLDC) method by 2.93 p.p. using our CCW (vectorized) representation. Finally, our three CCW representations presented a large accuracy gain when compared to the Co-occurrence Space (CS) method [Zhang et al., 2012].

5.3 Image Domain Convolutional Neural Network based Approach Evaluation

This section describes the experimental results obtained with the MOS and MOS+D approaches for the activity recognition problem. We first introduce the implementation details and then we compare proposed approaches to other CNN methods in the literature. We used VD2S [Wang et al., 2015] and TSN [Wang et al., 2016a] as baseline comparisons. To isolate only the contributions brought by our method to the activity recognition problem, the baseline was tested on the same datasets with the same split of training and testing data. Since CNN approaches require too many data samples

	Approach	KTH **Acc. (%)**	UCF Sports **Acc. (%)**	HMDB51 **Acc. (%)**
Published BoW results	BoW + HOG	79.00	77.40	28.40
	BoW + HOF	88.00	84.00	35.50
	BoW + HOG-HOF	86.10	81.60	43.60
	BoW + HOG3D	85.30	85.60	36.20
	BoW + MBH	89.04	90.53	51.50
	BoW + GBH [1]	92.70	-	38.80
	BoW + OFCM	**96.30**	**92.80**	**56.91**
	DT [Wang et al., 2011]	94.20	88.20	46.60
Published co-occurrence results	CRF [Banerjee & Nevatia, 2011]	93.98	-	-
	CS [Zhang et al., 2012]	91.20	-	26.82
	ST-NGLDC [Sun & Liu, 2013]	91.82	89.74	-
Our results	CCW + OFCM (Haralick features)	93.83	90.53	52.62
	CCW + OFCM (vectorized matrices)	96.14	**91.33**	55.08
	CCW + OFCM (concatenation)	**96.30**	91.07	**55.25**

Table 5.8. Activity recognition accuracy (%) results of OFCM + CCW representation and classic spatiotemporal features of the literature + BoW on KTH [Schuldt et al., 2004] and UCF Sports [Rodriguez et al., 2008] datasets. Results for HOG, HOF, HOG-HOF and HOG3D were obtained from [Wang et al., 2009].

for training, the evaluations are performed considering two well-known datasets for the activity recognition problem, UCF101 [Soomro et al., 2012] and NTU RGB+D 60 [Shahroudy et al., 2016], in which we employ the evaluation protocols and metrics proposed by their authors.

5.3.1 Implementation Details

5.3.1.1 Pre-training

As stated by Wang et al. [2015], the UCF101 dataset training split is too small to train a deep convolutional network. Hence, a possible solution used by several works [Simonyan & Zisserman, 2014; Wang et al., 2015; Feichtenhofer et al., 2016; Wang et al., 2016a] is to use models learned using the ImageNet dataset as the initialization for network training. In this way, we also employed the ImageNet model as pre-training.

5.3.1.2 Training

Following the implementation details used by our baselines [Wang et al., 2015, 2016a], we set the learning rate initially to 0.005. For the Very Deep Two-Stream network (VD2S) Wang et al. [2015], the learning rate decreases at every $5,000$ iterations dividing it by 10. The maximum number of iterations was set to $15,000$. We followed a similar scheme for the Temporal Segment Networks (TSN) Wang et al. [2016a] reducing the learning rate after $12,000$ and $18,000$ iterations. For the TSN, the number of iterations was set to $20,000$. We kept the same schedule for all training sets.

Similarly to Simonyan & Zisserman [2014]; Wang et al. [2015, 2016a], the network weights are learned using the mini-batch stochastic gradient descent with a momentum set to 0.9 and weight decay of 0.0005. We also set high dropout ratio for the fully connected layers (0.9 and 0.8).

Krizhevsky et al. [2012] demonstrated that data augmentation techniques can be very effective in avoiding overfitting. Thus, we cropped and flipped four corners and the center of the frame. In addition, we applied a multi-scale cropping method and randomly sampled the cropping width and height from $\{256, 224, 192, 168\}$ (finally, we resize the cropped regions to 224×224). It is important to state that our baseline [Wang et al., 2015] employed the same data augmentation procedure.

5.3.1.3 Test

To perform a fair comparison, we applied the same test scheme used by our baseline [Wang et al., 2015], described as follows. First, we sample 25 magnitude/orientation flow images for testing. Then, from each of these, we obtain 10 convolutional network inputs (by cropping and flipping four corners and the center). Finally, the prediction score for the input video is obtained by averaging the sampled images' scores and their crops. The same testing scheme was used by the original two-stream convolutional network [Simonyan & Zisserman, 2014]. For the fusion of MOS and other streams, we use a non-weighted linear fusion that consists in a combination of their prediction scores.

5.3.1.4 Optical Flow Extraction

As mentioned in Chapter 3, the magnitude/orientation images are computed from the optical flow information. To that end, we extract the optical flow information using the TVL1 algorithm [Zach et al., 2007], implemented in OpenCV with CUDA. For the sake of comparison, our baseline [Wang et al., 2015] used the same optical flow algorithm. To obtain the magnitude and orientation image information we empirically set the parameters $h = 15$ and $l = -15$ to compute M; and $h = 180$, $l = -180$ and $m = 128$ to compute θ'.

5.3.2 Depth Information Estimation

To extract the depth information used as weighting scheme for the magnitude information (MOS+D), we used the method provided by Godard et al. [2017] with default parameters and the pre-trained model on the Cityscapes dataset [Cordts et al., 2016]. The implementation and model were made available by the authors[2]. To obtain the weighted magnitude by depth images information we empirically set the parameters $d = 215$.

5.3.3 Results and Comparisons

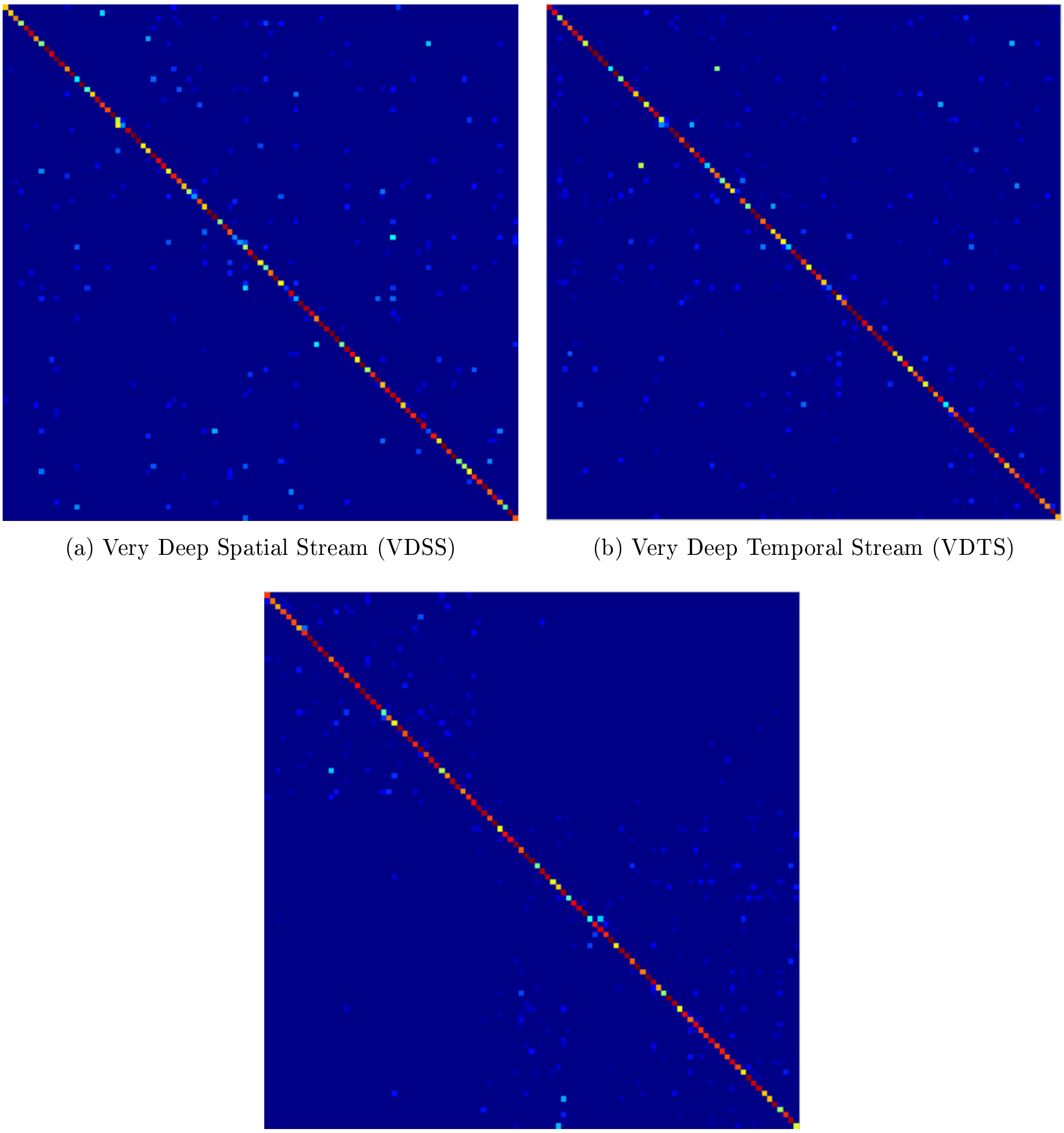
We report the activity recognition performance of our MOS with VGG-16 architecture and the VD2S baseline [Wang et al., 2015] on the UCF101 dataset in Table 5.3.3. It shows a comparison of our method to the three different streams of our VD2S baseline [Wang et al., 2015]: (i) Very Deep Spatial Stream (VDSS); (ii) Very Deep Temporal Stream (VDTS); and VD2S. According to the results, a considerable improvement was obtained with MOS when compared to the baseline single streams, reaching 90.8% of accuracy on split 1 of the UCF101 dataset. There is an improvement of 5.1 p.p. when compared to VDTS [Wang et al., 2015] and 11.0 p.p. when compared to VDSS [Wang et al., 2015]. This shows that the optical flow preprocessing (i.e., extraction of magnitude and orientation information) brings improvement over using raw optical flow information. Furthermore, it is worth noting that our best result using MOS on split 1 is close to the best one reported (VD2S), which is obtained by using a combination of two different streams (spatial and temporal informations), while we only used our single MOS (temporal information). The same observations can be considered when analyzing the results of our temporal stream on splits 2 and 3. Therefore, such results can be considered remarkably good and confirm that pre-processing the inputs helps on guiding the network to extract certain information and, although temporal evolution patterns can be learned implicitly with CNNs, an explicit modeling is preferable and is able to achieve better results.

Figure 5.2 shows the confusion matrices of VDSS, VDTS and our MOS for the UCF101 split 1 (we highlighted the false positives and false negatives to make it more visible on where each method fails). We can observe that our approach fails on classes that are more semantically closer to each other[3], whereas VDSS and VDTS fail in

[2] `https://github.com/mrharicot/monodepth`

[3] Since the activities on the confusion matrices are sorted according to its labels (e.g., ApplyEyeMakeup, ApplyLipstick, or BaseballPitch, Basketball, BasketballDunk), near regions in the confusion matrix denote semantically closer activities.

a random manner. In addition, the three methods produce false positives and false negatives different from each other, indicating the possibility of fusion.



(a) Very Deep Spatial Stream (VDSS)

(b) Very Deep Temporal Stream (VDTS)

(c) Magnitude-Orientation Stream (MOS)

Figure 5.2. Confusion matrices on UCF101 split 1. False positives and false negatives were highlighted to show where each method fails.

To exploit a possible complementarity of the three approaches (VDSS, VDTS and our MOS), we combined the different streams by employing a late fusion technique using a weighted linear combination of their prediction scores. According to the results showed in Table 5.3.3, any type of combination performed with our MOS provides better results than VD2S [Wang et al., 2015], with the best result achieving an improvement of 2.4 p.p. over VD2S [Wang et al., 2015].

To verify the statistical significance of these combination results, a statistical test for the differences between the means was performed using a Student t-test [Jain, 1991], paired over the dataset splits. The test consists in determining a confidence interval for the differences and simply checking if the interval includes zero (i.e., if the confidence interval does not include zero, the dfference is significant at that confidence level). Thus, at 95% confidence level, we can conclude that the difference is significant for our combination results.

		Split 1	**Split 2**	**Split 3**	**Average**
	Approach	**Acc. (%)**	**Acc. (%)**	**Acc. (%)**	**Acc. (%)**
Baseline	VDSS [Wang et al., 2015]	79.8	77.3	77.8	78.4 ± 1.1
	VDTS [Wang et al., 2015]	85.7	88.2	87.4	87.0 ± 1.0
	VD2S [Wang et al., 2015]	90.9	91.6	91.6	**91.4** ± 0.3
Our results	MOS	90.8	89.3	91.5	90.5 ± 0.9
	MOS + VDSS [Wang et al., 2015]	93.1	91.9	92.6	**92.5** ± 0.5
	MOS + VDTS [Wang et al., 2015]	91.4	92.2	93.6	**92.4** ± 0.9
	MOS + VD2S [Wang et al., 2015]	93.7	93.1	94.8	**93.8** ± 0.7

Table 5.9. Activity recognition results (accuracy % and standard deviation) of MOS with VGG-16 architecture and VD2S [Wang et al., 2015] baseline on UCF101 [Soomro et al., 2012] activity dataset. Results for the baseline were obtained running the code provided by Wang et al. [2015]. Note that our results were achieved with only our single Magnitude-Orientation Stream (temporal information) while the results of Wang et al. [2015] consider two streams (spatial and temporal information).

We also report the activity recognition performance of our MOS with Inception architecture in comparison with the TSN [Wang et al., 2016a] baseline. Table 5.3.3 provides a comparison of our approach to three different streams on the UCF101 dataset: (i) Temporal Segment Stream (TSS), (ii) Spatial Segment Stream (SSS), and (iii) TSN. According to the results, a considerable improvement was achieved with MOS when compared to the TSN [Wang et al., 2016a] baseline single streams, reaching 92.4% of accuracy on UCF101. We can note an improvement of 7.3 p.p. when compared to SSS [Wang et al., 2016a] and 2.7 p.p. when compared to TSS [Wang et al., 2016a]. Once more, such results confirm that pre-processing the optical flow inputs helps guiding the network to extract a better information.

We also exploited a possible complementarity of the spatial and temporal streams from TSN and our MOS approach. Here, we applied the same late fusion technique used on VGG-16 architecture experiments, which consists of a weighted linear combination of the prediction scores. Last line of Table 5.3.3 shows the combination results, with the best result improving 2.7 p.p when compared to TSN [Wang et al., 2015]. We also verified the statistical significance of these combination results using Student t-test [Jain, 1991], paired over the dataset splits. We can conclude that, at 95% confidence level, the difference is significant for our combination results.

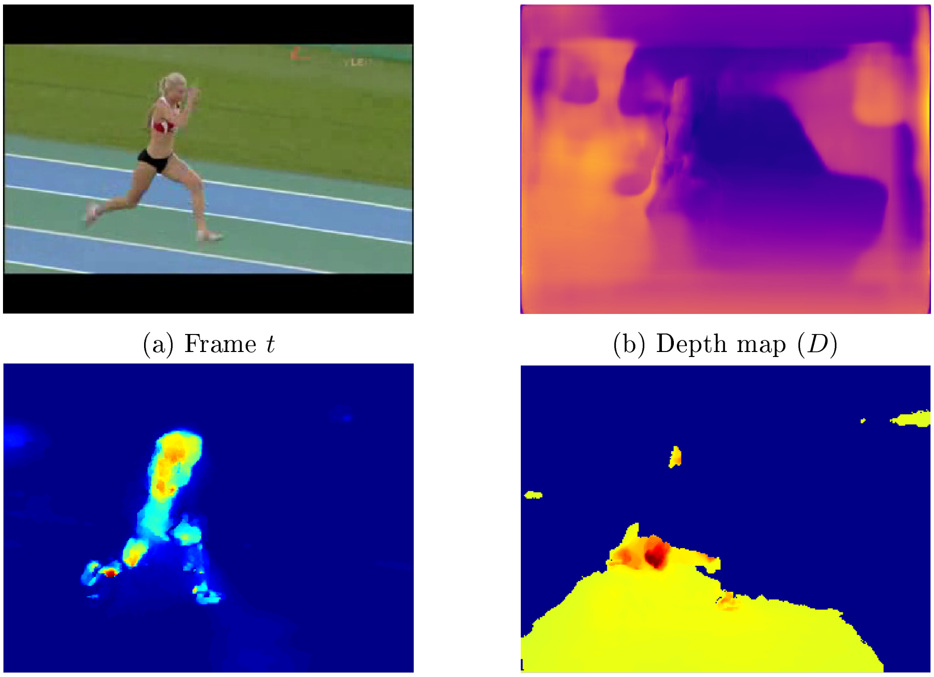
	Approach	Split 1 Acc. (%)	Split 2 Acc. (%)	Split 3 Acc. (%)	Average Acc. (%)
Baseline	SSS Wang et al. [2016a]	85.5	84.9	84.5	85.1 ± 0.4
	TSS Wang et al. [2016a]	87.6	90.2	91.3	89.7 ± 1.6
	TSN Wang et al. [2016a]	93.5	94.3	94.5	**94.0** ± 0.4
Our results	MOS	91.5	93.0	92.9	92.4 ± 0.7
	MOS + SSS Wang et al. [2016a]	96.2	96.7	96.1	**96.3** ± 0.3
	MOS + TSS Wang et al. [2016a]	93.4	94.7	94.8	**94.3** ± 0.6
	MOS + TSN Wang et al. [2016a]	96.5	97.0	96.8	**96.7** ± 0.2

Table 5.10. Activity recognition results (accuracy % and standard deviation) of MOS with Inception architecture and TSN [Wang et al., 2016a] baseline on UCF101 [Soomro et al., 2012] activity dataset. Results for the baseline were obtained running the code provided by Wang et al. [2016a]. Note that our results were achieved with only our single MOS (temporal information) while the results of Wang et al. [2016a] consider two streams (spatial and temporal information).

Table 5.3.3 shows the results achieved with our proposed magnitude weighted by depth scheme MOS+D on the UCF101 dataset. We note a difference of 3.0 p.p. smaller than our main method. Such worse results are due to poorly estimated depth maps. Although the magnitude weighted depth scheme circumvents problems related to activities taken regardless of their distance to the camera, as shown in Figure 3.9, the magnitude information may be damaged if the depth maps are not well estimated, as can be seen in Figure 5.3.

Approach	Split 1 Acc. (%)	Split 2 Acc. (%)	Split 3 Acc. (%)	Average Acc. (%)
MOS	91.5	93.0	92.9	92.4
MOS+D	88.6	89.8	89.9	89.4

Table 5.11. Activity recognition accuracy (%) results of Magnitude-Orientation Stream with Inception architecture with and without depth weighting on the UCF101 [Soomro et al., 2012] dataset.



(a) Frame t

(b) Depth map (D)

(c) Magnitude (M)

(d) Harmed Magnitude weighted by depth ($M^{'}$)

Figure 5.3. Comparison between magnitude information affected when weighted by poorly estimated depth maps.

Table 5.3.3[4] presents results for many works on the UCF101 dataset. The first part of the table shows results of methods that extract temporal information using handcrafted features. We compare our MOS approach with the results of local feature-based methods, such as BoW + features, FV + features, and IDT. The best result by such type of methods was achieved with IDT + higher FV [Peng et al., 2016], reaching 87.9%. Our best result using the proposed approach combined with VD2S outperforms that by 5.9 p.p..

The second part of Table 5.3.3 shows the results achieved with Neural Network (NN) approaches. According to the results, by only using our MOS, we outperform many methods [Karpathy et al., 2014; Srivastava et al., 2015; Tran et al., 2015; Sun et al., 2015; Simonyan & Zisserman, 2014; Wang et al., 2016a]. It is worth mentioning that we also improved the results achieved by the original two-stream from Simonyan & Zisserman [2014]. Using the VGG-16 architecture, we outperform it by 2.5 p.p. (temporal stream) and by 5.8 p.p. (combining it with VD2S). Further more, using the Inception architecture, we outperform it by 4.4 p.p. (temporal stream) and by 8.7 p.p. (combining it with TSN). Finally, we can observe that our best result did not

[4] Results for features + BoW were obtained from Shi et al. [2015] and features + FV were obtained from Shi [2014].

outperform only Carreira & Zisserman [2017] I3D method. However, it is important to emphasize that they used a huge dataset for pre-training. Nevertheless, we believe our results are remarkably good since 3D convolutional operations are more computationally expensive than the 2D convolutional operations used in our approach. For instance, the Two-Stream I3D network used by Carreira & Zisserman [2017] has 25 million parameters, while the 2D Two-Stream employed by us has less than half (12 million parameters).

	Approach	**UCF101 Acc. (%)**
Handcrafted Methods	HOF + BoW [Laptev et al., 2008]	61.8
	HOG-HOF + BoW [Laptev et al., 2008]	71.8
	MBH + BoW [Dalal et al., 2006]	77.1
	GBH + BoW [Shi et al., 2015]	68.5
	HOG3D + BoW [Kläser et al., 2008]	61.4
	HOF + FV [Laptev et al., 2008]	65.9
	HOG-HOF + FV [Laptev et al., 2008]	75.4
	MBH + FV [Dalal et al., 2006]	81.0
	GBH + FV [Shi et al., 2015]	74.2
	HOG3D + FV [Kläser et al., 2008]	64.7
	IDT [Wang & Schmid, 2013]	85.9
	IDT + higher FV [Peng et al., 2016]	87.9
	IDT + MVSV [Cai et al., 2014]	83.5
NN Methods	Deep Networks [Karpathy et al., 2014]	65.4
	Composite LSTM [Srivastava et al., 2015]	75.8
	C3D Tran et al. [2015]	85.2
	Factorized CNN [Sun et al., 2015]	88.1
	Two-Stream [Simonyan & Zisserman, 2014]	88.0
	Two-Stream F [Feichtenhofer et al., 2016]	92.5
	KVMF [Zhu et al., 2016]	93.1
	TSN (3 modalities) [Wang et al., 2016a]	94.2
	R-STAN-101 (RGB+FLOW) [Liu et al., 2019b]	94.5
	STM ResNet-50 [Jiang et al., 2019]	96.2
	Two-Stream I3D [Carreira & Zisserman, 2017]	**98.0**
Our results	MOS (VGG-16)	90.5
	MOS (VGG-16) + VD2S	93.8
	MOS (Inception)	92.4
	MOS+D (Inception)	89.4
	MOS (Inception) + TSN	**96.7**

Table 5.12. Activity recognition accuracy comparison on the UCF101 dataset [Soomro et al., 2012].

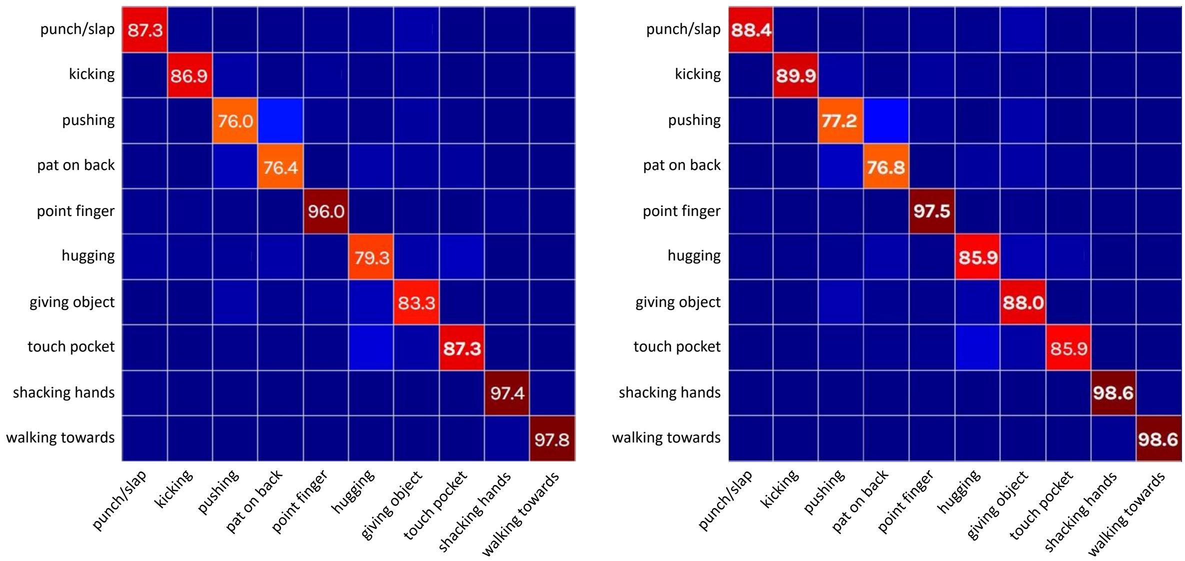
To show that the poor results achieved on the UCF101 dataset with our magnitude weighted by depth scheme were caused by poorly estimated depth maps, we employed the NTU RGB+D 60 dataset which provides accurate depth maps captured by a depth sensor (Kinect). Table 5.3.3 shows the results achieved on the NTU RGB+D 60 dataset with and without our weighting scheme by using the cross-subject evaluation protocol. We can note that both methods achieved very similar results, however, when a non-weighted linear fusion is applied on the methods we achieve an improvement of 2.3 p.p. when compared to MOS. This shows that, although the methods achieved very similar results, the network is learning different information from each data. Figure 5.4 illustrates a part of the confusion matrices focused on activities that involve interactions of two people showing that for all these activities, the non-weighted linear fusion improved the results. Such improvement can be attributed to the depth information, since in such activities people are in different depth planes (see Figure 5.5). Again, we emphasize here that our intention in using the NTU RGB+D 60 dataset with this approach was only to validate that the use of accurate depth information in our approach leads to better learning for motion in different depth planes.

Approach	**Cross-subject Acc. (%)**
MOS	73.1
MOS+D	72.6
Non-weighted Linear Fusion	**75.4**

Table 5.13. Activity recognition accuracy (%) results of MOS with Inception architecture with and without depth weighting scheme on the NTU RGB+D 60 [Shahroudy et al., 2016] dataset.

5.3.4 Discussion

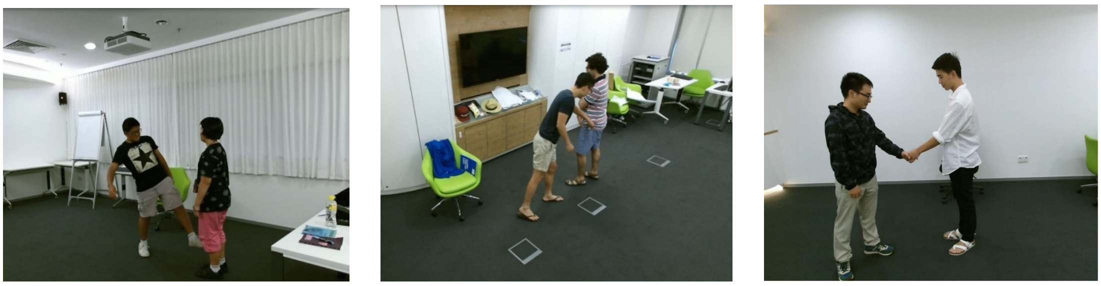
To better analyze our proposed approach, we take a closer look at the activities from UCF101 that our method achieved higher performance than the baseline approaches. For instance, some activities the were most correctly classified by MOS and misclassified by the baselines include *apply lipstick, front crawl, basketball, shaving beard* and *rafting*, among others. We note that the baselines usually confused the activities *apply lip stick* and *shaving beard* with *haircut*, *brushing teeth* or *apply eye makeup* which are activities with movements on very similar areas. Moreover, the baselines confused the activity *front crawl* with *breast stroke*, which are both swimming styles. Another interesting analysis is the confusion from the activity *basketball* and *volleyball spiking*, where we note that the confusion lies on the fact that both activities start with a jump followed



(a) Magnitude-Orientation Stream (MOS)

(b) Non-weighted Linear Fusion

Figure 5.4. Confusion matrices from the NTU RGB+D 60 dataset focused on activities that involves interactions of two people.



(a) Kicking

(b) Touch Pocket

(c) Shaking Hands

Figure 5.5. Example of activities involving interactions of two people on the NTU RGB+D 60 dataset. As can be noted, people are in different depth planes during such activities.

by a arm movement with a ball. Furthermore, the activity *rafting* is highly confused with *kayaking* which can be explained by the fact that both take place in a river with some type of boat but they differ in velocity.

The correct classification of the aforementioned activities by our MOS approach shows that feeding the network with explicit orientation information instead of x and y displacements might improve the classification of activities with movements on very close areas or even with similar movements. Besides, we might note the importance of using magnitude information (velocity) since the velocity information can be used to distinguish between similar activities with different velocities.

We also investigated the cases where our method failed. The most misclassified activities correspond to cases, such as *cricket bowling, pizza tossing, walking with dog* and activities involving playing an instrument. Our method confused *cricket bowling* with *bowling*, as both activities are composed of movements with the arm with a ball. In addition, the activity *pizza tossing* is confused with many other activities. Furthermore, the analysis of the misclassified videos revealed that the method had trouble classifying activities that are only distinguishable by the object used as they have very similar movements, such as playing instrument activities (cello, guitar and sitar; or daf, dhol and tabla). The same difficulties were also noted on the baseline methods. Another misclassification of our approach is *walking with dog* with *horse riding*. Such analysis indicates the use of object information could help enhancing the classification.

5.4 Skeleton Domain Convolutional Neural Network based Approach Evaluation

This section describes the experimental results obtained with the SkeleMotion and TSRJI approaches for the activity recognition problem. We first introduce the implementation details and then we compare our proposed approaches to other CNN methods in the literature. To prove that a good structural organization of joints is important to preserve the spatial relations of the skeleton data, we compare our approach with a baseline employing random joints order when creating the representation (i.e., the creation of the chains' order C^t does not take into account any semantic meaning of adjacent joints). Besides the classical skeleton image representation of Du et al. [2015] we also compare with other skeleton image representations used by state-of-the-art approaches [Wang et al., 2016b; Ke et al., 2017; Li et al., 2017, 2018; Yang et al., 2018] on the RGB+D 60 [Shahroudy et al., 2016] dataset as well as to state-of-the-art methods on the NTU RGB+D 120 [Liu et al., 2019] dataset, in which we applied the same split of training and testing data and we employ the evaluation protocols and metrics proposed by their authors.

5.4.1 Implementation Details

To isolate only the contributions brought by the proposed representation to the activity recognition problem, all compared skeleton image representations were implemented and tested on the same datasets and used the same network architecture.

The network architecture employed is the modified version of the CNN proposed by Li et al. [2017] and introduced in Section 3.4.1. Max pooling and ReLU neuron are adopted and the dropout regularization ratio is set to 0.5. The learning rate is set to 0.001 and batch size is set to 1000. The training is stopped after 200 epochs. The loss function employed was the categorical cross-entropy. We opted for using such architecture since it demonstrated good performance and, according to the authors, it can be easily trained from scratch without any pre-training. Moreover, it is superior on its compact model size and fast inference speed as well.

To cope with activities involving multi-person interaction (e.g., shaking hands), we apply a common choice in the literature, which is to stack skeleton image representations of different people as the network input.

To obtain the orientation skeleton image representation θ' we empirically set the parameter $m = 0.004$, as described in Section 3.4.2.

5.4.2 Results and Comparisons

In this section, we present experiments for parameter optimization and report a comparison of our proposed skeleton representation. We used a subset of the NTU RGB+D 60 [Shahroudy et al., 2016] training set (considering cross-view protocol) to perform parameter setting and then used such parameter on the remaining experiments. We focused on the optimization of the number of temporal scales used on temporal scale aggregation (TSA).

	Temporal distances	**Magnitude Acc. (%)**	**Orientation Acc. (%)**
Two Temporal Scales	1, 5	64.9	62.4
	1, 10	67.4	62.9
	1, 15	66.0	64.1
	1, 20	66.1	63.5
Three Temporal Scales	1, 5, 10	68.6	64.6
	1, 10, 20	69.0	**65.4**
	5, 10, 15	**70.1**	65.1
Four Temporal Scales	1, 5, 10, 15	69.6	65.2
	5, 10, 15, 20	67.9	64.4

Table 5.14. Activity recognition accuracy (%) results on a subset of the NTU RGB+D 60 [Shahroudy et al., 2016] dataset when applying temporal scale aggregation (TSA) on our SkeleMotion representation.

To set the number of temporal scales of our SkeleMotion approach, we empirically varied it from two to four temporal scales considering 20 frames in total. Table 5.4.2 shows the results obtained by such variation. We can see that the best result is obtained by using three temporal scales for both magnitude (5, 10, 15) and orientation (1, 10, 20). Moreover, we noticed that the performance tends to saturate or drop when considering four temporal scales.

	Approach	**Cross-subject Acc. (%)**	**Cross-view Acc. (%)**
Baselines	Random joints order	67.8	74.2
	Du et al. [2015]	68.7	73.0
	Wang et al. [2016b]	39.1	35.9
	Ke et al. [2017]	**70.8**	75.5
	Li et al. [2018]	56.8	61.3
	Yang et al. [2018]	69.5	**75.6**
Our results	Orientation	60.6	65.6
	Magnitude	58.4	64.2
	Orientation (TSA)	65.3	73.2
	Magnitude (TSA)	**69.6**	**80.1**
	TSRJI (Stacked)	69.3	**76.7**

Table 5.15. Activity recognition accuracy (%) results on the NTU RGB+D 60 [Shahroudy et al., 2016] dataset. Results for the baselines were obtained running each method implementation.

Table 5.4.2 presents a comparison of our approach with skeleton image representations of the literature. The methods that have more than one "image" per representation ([Wang et al., 2016a] and [Ke et al., 2017]) were stacked to be used as input to the network. The same was performed for our SkeleMotion approach, considering magnitude and orientation, and for the TSRJI (Stacked) approach, considering the images for each reference joint (i.e., S_a, S_b, S_c, S_d). Regarding the cross-subject protocol, the best result was obtained by Reference Joints representation from Ke et al. [2017], which achieved 70.8% accuracy while our best result (SkeleMotion Magnitude (TSA)) achieves a competitive accuracy of 69.6%. Furthermore. we also achieved a close competitive accuracy of 69.3% with our TSRJI (Stacked) approach. It is worth noting that there is a considerable improvement of 12.8 (p.p.) obtained by SkeleMotion Magnitude (TSA) when compared to Li et al. [2017] baseline, which also explicitly encodes motion information. On the other side, the best result on cross-view protocol was obtained by our SkeleMotion Magnitude (TSA) approach, which achieved 80.1% accuracy. There is an improvement of 4.5 (p.p.) when compared to the Tree Structure Skeleton Image (TSSI) from Yang et al. [2018], which was the best baseline result. Again, there is a considerable improvement of 18.8 (p.p.) when compared to Li et al. [2017] baseline. Considering the TSRJI (Stacked) approach, we can note a slight improvement of 1.1

p.p. when compared to the Tree Structure Skeleton Image (TSSI) from Yang et al. [2018] (detailed improvements are shown in Figure 5.6).

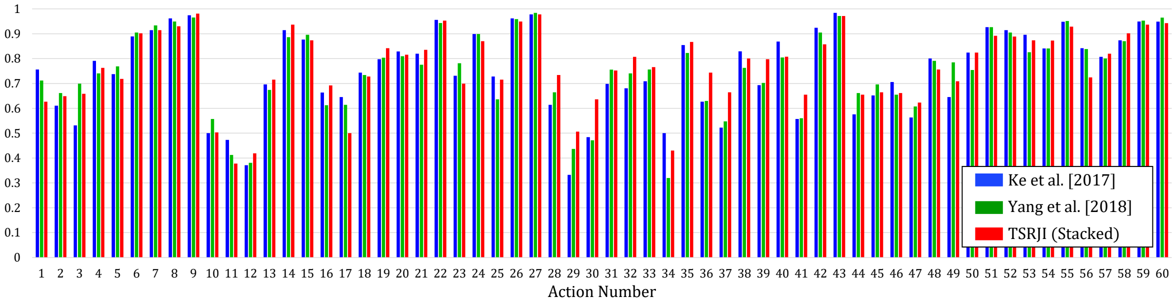


Figure 5.6. Comparison of TSRJI (Stacked) with Ke et al. [2017] and Yang et al. [2018] on the NTU RGB+D 60 [Shahroudy et al., 2016] dataset for cross-view protocol. Best viewed in color.

Comparing to the random joints order baseline (Table 5.4.2), it is worth noting an improvement of 1.5 p.p. on cross-subject protocol and 1.5 p.p. on cross-view protocol obtained by our TSRJI (Stacked). Moreover, SkeleMotion Magnitude (TSA) obtained an improvement of 1.8 p.p. on cross-subject protocol and 5.9 p.p. on cross-view protocol. This shows the importance of keeping a structural organization of joints that preserves spatial relations of relevant joint pairs, bringing semantic meaning of adjacent joints to the representation.

To exploit a possible complementarity of the temporal (Li et al. [2018] or our SkeleMotion) and spatial (Yang et al. [2018] or our TSRJI) skeleton image representations, we combined the different approaches by employing early and late fusion techniques. For the early fusion, we simply stacked the representations to be used as input to the network. On the other hand, the late fusion technique applied was a non-weighted linear combination of the prediction scores of each method. We also employed experiments by using a late fusion technique with our TSRJI. To that end, each reference isolate joint image S is used as input to a CNN. The late fusion technique applied was a non-weighted linear combination of the prediction scores generated by each CNN output. According to the results showed in Table 5.4.2, any type of combination performed with our SkeleMotion provides better results than their solo versions. It is worth noting that our TSRJI with late fusion technique for each reference joint image S provides an improvement of 4.0 p.p. on cross-subject protocol and 3.6 p.p. on cross-view protocol when compared to its early fusion (Stacked) version. Regarding cross-subject protocol, our best results achieves 73.5% accuracy with early fusion technique against the 77.9% of the late fusion approach. Furthermore, on cross-view protocol, our best results achieves 82.4% accuracy with early fusion technique against the 86.1% of the late fusion approach. Detailed improvements are shown in Figures 5.7 and 5.8.

	Approach	**Cross-subject Acc. (%)**	**Cross-view Acc. (%)**
Early Fusion	TSRJI (Stacked)	69.3	76.7
	Magnitude-Orientation	65.6	71.1
	Magnitude-Orientation + [Yang et al., 2018]	70.1	78.8
	Magnitude-Orientation (TSA)	70.5	78.7
	Magnitude (TSA) + [Yang et al., 2018]	71.7	**82.4**
	Orientation (TSA) + [Yang et al., 2018]	69.6	78.9
	Magnitude-Orientation (TSA) + [Yang et al., 2018]	**73.5**	82.1
Late Fusion	TSRJI	73.3	80.3
	Magnitude-Orientation	65.6	71.1
	Magnitude-Orientation + [Yang et al., 2018]	73.2	79.5
	Magnitude-Orientation (TSA)	72.2	81.7
	Magnitude (TSA) + [Yang et al., 2018]	75.4	83.2
	Orientation (TSA) + [Yang et al., 2018]	73.6	80.6
	Magnitude-Orientation (TSA) + [Yang et al., 2018]	76.5	84.7
	Magnitude (TSA) + TSRJI	76.5	84.6
	Orientation (TSA) + TSRJI	75.7	83.5
	Magnitude-Orientation (TSA) + TSRJI	**77.9**	**86.1**

Table 5.16. Comparison between late and early fusion techniques on the NTU RGB+D 60 [Shahroudy et al., 2016] dataset.

Finally, Table 5.4.2 presents the experiments of our proposed skeleton image representations on the NTU RGB+D 120 [Liu et al., 2019] dataset. Due to the results obtained on Table 5.4.2, here we employed the late fusion scheme for method combination. We obtained good results with our SkeleMotion and TSRJI representations outperforming many skeleton-based methods [Shahroudy et al., 2016; Hu et al., 2018, 2017; Liu et al., 2016; Liu et al., 2018a, 2017a; Ke et al., 2017; Liu et al., 2019a; Liu et al., 2017; Liu et al., 2018b; Ke et al., 2018; Liu & Yuan, 2018]. By just employing TSRJI, we achieved state-of-the-art results, outperforming the best reported method (Body Pose Evolution Map Liu & Yuan [2018]) on cross-subject protocol (accuracy of 65.5%). When combining our representations we also achieve state-of-the-art results, outperforming Liu & Yuan [2018] by up to 6.5 p.p. on cross-subject protocol and achieve competitive results on cross-setup protocol (up to 0.8 p.p. better).

In comparison with LSTM approaches, we outperform the best reported method (Two-Stream Attention LSTM) by 9.9 p.p. on cross-subject protocol. Regarding the cross-setup protocol, we outperform them by 4.4 p.p. using our combined skeleton im-

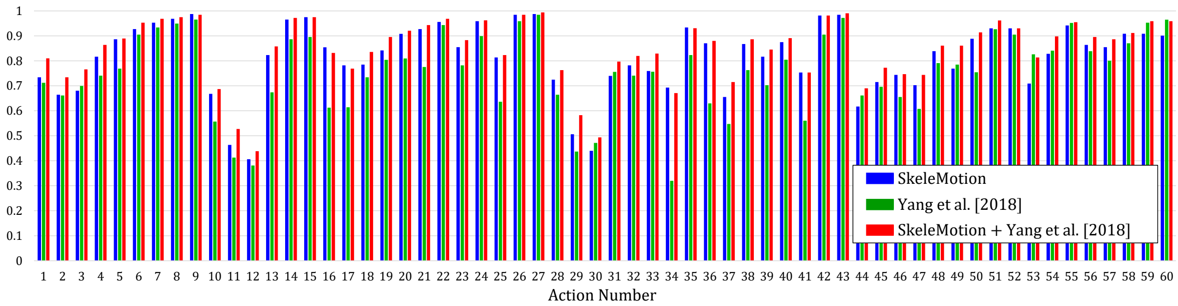


Figure 5.7. The complementarity between SkeleMotion (Magnitude-Orientation TSA) and Yang et al. [2018] TSSI representation on the NTU RGB+D 60 [Shahroudy et al., 2016] dataset for cross-view protocol. Best viewed in color.

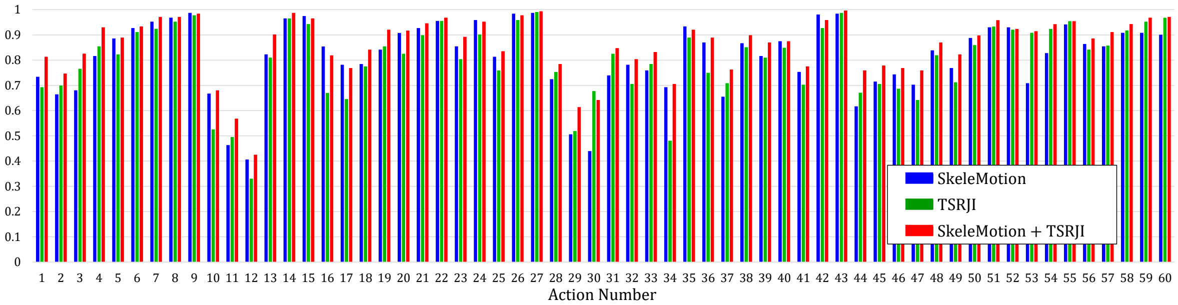


Figure 5.8. The complementarity between SkeleMotion (Magnitude-Orientation TSA) and our TSRJI on the NTU RGB+D 60 [Shahroudy et al., 2016] dataset for cross-view protocol. Best viewed in color.

age representations. This indicates that, our skeleton image representation approaches used as input for CNNs leads to a better learning of temporal dynamics than the approaches that employs LSTM.

5.4.3 Discussion

Since our proposed TSRJI representation is based on the combination of the tree structural organization from Yang et al. [2018] and the reference joints technique from Ke et al. [2017], we better analyze our achieved results by taking a closer look at the activities from the NTU RGB+D 60 dataset that our method achieved higher performance than Ke et al. [2017] and Yang et al. [2018]. To that end, Figure 5.6, presents the detailed improvements of our TSRJI (Stacked) representation. The activities that were most correctly classified by TSRJI (Stacked) and misclassified by the baselines are: *standing up (9); writing (12); tear up paper (13); wear jacket (14); wear a shoe (16); take off glasses (19); take off a hat cap (21); make a phone call (28); playing with phone (29); typing on a keyboard (30); taking a selfie (32); check time (33); nod head*

	Approach	**Cross-subject Acc. (%)**	**Cross-setup Acc. (%)**
Literature results	Part-Aware LSTM [Shahroudy et al., 2016]	25.5	26.3
	Soft RNN [Hu et al., 2018]	36.3	44.9
	Dynamic Skeleton [Hu et al., 2017]	50.8	54.7
	Spatio-Temporal LSTM [Liu et al., 2016]	55.7	57.9
	Internal Feature Fusion [Liu et al., 2018a]	58.2	60.9
	GCA-LSTM [Liu et al., 2017a]	58.3	59.2
	Multi-Task Learning Network [Ke et al., 2017]	58.4	57.9
	FSNet [Liu et al., 2019a]	59.9	62.4
	Skeleton Visualization [Liu et al., 2017]	60.3	63.2
	Two-Stream Attention LSTM [Liu et al., 2018b]	61.2	63.3
	Multi-Task CNN with RotClips [Ke et al., 2018]	62.2	61.8
	[Yang et al., 2018] + [Li et al., 2018]	63.7	60.8
	Body Pose Evolution Map [Liu & Yuan, 2018]	**64.6**	**66.9**
Our results	TSRJI	65.5	59.7
	TSRJI + [Li et al., 2018]	67.9	62.8
	Orientation (TSA)	52.2	54.1
	Magnitude (TSA)	57.6	60.4
	Magnitude-Orientation (TSA)	62.9	63.0
	Magnitude-Orientation (TSA) + [Yang et al., 2018]	67.7	66.9
	Magnitude-Orientation (TSA) + TSRJI	**71.1**	**67.7**

Table 5.17. Activity recognition accuracy (%) results on the NTU RGB+D 120 [Liu et al., 2019] dataset. Results for literature methods were obtained from [Liu et al., 2019].

bow (35); shake head (36); wipe face (37); sneeze or cough (41); point finger at the other person (54); touch other person pocket (57); and handshaking (58)[5]. We note that the baselines usually confused such activities, which are activities involving arm and hand movements.

We also analyze our results when employing the late fusion scheme. To better perform such comparison, we combined Ke et al. [2017] and Yang et al. [2018] representations with the same late fusion scheme employed by us. Figure 5.9, presents the detailed improvements of our TSRJI (Late Fusion) representation. For instance, some activities that were most correctly classified by TSRJI (Late Fusion) and misclassified by the baseline are: *brushing teeth (3); make a phone call (28); playing with phone (29); typing on a keyboard (30); wipe face (37); and sneeze or cough (41).* We note that the baseline confused activities involving arm and hand movements. It shows that our proposed representation performs better and provides a richer discriminability

[5] The number in parentheses represents the activity index.

than a simply combination of the based methods.

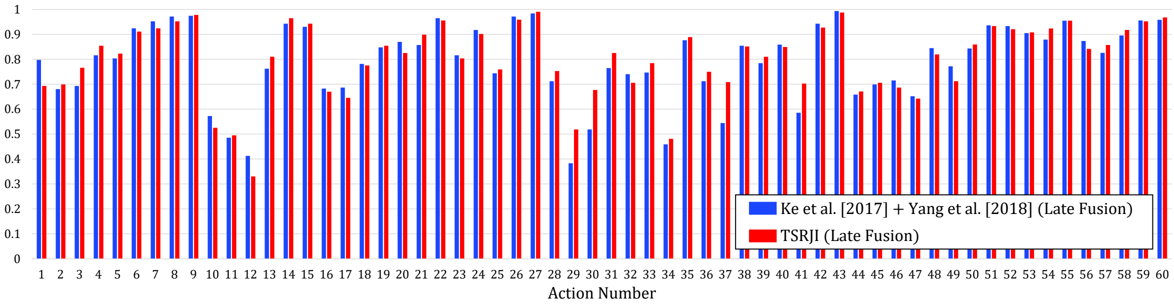


Figure 5.9. Comparison of TSRJI (Late Fusion) with Ke et al. [2017] + Yang et al. [2018] (Late Fusion) on the NTU RGB+D 60 Shahroudy et al. [2016] dataset for cross-view protocol. Best viewed in color.

The correct classifications of the aforementioned activities by our TSRJI representation show that feeding the network with explicit structural organization of relevant joint pairs might improve the classification. We believe that the reference joints technique helped improving such activities since the shoulders were two of the reference joints. In view of that, since such joints are stable they could reflect the motions of the arms and hand joints. Furthermore, the spatial relations of adjacent joint pairs were preserved by the use of the depth-first tree traversal order algorithm bringing more semantic meaning to the representation.

We also investigated the cases in which our approaches failed. The common misclassified activities from SkeleMotion and TSRJI representations correspond to cases, such as *clapping (10), rub two hands together (34), reading (11), writing (12), typing on a keyboard (30), put the palms together (39)* and *cross hands in front(40)*. Our approaches confused *clapping (10)* with *rub two hands together (34)*, as both activities are constituted by closer movements with the hands. Furthermore, the analysis of the misclassified videos revealed that the method had trouble classifying activities that are only distinguishable by the object used as they have very similar movements (e.g., the activities *writing (12)* is confused with *reading (11)*, *typing on a keyboard (30)* and *playing with phone (29)*).

Considering only the misclassification of TSRJI, we observed that it confuses *wear a shoe (16)* with *take off a shoe (17)* and *pointing to something (31)* with *taking a selfie (54)*. Such analysis indicates that the use of explicit motion information could help enhancing the classification. Figure 5.11 illustrates the confusion matrix of our TSRJI representation.

Regarding the SkeleMotion misclassification, we observed that it confuses *pat on back of other person (53)* and *point finger at the other person (54)*. These activities

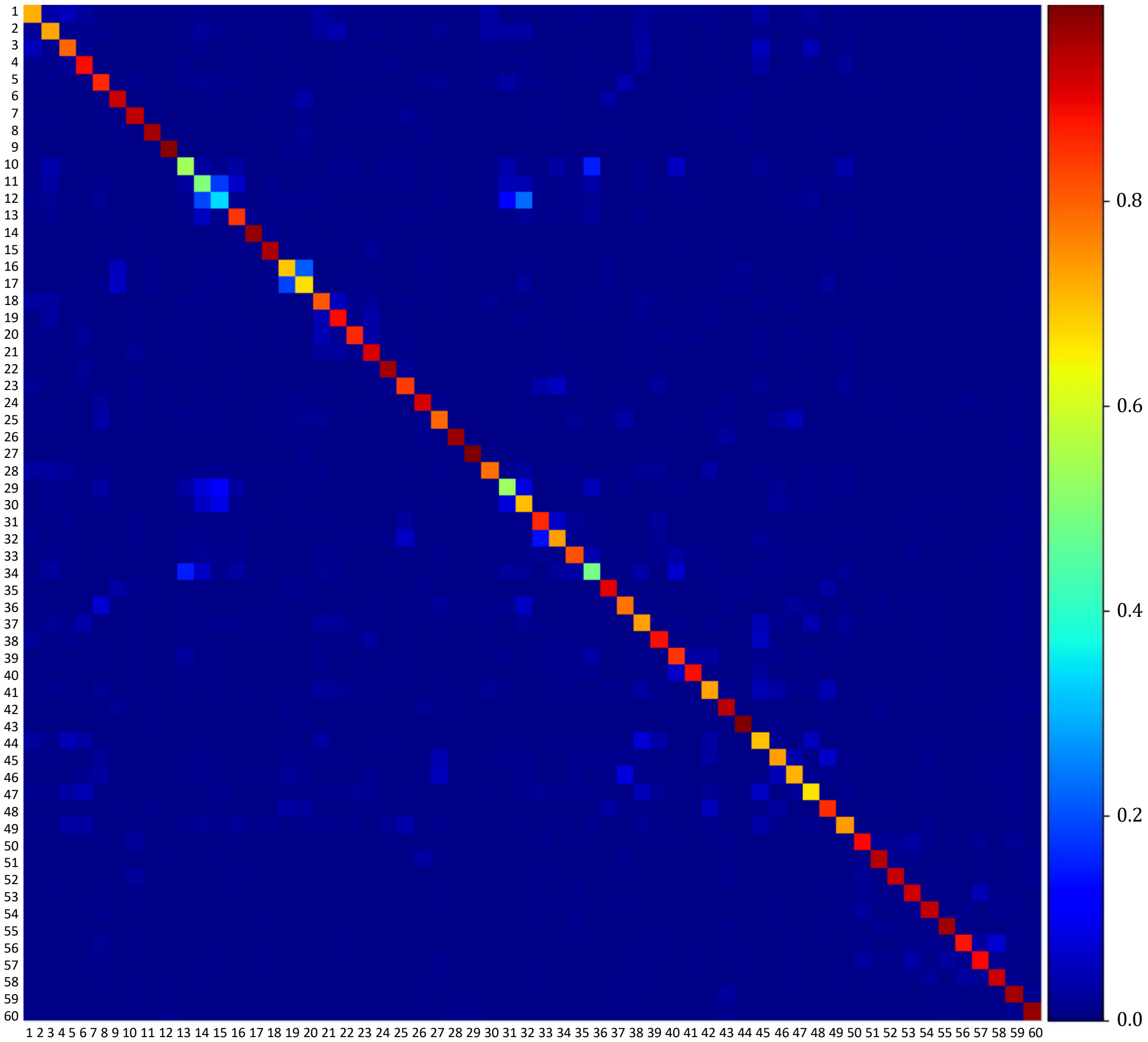


Figure 5.10. Confusion matrix of TSRJI (Late Fusion) on the NTU RGB+D 60 Shahroudy et al. [2016] dataset. Best viewed in color.

are very similar, being distinguishable by the fact that one of the subjects involved in the scene is with his back to the other. Such analysis indicates that the use of spatial structure of joints information could help enhancing the classification. Figure 5.11 illustrates the confusion matrix of our SkeleMotion.

As mentioned on the last analysis, the difficulties encountered in some misclassification of both methods are related to qualities captured by each other method (motion for SkeleMotion and spatial joint information in TSRJI). Thus, it is possible to observe in the confusion matrix of the late fusion from SkeleMotion + TSRJI (Figure 5.12) that some misclassification were improved or solved (e.g., *pat on back of other person (53)* and *point finger at the other person (54)*).

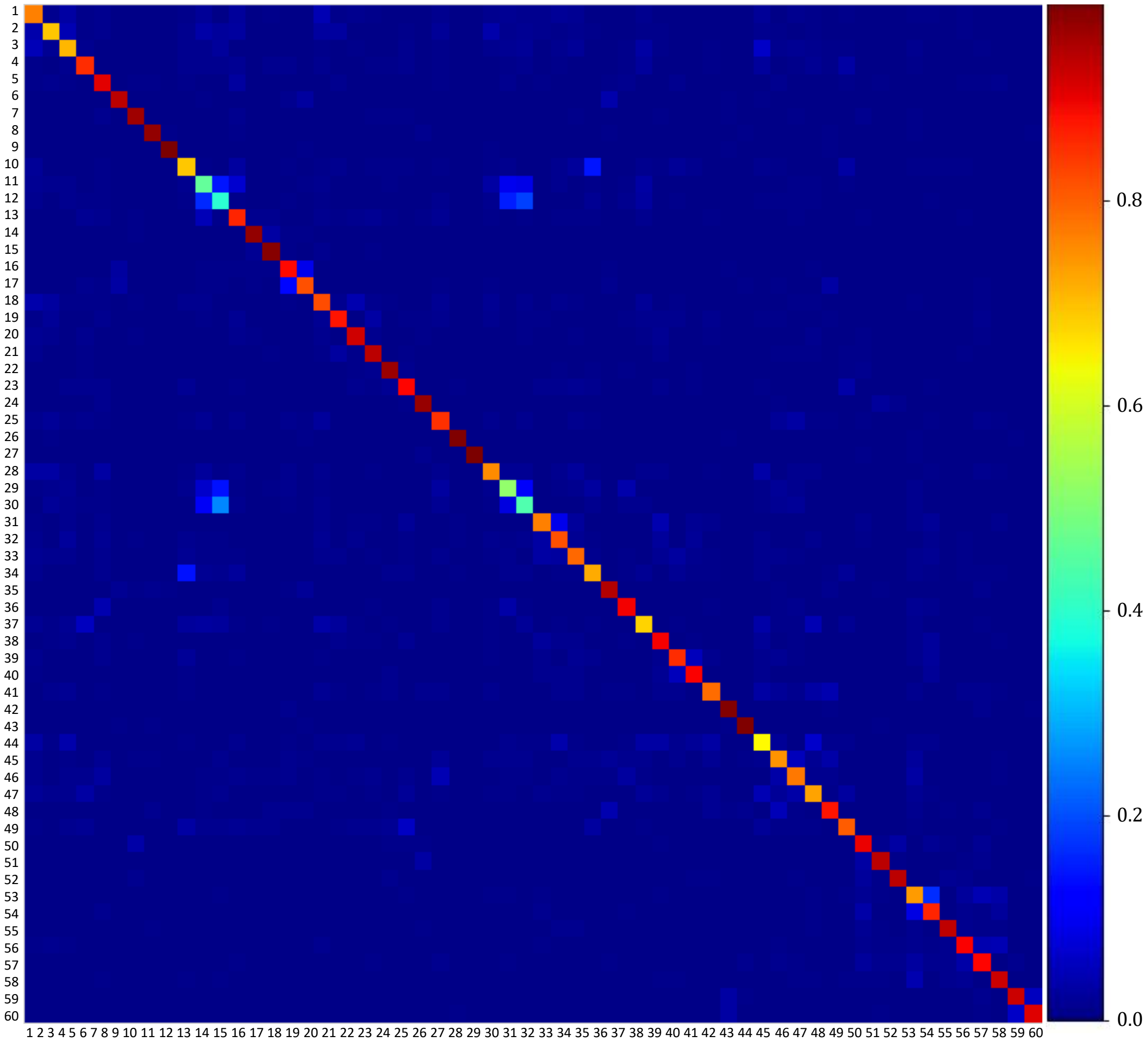


Figure 5.11. Confusion matrix of SkeleMotion on NTU RGB+D 60 Shahroudy et al. [2016] dataset. Best viewed in color.

5.5 Concluding Remarks

In this chapter, we presented the experimental results achieved by our proposed representations divided into three different groups. First, we have shown the experiments from OFCM spatiotemporal local feature descriptor. The improvement of OFCM over the other descriptors is especially striking. We believe the reason for this is that our spatiotemporal feature is capturing important temporal motion information since a pair (co-occurrence) of magnitudes or orientations have more description power than a single histogram bin of gradient, magnitude or orientation. In this way, the OFCM can express motion in more details than the other histogram-based features, which use single gradient orientation or single flow orientations discarding important information

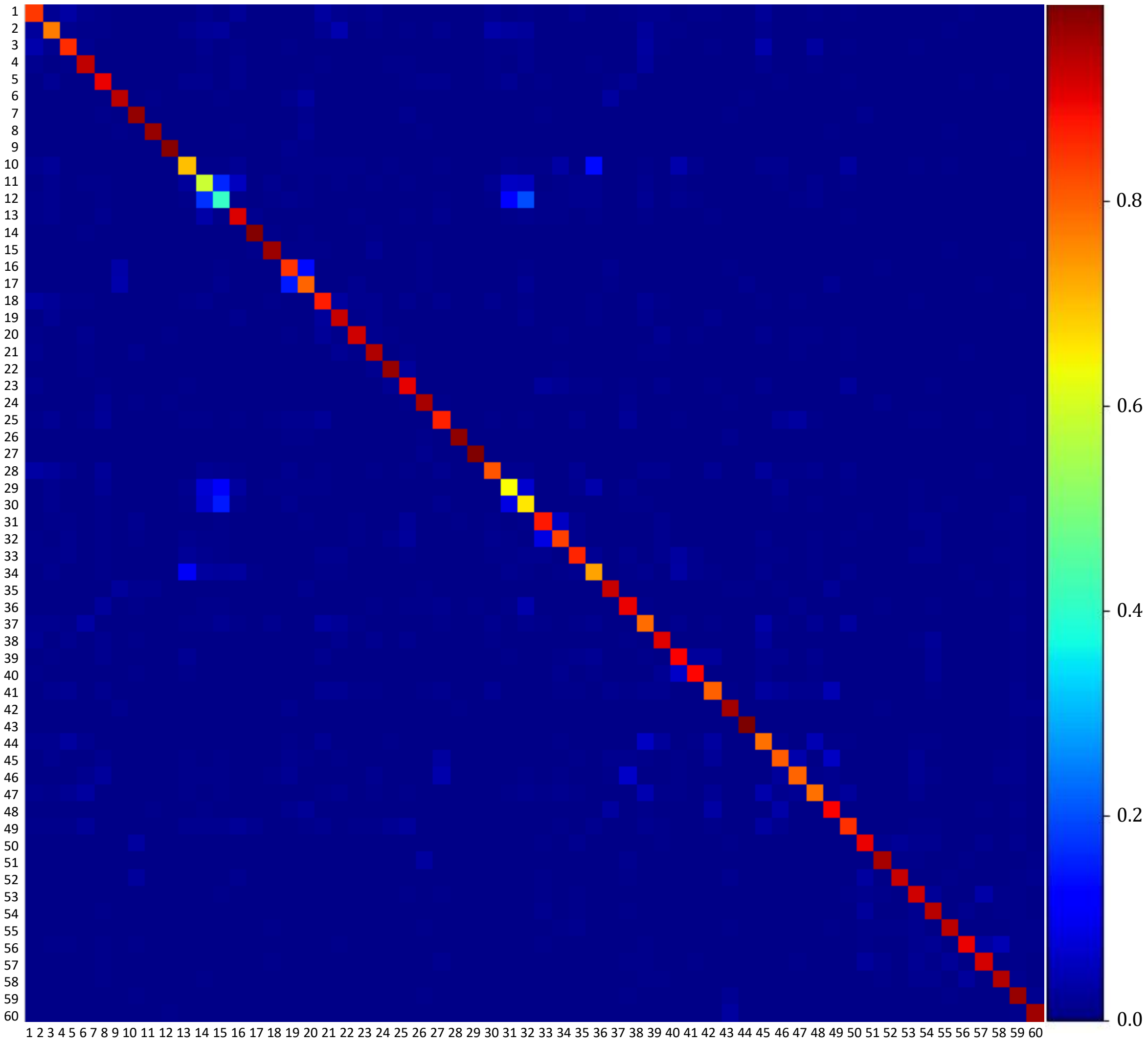


Figure 5.12. Confusion matrix of late fusion from SkeleMotion + TSRJI on the NTU RGB+D 60 Shahroudy et al. [2016] dataset. Best viewed in color.

concerning spatial relations among the flow field.

In the the second group of experiments, we evaluated the proposed mid-level representation, named CCW. The improvement achieved regarding the compactness of CCW representation over the BoW model is very significant. We believe the reason for such improvement lies on the information extracted from co-occurrence related to global structures in various local region-based features, moreover, a pair (co-occurrence) of codewords have more "vocabulary" than a single histogram bin of codewords. In this way, CCW can encode the information in more details than the BoW model which is a histogram-based features that use single codewords discarding important information concerning spatial relations among the features.

We presented the experiments regarding our image domain CNN-based methods, MOS and MOS+D, in the third group. We showed that our temporal stream provides better recognition accuracy than other isolate streams - i.e., only using spatial stream or temporal stream of the literature compared to our MOS temporal stream. We showed that our MOS approach learns complementary information to TSN method. The comparison between the performed experiments and state-of-the-art methods confirmed the relevance of the use of magnitude and orientation information to learn motion for activity recognition. Furthermore, we performed experiments employing depth information estimated from the RGB video data. We used it as weighting scheme on the magnitude information to compensate the distance of the subjects performing the activity. We are able to conclude that if the depth maps were not well estimated, the magnitude information might be deteriorated.

Finally, in the forth group, we presented the experimental results achieved by our skeleton domain CNN-based method, SkeleMotion and TSRJI. Again, the performed experiments with comparison to the state-of-the-art methods confirmed the relevance of the use of magnitude and orientation information to be used for motion learning for the activity recognition task. However, it is important to emphasize that here the magnitude and orientation are extracted from the skeleton joints information. Moreover, we showed that the structural organization of joints preserves spatial relations of more relevant joint pairs. Another interesting finding is the complementarity of our skeleton image representations, which improves the 3D activity recognition.

Chapter 6

Conclusions

In this chapter, we present our concluding remarks by providing the main contributions of this dissertation. Moreover, we register our thoughts about possible future work directions.

6.1 Contributions

We have presented and evaluated four novel representation methods for the activity recognition problem: (i) a local spatiotemporal feature descriptor; (ii) a mid-level representation; (iii) optical flow deep learning-based representations; and (iv) skeleton deep learning-based representations. We have demonstrated that the proposed representations outperform the state of the art on many challenging well-known datasets. Among the contributions of our work, we emphasize:

- Definition and implementation of a novel local spatiotemporal feature descriptor for the activity recognition task, named OFCM (Section 3.1). The representation is based on the co-occurrence matrices computed over magnitude and orientation information. These co-occurrence matrices express the distribution of velocity and direction components at a given offset over the optical flow. Our approach has the advantage of expressing motion in more details than other histogram-based features, which use single gradient orientation or single flow orientations, since a pair (co-occurrence) of magnitudes or orientations have more description power than a single histogram bin.

- Definition and implementation of a novel mid-level representation for the activity recognition task, named CCW (Section 3.2). This representation uses co-occurrence matrices computed over feature codewords. Such co-occurrence matrices express the distribution of features at a given offset over feature codewords from a pre-computed codebook. CCW has the advantage of being extremely compact when compared to the classical BoW model. The reason for such improvement lies on the information extracted from co-occurrence related to global structures in various local region-based features. In this way, the CCW representation encodes information in more details than the BoW model, which does not encodes information concerning spatial relations among features.

- Definition and implementation of novel image domain CNN-based representations, named MOS and MOS+D (Section 3.3). These representations utilize non-linear transformations on the optical flow components aiming to generate magnitude-orientation input images for a temporal stream. Our temporal stream has the advantage of capturing displacement information by using orientation of the optical flow and velocity of the movement considering the optical flow magnitude.
- Definition and implementation of novel skeleton domain CNN-based representations, named SkeleMotion (Section 3.4.2) and TSRJI (Section 3.4.3). SkeleMotion is based on temporal dynamics encoding and explicitly uses motion information (magnitude and orientation) of skeleton joints. On the other hand, TSRJI takes advantage of a structural organization of joints that preserves spatial relations of more relevant joint pairs and also incorporates different spatial relationships between the joints.
- We have presented an empirical evaluation of the aforementioned representations. We have demonstrated that the proposed representations achieved better or similar accuracy results in comparison to the state of the art on several challenging activity recognition datasets (KTH, UCF Sports, HMDB51, UCF101, NTU RGB+D 60 and NTU RGB+D 120).

6.2 Future Work

The purpose of this section is to analyze the open questions raised in this dissertation as future work.

Haralick Features Study

In Chapter 5 we have experimentally evaluated OFCM and CCW representations. To that end, we used 12 Haralick textural features in all experiments. Future work consists in investigating the impact of each Haralick feature on the magnitude and orientation co-occurrence matrices as well as on the feature co-occurrence matrices. Such analysis is of utmost importance, since the number of Haralick features used directly impacts the final size of both representations.

Input Motion Evaluation

In Chapter 5, we have experimentally evaluated MOS and MOS+D representations. To that end, we used the TVL1 optical flow algorithm in all experiments to generate the magnitude-orientation input images. Although it showed good results, there are other optical flow algorithms in the literature (e.g., Brox algorithm [Brox et al., 2004] and Farneback algorithm [Farnebäck, 2003]) as well as other input motion approaches (e.g., MPEG motion vectors [Perez et al., 2017]) that suggest further investigation to be used to estimate the magnitude-orientation input images.

Deep Learning Architecture Analysis

In Chapter 5, we have experimentally evaluated our skeleton domain CNN-based representations. To that end, we employed a modified version of the CNN architecture proposed by Li et al. [2017]. Despite showing promising results, there are other CNN architectures in the literature (e.g., ResNet-50 [He et al., 2016]) that suggest further investigation to be used as backbone during the learning phase.

Attention Mechanism Employment

Recently, some works in the literature employed attention mechanisms in CNNs to make the network focus more on crucial and discriminative temporal and spatial features [Wang et al., 2017; Woo et al., 2018; Chi et al., 2019; Liu et al., 2019b]. To that end, a possible direction for future work consists in evaluating the behavior of the proposed CNN-based representations in conjunction with attention mechanisms.

Open Set Problem

All the experiments performed in this dissertation were based on a closed-set scenario - i.e., all testing classes are known while training. However, in a more realistic scenario situations that have not been seen before may arise. Considering the representations introduced in this dissertation, if they were applied in such scenario, the model would classify it as one of the classes learned during the training phase. In view of that, direction for future work consists in an adaptation of the proposed representations in a class of learning problems known as open-set recognition [Oza & Patel, 2019].

Generalization Power

To evaluate the generalization power of the features learned by the proposed representations, future work consists in applying cross-dataset evaluations. Moreover, all representations proposed in this dissertation were developed for the activity recognition problem. However, we believe that all four proposed approaches can be applied to other computer vision applications involving video description. To that end, a possible direction for future work consists in evaluating the behavior of proposed representations in other video-related tasks - e.g., activity detection or scene description.

6.3 Publications

Journals

- **C. Caetano**, V. H. C. D. Melo, F. Brémond, J. A. dos Santos and W. R. Schwartz. Magnitude-Orientation Stream Network and Depth Information applied to Activity Recognition. Journal of Visual Communication and Image Representation (JVCI): Special Issue on Feature representations for medical images and activity understanding, 2019.

International Conferences

- **C. Caetano**, J. A. dos Santos and W. R. Schwartz. Optical Flow Co-occurrence Matrices: A novel spatiotemporal feature descriptor. In International Conference on Pattern Recognition (ICPR), Mexico, Cancun, 2016.
- **C. Caetano**, V. H. C. D. Melo, J. A. dos Santos and W. R. Schwartz. Activity recognition based on a magnitude-orientation stream network. Conference on Graphics, Patterns and Images (SIBGRAPI), Niterói, Brazil, 2017. **(Awarded as the best Computer Vision/Image Processing/Pattern Recognition main track paper award).**
- **C. Caetano**, J. A. dos Santos and W. R. Schwartz. Statistical measures from co-occurrence of codewords for action recognition. International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP), Madeira, Portugal, 2018.

- **C. Caetano**, J. Sena, F. Brémond, J. A. dos Santos and W. R. Schwartz. SkeleMotion: A New Representation of Skeleton Joint Sequences Based on Motion Information for 3D Action Recognition. IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, Taiwan, 2019.
- **C. Caetano**, F. Brémond and W. R. Schwartz. Skeleton Image Representation for 3D Action Recognition based on Tree Structure and Reference Joints. Conference on Graphics, Patterns and Images (SIBGRAPI), Rio de Janeiro, Brazil, 2019.

Bibliography

Aggarwal, J. & Ryoo, M. (2011). Human activity analysis: A review. *ACM Comput. Surv.* 1, 9, 49

Banerjee, P. & Nevatia, R. (2011). Learning neighborhood cooccurrence statistics of sparse features for human activity recognition. Em *2011 8th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS).* 8, 13, 19, 63, 67, 68

Bouguet, J. (2000). Pyramidal implementation of the lucas kanade feature tracker. *Intel Corporation, Microprocessor Research Labs.* 29

Brox, T.; Bruhn, A.; Papenberg, N. & Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. Em *ECCV.* 93

Caetano, C.; de Melo, V. H.; Brémond, F.; dos Santos, J. A. & Schwartz, W. R. (2019a). Magnitude-orientation stream network and depth information applied to activity recognition. *Journal of Visual Communication and Image Representation.* 9

Caetano, C.; dos Santos, J. A. & Schwartz, W. R. (2016). Optical Flow Co-occurrence Matrices: A novel spatiotemporal feature descriptor. Em *ICPR.* 8

Caetano, C.; dos Santos, J. A. & Schwartz, W. R. (2018). Statistical measures from co-occurrence of codewords for action recognition. Em *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,.* 8

Caetano, C. A.; Brémond, F. & Schwartz, W. R. (2019b). Skeleton image representation for 3d action recognition based on tree structure and reference joints. Em *2019 32th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI).* 9

Caetano, C. A.; Melo, V. H. C. D.; dos Santos, J. A. & Schwartz, W. R. (2017). Activity recognition based on a magnitude-orientation stream network. Em *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI).* 9

Caetano, C. A.; Sena, J.; Brémond, F.; dos Santos, J. A. & Schwartz, W. R. (2019c). Skelemotion: A new representation of skeleton joint sequences based on motion information for 3d action recognition. Em *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS).* 9

Cai, Z.; Wang, L.; Peng, X. & Qiao, Y. (2014). Multi-view Super Vector for Action Recognition. Em *CVPR.* 75

Carreira, J. & Zisserman, A. (2017). Quo vadis, action recognition? A new model and the Kinetics dataset. Em *CVPR.* 3, 22, 27, 75

Chi, L.; Tian, G.; Mu, Y. & Tian, Q. (2019). Two-stream video classification with cross-modality attention. Em *ICCVW.* 93

Choutas, V.; Weinzaepfel, P.; Revaud, J. & Schmid, C. (2018). Potion: Pose motion representation for action recognition. Em *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 22, 27

Colque, R. V. H. M.; Caetano, C.; de Andrade, M. T. L. & Schwartz, W. R. (2017). Histograms of Optical Flow Orientation and Magnitude and Entropy to Detect Anomalous Events in Videos. *IEEE Transactions on Circuits and Systems for Video Technology.* 15

Colque, R. V. H. M.; Caetano, C. & Schwartz, W. R. (2015). Histograms of Optical Flow Orientation and Magnitude to Detect Anomalous Events in Videos. Em *SIBGRAPI.* 15

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S. & Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. Em *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 70

Dalal, N. & Triggs, B. (2005). Histograms of oriented gradients for human detection. Em *CVPR.* 2, 14, 59, 63

Dalal, N.; Triggs, B. & Schmid, C. (2006). Human Detection Using Oriented Histograms of Flow and Appearance. Em *ECCV.* 2, 14, 27, 59, 63, 75

Danafar, S. & Gheissari, N. (2007). Action recognition for surveillance applications using optic flow and SVM. Em *ACCV.* 1

Devanne, M.; Wannous, H.; Berretti, S.; Pala, P.; Daoudi, M. & Del Bimbo, A. (2015). 3-d human action recognition by shape analysis of motion trajectories on riemannian manifold. *IEEE Transactions on Cybernetics.* 3

Diba, A.; Pazandeh, A. M. & Van Gool, L. (2016). Efficient Two-Stream Motion and Appearance 3D CNNs for Video Classification. Em *ECCV.* 5

Dollar, P.; Rabaud, V.; Cottrell, G. & Belongie, S. (2005). Behavior recognition via sparse spatio-temporal features. Em *PETS.* 2, 25

Du, Y.; Fu, Y. & Wang, L. (2015). Skeleton based action recognition with convolutional neural network. Em *IAPR Asian Conference on Pattern Recognition (ACPR).* 3, 22, 23, 24, 25, 27, 78, 80

Farnebäck, G. (2003). Two-frame motion estimation based on polynomial expansion. Em *Proceedings of the 13th Scandinavian Conference on Image Analysis.* 93

Feichtenhofer, C.; Pinz, A. & Zisserman, A. (2016). Convolutional two-stream network fusion for video action recognition. Em *CVPR*. 5, 25, 68, 75

Godard, C.; Mac Aodha, O. & Brostow, G. J. (2017). Unsupervised monocular depth estimation with left-right consistency. Em *CVPR*. 35, 39, 70

Gowayyed, M. A.; Torki, M.; Hussein, M. E. & El-Saban, M. (2013). Histogram of oriented displacements (hod): Describing trajectories of human joints for action recognition. Em *International Joint Conference on Artificial Intelligence (IJCAI)*. 3

Han, F.; Reily, B.; Hoff, W. & Zhang, H. (2017). Space-time representation of people based on 3d skeletal data. *Computer Vision and Image Understanding*. 3, 17

Haralick, R. M.; Shanmugam, K. S. & Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*. 12, 13, 29, 31, 111

He, K.; Zhang, X.; Ren, S. & Sun, J. (2016). Deep residual learning for image recognition. Em *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 24, 93

Herath, S.; Harandi, M. & Porikli, F. (2017). Going deeper into action recognition. *Image Vision Comput*. 3

Hu, J.; Zheng, W.; Lai, J. & Zhang, J. (2017). Jointly learning heterogeneous features for rgb-d activity recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 82, 84

Hu, J.; Zheng, W.; Ma, L.; Wang, G.; Lai, J. & Zhang, J. (2018). Early action prediction by soft regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 82, 84

Hussein, M. E.; Torki, M.; Gowayyed, M. A. & El-Saban, M. (2013). Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. Em *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*. 3, 17

Ioffe, S. & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Em *ICML*. 21

Jain, R. (1991). *The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, and modeling*. Wiley. 72, 73

Ji, S.; Xu, W.; Yang, M. & Yu, K. (2013). 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Trans. Pattern Anal. Mach. Intell*. 20

Jiang, B.; Jiang, B.; Gan, W.; Wu, W. & Yan, J. (2019). STM: SpatioTemporal and Motion Encoding for Action Recognition. Em *ICCV*. 75

Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R. & Fei-Fei, L. (2014). Large-Scale Video Classification with Convolutional Neural Networks. Em *CVPR*. 20, 25, 74, 75

Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P. et al. (2017). The kinetics human action video dataset. Relatório técnico, arXiv preprint arXiv:1705.06950. 22

Ke, Q.; Bennamoun, M.; An, S.; Sohel, F. & Boussaid, F. (2017). A new representation of skeleton sequences for 3d action recognition. Em *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* xvi, 9, 22, 24, 26, 27, 41, 45, 46, 78, 80, 81, 82, 83, 84, 85

Ke, Q.; Bennamoun, M.; An, S.; Sohel, F. & Boussaid, F. (2018). Learning clip representations for skeleton-based 3d action recognition. *IEEE Transactions on Image Processing.* 82, 84

Keval, H. (2006). CCTV Control Room Collaboration and Communication: Does it Work? Em *Human Centred Technology Workshop.* 1

Kläser, A.; Marszałek, M. & Schmid, C. (2008). A Spatio-Temporal Descriptor Based on 3D-Gradients. Em *BMVC.* 14, 28, 59, 60, 63, 64, 75

Kobayashi, T. & Otsu, N. (2008). Image feature extraction using gradient local auto-correlations. Em *ECCV.* 8, 13

Kobayashi, T. & Otsu, N. (2012). Motion recognition using local auto-correlation of space-time gradients. *Pattern Recogn. Lett.*, 33:1188--1195. 8, 13

Krig, S. (2014). Interest point detector and feature descriptor survey. Em *Computer Vision Metrics.* Apress. 2

Krizhevsky, A.; Sutskever, I. & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Em *NIPS.* 20, 23, 24, 69

Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T. & Serre, T. (2011). HMDB: A large video database for human motion recognition. Em *ICCV.* xvi, xix, 9, 39, 51, 52, 53, 58

Laptev, I. (2005). On space-time interest points. *IJCV.* 19

Laptev, I. & Lindeberg, T. (2006). Local descriptors for spatio-temporal recognition. Em *Spatial Coherence for Visual Motion Analysis.* Springer Berlin Heidelberg. 2

Laptev, I.; Marszalek, M.; Schmid, C. & Rozenfeld, B. (2008). Learning realistic human actions from movies. Em *CVPR.* 14, 27, 28, 59, 63, 75

Li, C.; Zhong, Q.; Xie, D. & Pu, S. (2017). Skeleton-based action recognition with convolutional neural networks. Em *IEEE International Conference on Multimedia Expo Workshops (ICMEW).* 22, 24, 27, 41, 78, 80, 93

Li, C.; Zhong, Q.; Xie, D. & Pu, S. (2018). Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. Em *International Joint Conference on Artificial Intelligence (IJCAI).* 22, 24, 27, 42, 43, 78, 80, 81, 84

Liang, B. & Zheng, L. (2015). A survey on human action recognition using depth sensors. Em *International Conference on Digital Image Computing: Techniques and Applications (DICTA).* 38

Liu, D.; Hua, G.; Viola, P. & Chen, T. (2008). Integrated feature selection and higher-order spatial feature extraction for object categorization. Em *CVPR.* 19

Liu, J.; Shahroudy, A.; Perez, M.; Wang, G.; Duan, L.-Y. & Kot, A. C. (2019). Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* xx, 9, 57, 58, 78, 82, 84

Liu, J.; Shahroudy, A.; Wang, G.; Duan, L. & Kot Chichung, A. (2019a). Skeleton-based online action prediction using scale selection network. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 82, 84

Liu, J.; Shahroudy, A.; Xu, D.; Kot, A. C. & Wang, G. (2018a). Skeleton-based action recognition using spatio-temporal lstm network with trust gates. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 82, 84

Liu, J.; Shahroudy, A.; Xu, D. & Wang, G. (2016). Spatio-temporal lstm with trust gates for 3d human action recognition. Em *International Conference on Computer Vision (ECCV).* 82, 84

Liu, J.; Wang, G.; Duan, L.; Abdiyeva, K. & Kot, A. C. (2018b). Skeleton-based human action recognition with global context-aware attention lstm networks. *IEEE Transactions on Image Processing.* 82, 84

Liu, J.; Wang, G.; Hu, P.; Duan, L. & Kot, A. C. (2017a). Global context-aware attention lstm networks for 3d action recognition. Em *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 82, 84

Liu, M.; Chen, C. & Liu, H. (2017b). 3d action recognition using data visualization and convolutional neural networks. Em *IEEE International Conference on Multimedia Expo Workshops (ICME).* 22, 23, 27

Liu, M.; Liu, H. & Chen, C. (2017). Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recogn.* 82, 84

Liu, M. & Yuan, J. (2018). Recognizing human actions as the evolution of pose estimation maps. Em *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 82, 84

Liu, Q.; Che, X. & Bie, M. (2019b). R-stan: Residual spatial-temporal attention network for action recognition. *IEEE Access.* 75, 93

Lloyd, S. (2006). Least squares quantization in pcm. *IEEE Trans. Inf. Theor.* 19

Maki, A.; Seki, A.; Watanabe, T. & Cipolla, R. (2011). Co-occurrence flow for pedestrian detection. Em *ICIP.* 8, 13

Mota, V.; Perez, E.; Maciel, L.; Vieira, M. & Gosselin, P. (2014). A tensor motion descriptor based on histograms of gradients and optical flow. *Pattern Recognition Letters*, 39:85 – 91. 2

Nosaka, R.; Ohkawa, Y. & Fukui, K. (2012). Feature extraction based on co-occurrence of adjacent local binary patterns. Em *PSIVT.* 34

Oliva, A. & Torralba, A. (2007). The role of context in object recognition. *Trends in Cognitive Sciences*, 11:520 – 527. 2, 25

Oza, P. & Patel, V. M. (2019). C2ae: Class conditioned auto-encoder for open-set recognition. Em *CVPR*. 94

Park, E.; Han, X.; Berg, T. L. & Berg, A. C. (2016). Combining multiple sources of knowledge in deep CNNs for action recognition. Em *WACV*. 5, 21

Peng, X.; Wang, L.; Wang, X. & Qiao, Y. (2016). Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *Computer Vision and Image Understanding*. 74, 75

Perez, M.; Avila, S.; Moreira, D.; Moraes, D.; Testoni, V.; Valle, E.; Goldenstein, S. & Rocha, A. (2017). Video pornography detection through deep learning techniques and motion information. *Neurocomputing*. 21, 93

Poppe, R. (2010). A survey on vision-based human action recognition. *Image Vision Comput.*, 28:976--990. 2, 11

Reddy, V.; Sanderson, C. & Lovell, B. (2011). Improved anomaly detection in crowded scenes via cell-based analysis of foreground speed, size and texture. Em *CVPRW*. 1

Richardson, I. E. G. (2003). *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*. John Wiley & Sons, Inc. 21

Rodriguez, M.; Ahmed, J. & Shah, M. (2008). Action mach a spatio-temporal maximum average correlation height filter for action recognition. Em *CVPR*. xvi, xix, xx, 9, 50, 51, 58, 62, 63, 68

Ryan, D.; Denman, S.; Fookes, C. & Sridharan, S. (2011). Textures of optical flow for real-time anomaly detection in crowds. Em *AVSS*. 2, 25, 29

Sánchez, J.; Perronnin, F.; Mensink, T. & Verbeek, J. (2013). Image Classification with the Fisher Vector: Theory and Practice. *Int. J. Comput. Vision*. 3

Schuldt, C.; Laptev, I. & Caputo, B. (2004). Recognizing human actions: A local svm approach. Em *ICPR*. xvi, xix, xx, 2, 9, 49, 50, 58, 61, 62, 63, 65, 66, 68

Shahroudy, A.; Liu, J.; Ng, T. & Wang, G. (2016). Ntu rgb+d: A large scale dataset for 3d human activity analysis. Em *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. xvi, xvii, xx, 9, 22, 54, 56, 57, 58, 68, 76, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 107, 110

Shao, L.; Zhen, X.; Tao, D. & Li, X. (2014). Spatio-temporal laplacian pyramid coding for action recognition. *Cybernetics, IEEE Transactions on*, 44:817–827. 2

Shi, F. (2014). *Local Part Model for Action Recognition in Realistic Videos*. Tese de doutorado, School of Electrical Engineering and Computer Science, Faculty of Engineering, University of Ottawa. 74

Shi, F.; Laganiere, R. & Petriu, E. (2015). Gradient Boundary Histograms for Action Recognition. Em *WACV*. 15, 31, 59, 63, 66, 74, 75

Simonyan, K. & Zisserman, A. (2014). Two-stream Convolutional Networks for Action Recognition in Videos. Em *NIPS*. 3, 20, 27, 36, 68, 69, 74, 75

Simonyan, K. & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. Em *ICLR*. 21, 24

Sivic, J. & Zisserman, A. (2003). Video Google: A Text Retrieval Approach to Object Matching in Videos. Em *ICCV*. 3, 11, 19, 27

Smeulders, A. W. M.; Worring, M.; Santini, S.; Gupta, A. & Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.* 2

Song, S.; Lan, C.; Xing, J.; Zeng, W. & Liu, J. (2017). An end-to-end spatio-temporal attention model for human action recognition from skeleton data. Em *AAAI Conference on Artificial Intelligence*. 22

Soomro, K. & Zamir, A. R. (2014). *Action Recognition in Realistic Sports Videos*, capítulo 9, pp. 181--208. Springer International Publishing. 51

Soomro, K.; Zamir, A. R. & Shah, M. (2012). UCF101: A Dataset of 101 Human Action Classes From Videos in The Wild. Relatório técnico, CRCV-TR. xvi, xx, 7, 9, 30, 38, 39, 40, 54, 55, 58, 68, 72, 73, 75, 107, 108, 109

Srivastava, N.; Mansimov, E. & Salakhutdinov, R. (2015). Unsupervised Learning of Video Representations Using LSTMs. Em *ICML*. 74, 75

Stehling, R. O.; Nascimento, M. A. & Falcão, A. X. (2002). A compact and efficient image retrieval approach based on border/interior pixel classification. Em *CIKM*. 30

Sun, L.; Jia, K.; Yeung, D. Y. & Shi, B. E. (2015). Human Action Recognition Using Factorized Spatio-Temporal Convolutional Networks. Em *ICCV*. 74, 75

Sun, Q. & Liu, H. (2013). Learning spatio-temporal co-occurrence correlograms for efficient human action classification. Em *ICIP*. 8, 13, 20, 63, 68

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V. & Rabinovich, A. (2015). Going Deeper With Convolutions. Em *CVPR*. 21

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L. & Paluri, M. (2015). Learning Spatiotemporal Features With 3D Convolutional Networks. Em *ICCV*. 20, 74, 75

Varol, G.; Laptev, I. & Schmid, C. (2018). Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 21

Veeriah, V.; Zhuang, N. & Qi, G.-J. (2015). Differential recurrent neural networks for action recognition. Em *IEEE International Conference on Computer Vision (ICCV)*. 22

Vemulapalli, R.; Arrate, F. & Chellappa, R. (2014). Human action recognition by representing 3d skeletons as points in a lie group. Em *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 3, 18, 27

Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X. & Tang, X. (2017). Residual attention network for image classification. Em *CVPR*. 93

Wang, H.; Klaser, A.; Schmid, C. & Liu, C.-L. (2011). Action recognition by dense trajectories. Em *CVPR*. 2, 14, 15, 16, 27, 31, 59, 60, 62, 63, 67, 68

Wang, H. & Schmid, C. (2013). Action Recognition with Improved Trajectories. Em *ICCV*. 14, 16, 75

Wang, H.; Ullah, M. M.; Klaser, A.; Laptev, I. & Schmid, C. (2009). Evaluation of local spatio-temporal features for action recognition. Em *BMVC*. 60, 61, 63, 64, 68

Wang, J.; Liu, Z. & Wu, Y. (2014). *Human Action Recognition with Depth Cameras*. Springer Publishing Company, Incorporated. 3, 38

Wang, J.; Liu, Z.; Wu, Y. & Yuan, J. (2012). Mining actionlet ensemble for action recognition with depth cameras. Em *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 3, 17, 18

Wang, J. & Xu, Z. (2016). Spatio-temporal texture modelling for real-time crowd anomaly detection. *Computer Vision and Image Understanding*. 1

Wang, L.; Xiong, Y.; Wang, Z. & Qiao, Y. (2015). Towards Good Practices for Very Deep Two-Stream ConvNets. *CoRR*. xx, 21, 27, 35, 36, 37, 38, 40, 67, 68, 69, 70, 72, 73

Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X. & Van Gool, L. (2016a). Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. Em *ECCV*. xx, 3, 21, 23, 35, 37, 40, 67, 68, 69, 72, 73, 74, 75, 80

Wang, P.; Li, W.; Li, C. & Hou, Y. (2018). Action recognition based on joint trajectory maps with convolutional neural networks. *Knowledge-Based Systems*. 22, 23, 27

Wang, P.; Li, Z.; Hou, Y. & Li, W. (2016b). Action recognition based on joint trajectory maps using convolutional neural networks. Em *ACM International Conference on Multimedia (MM)*. 22, 27, 78, 80

Watanabe, T.; Ito, S. & Yokoi, K. (2008). Co-occurrence histograms of oriented gradients for pedestrian detection. Em *PSIVT*. 8, 13

Wiliem, A.; Madasu, V.; Boles, W. & Yarlagadda, P. (2012). A suspicious behaviour detection using a context space model for smart surveillance systems. *Comput. Vis. Image Underst.* 1

Woo, S.; Park, J.; Lee, J.-Y. & So Kweon, I. (2018). Cbam: Convolutional block attention module. Em *ECCV*. 93

Yang, X. & Tian, Y. L. (2012). Eigenjoints-based action recognition using naïve-bayes-nearest-neighbor. Em *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 3

Yang, Z.; Li, Y.; Yang, J. & Luo, J. (2018). Action recognition with spatio-temporal visual attention on skeleton image sequences. *IEEE Transactions on Circuits and Systems for Video Technology.* xvi, 3, 9, 22, 24, 26, 27, 41, 42, 45, 78, 80, 81, 82, 83, 84, 85

Zach, C.; Pock, T. & Bischof, H. (2007). A Duality Based Approach for Realtime TV-L1 Optical Flow. Em *Proceedings of the 29th DAGM Conference on Pattern Recognition.* 69

Zanfir, M.; Leordeanu, M. & Sminchisescu, C. (2013). The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. Em *IEEE International Conference on Computer Vision (ICCV).* 3

Zatsiorsky, V. M. (1998). *Kinematics of Human Motion.* Human Kinetics Publishers. 16

Zeiler, M. D. & Fergus, R. (2014). *Visualizing and Understanding Convolutional Networks*, pp. 818--833. Springer International Publishing. 21

Zhang, L.; Zhen, X. & Shao, L. (2012). High order co-occurrence of visualwords for action recognition. Em *ICIP.* 8, 13, 19, 63, 67, 68

Zhang, P.; Lan, C.; Xing, J.; Zeng, W.; Xue, J. & Zheng, N. (2017). View adaptive recurrent neural networks for high performance human action recognition from skeleton data. Em *IEEE International Conference on Computer Vision (ICCV).* 22

Zhu, W.; Hu, J.; Sun, G.; Cao, X. & Qiao, Y. (2016). A Key Volume Mining Deep Framework for Action Recognition. Em *CVPR.* 75

Zhu, Y. & Newsam, S. (2016). Depth2action: Exploring embedded depth for large-scale action recognition. Em *Computer Vision – ECCV 2016 Workshops.* 21

Zolfaghari, M.; Oliveira, G. L.; Sedaghat, N. & Brox, T. (2017). Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection. Em *International Conference on Computer Vision (ICCV).* 21

Zouba, N.; Bremond, F.; Thonnat, M. & Vu, V. T. (2007). Thinh. multi-sensors analysis for everyday activity monitoring. *SETIT, Tunisie.* xv, 5

Apêndice A

Activity Dataset Tables

In this appendix, we summarize the number of videos for each activity in the UCF101 dataset [Soomro et al., 2012] and the NTU RGB+D 60 dataset [Shahroudy et al., 2016].

Table A.1. Number of videos for each activity from the UCF101 dataset [Soomro et al., 2012].

		Split 1		Split 2		Split 3	
Activity	**#Total videos**	**Training**	**Test**	**Training**	**Test**	**Training**	**Test**
1: ApplyEyeMakeup	145	101	44	106	39	108	37
2: ApplyLipstick	114	82	32	85	29	81	33
3: Archery	145	104	41	104	41	107	38
4: BabyCrawling	132	97	35	97	35	93	39
5: BalanceBeam	108	77	31	79	29	77	31
6: BandMarching	155	112	43	110	45	112	43
7: BaseballPitch	150	107	43	111	39	110	40
8: Basketball	134	99	35	101	33	92	42
9: BasketballDunk	131	94	37	95	36	91	40
10: BenchPress	160	112	48	116	44	118	42
11: Biking	134	96	38	98	36	94	40
12: Billiards	150	110	40	104	46	108	42
13: BlowDryHair	131	93	38	91	40	97	34
14: BlowingCandles	109	76	33	79	30	81	28
15: BodyWeightSquats	112	82	30	80	32	82	30
16: Bowling	155	112	43	112	43	112	43
17: BoxingPunchingBag	163	114	49	122	41	118	45
18: BoxingSpeedBag	134	97	37	94	40	97	37
19: BreastStroke	101	73	28	73	28	72	29
20: BrushingTeeth	131	95	36	97	34	97	34
21: CleanAndJerk	112	79	33	79	33	82	30
22: CliffDiving	138	99	39	97	41	100	38
23: CricketBowling	139	103	36	100	39	97	42
24: CricketShot	167	118	49	118	49	124	43
25: CuttingInKitchen	110	77	33	79	31	82	28
26: Diving	150	105	45	106	44	112	38
27: Drumming	161	116	45	118	43	113	48
28: Fencing	111	77	34	78	33	83	28
29: FieldHockeyPenalty	126	86	40	89	37	96	30
30: FloorGymnastics	125	89	36	90	35	88	37
31: FrisbeeCatch	126	89	37	92	34	89	37
32: FrontCrawl	137	100	37	99	38	98	39
33: GolfSwing	139	100	39	106	33	99	40
34: Haircut	130	97	33	93	37	95	35
35: Hammering	140	107	33	99	41	98	42
36: HammerThrow	150	105	45	104	46	114	36
37: HandstandPushups	128	100	28	91	37	86	42
38: HandstandWalking	111	77	34	83	28	80	31
39: HeadMassage	147	106	41	107	40	106	41
40: HighJump	123	86	37	89	34	92	31
41: HorseRace	124	89	35	91	33	88	36
42: HorseRiding	164	115	49	120	44	120	44
43: HulaHoop	125	91	34	89	36	90	35
44: IceDancing	158	112	46	116	42	116	42
45: JavelinThrow	117	86	31	84	33	84	33
46: JugglingBalls	121	81	40	87	34	93	28
47: JumpingJack	123	86	37	87	36	92	31
48: JumpRope	144	106	38	105	39	101	43
49: Kayaking	141	105	36	105	36	95	46
50: Knitting	123	89	34	87	36	89	34

Table A.2. Number of videos for each activity from the UCF101 dataset [Soomro et al., 2012].

		Split 1		**Split 2**		**Split 3**	
Activity	**#Total videos**	**Training**	**Test**	**Training**	**Test**	**Training**	**Test**
51: LongJump	131	92	39	89	42	97	34
52: Lunges	127	90	37	88	39	94	33
53: MilitaryParade	125	92	33	89	36	88	37
54: Mixing	136	91	45	92	44	105	31
55: MoppingFloor	110	76	34	82	28	78	32
56: Nunchucks	132	97	35	92	40	95	37
57: ParallelBars	114	77	37	81	33	86	28
58: PizzaTossing	113	80	33	84	29	84	29
59: PlayingCello	164	120	44	119	45	117	47
60: PlayingDaf	151	110	41	110	41	106	45
61: PlayingDhol	164	115	49	116	48	123	41
62: PlayingFlute	155	107	48	113	42	113	42
63: PlayingGuitar	160	117	43	114	46	112	48
64: PlayingPiano	105	77	28	77	28	72	33
65: PlayingSitar	157	113	44	114	43	111	46
66: PlayingTabla	111	80	31	79	32	80	31
67: PlayingViolin	100	72	28	72	28	72	28
68: PoleVault	149	109	40	110	39	104	45
69: PommelHorse	123	88	35	89	34	88	35
70: PullUps	100	72	28	72	28	72	28
71: Punch	160	121	39	112	48	115	45
72: PushUps	102	72	30	74	28	74	28
73: Rafting	111	83	28	78	33	77	34
74: RockClimbingIndoor	144	103	41	105	39	107	37
75: RopeClimbing	119	85	34	86	33	84	35
76: Rowing	137	101	36	93	44	100	37
77: SalsaSpin	133	90	43	94	39	100	33
78: ShavingBeard	161	118	43	118	43	113	48
79: Shotput	144	98	46	102	42	109	35
80: SkateBoarding	120	88	32	85	35	86	34
81: Skiing	135	95	40	98	37	97	38
82: Skijet	100	72	28	72	28	72	28
83: SkyDiving	110	79	31	81	29	76	34
84: SoccerJuggling	147	108	39	108	39	104	43
85: SoccerPenalty	137	96	41	97	40	102	35
86: StillRings	112	80	32	82	30	81	31
87: SumoWrestling	116	82	34	83	33	83	33
88: Surfing	126	93	33	88	38	87	39
89: Swing	131	89	42	97	34	95	36
90: TableTennisShot	140	101	39	107	33	100	40
91: TaiChi	100	72	28	72	28	72	28
92: TennisSwing	166	117	49	118	48	121	45
93: ThrowDiscus	130	92	38	93	37	94	36
94: TrampolineJumping	119	87	32	84	35	85	34
95: Typing	136	93	43	90	46	105	31
96: UnevenBars	104	76	28	76	28	72	32
97: VolleyballSpiking	116	81	35	79	37	88	28
98: WalkingWithDog	123	87	36	88	35	89	34
99: WallPushups	130	95	35	94	36	88	42
100: WritingOnBoard	152	107	45	115	37	110	42
101: YoYo	128	92	36	93	35	92	36

Table A.3. Number of videos for each activity from the NTU RGB+D 60 dataset [Shahroudy et al., 2016].

		Cross-subject		**Cross-view**	
Activity	**#Total videos**	**Training**	**Test**	**Training**	**Test**
1: Drink	948	276	672	316	632
2: Eat meal	948	276	672	316	632
3: Brushing teeth	948	276	672	316	632
4: Brushing hair	948	276	672	316	632
5: Drop	948	276	672	316	632
6: Pickup	948	276	672	316	632
7: Throw	948	276	672	316	632
8: Sitting down	948	276	672	316	632
9: Standing up	948	276	672	316	632
10: Clapping	948	276	672	316	632
11: Reading	948	276	672	316	632
12: Writing	948	276	672	316	632
13: Tear up paper	948	276	672	316	632
14: Wear jacket	948	276	672	316	632
15: Take off jacket	948	276	672	316	632
16: Wear a shoe	948	276	672	316	632
17: Take off a shoe	948	276	672	316	632
18: Wear on glasses	948	276	672	316	632
19: Take off glasses	948	276	672	316	632
20: Put on a hat cap	948	276	672	316	632
21: Take off a hat cap	948	276	672	316	632
22: Cheer up	948	276	672	316	632
23: Hand waving	948	276	672	316	632
24: Kicking something	948	276	672	316	632
25: Put take out pocket	948	276	672	316	632
26: Hopping	948	276	672	316	632
27: Jump up	948	276	672	316	632
28: Make a phone call	948	276	672	316	632
29: Playing with phone	948	276	672	316	632
30: Typing on a keyboard	948	276	672	316	632
31: Pointing to something	948	276	672	316	632
32: Taking a selfie	948	276	672	316	632
33: Check time	948	276	672	316	632
34: Rub two hands together	948	276	672	316	632
35: Nod head bow	948	276	672	316	632
36: Shake head	948	276	672	316	632
37: Wipe face	948	276	672	316	632
38: Salute	948	276	672	316	632
39: Put the palms together	948	276	672	316	632
40: Cross hands in front	948	276	672	316	632
41: Sneeze cough	948	276	672	316	632
42: Staggering	948	276	672	316	632
43: Falling	948	276	672	316	632
44: Touch head	948	276	672	316	632
45: Touch chest	948	276	672	316	632
46: Touch back	948	276	672	316	632
47: Touch neck	948	276	672	316	632
48: Nausea	948	276	672	316	632
49: Use a fan	948	276	672	316	632
50: Punching slapping other person	948	276	672	316	632
51: Kicking other person	948	276	672	316	632
52: Pushing other person	948	276	672	316	632
53: Pat on back of other person	948	276	672	316	632
54: Point finger at the other person	948	276	672	316	632
55: Hugging other person	948	276	672	316	632
56: Giving something to other person	948	276	672	316	632
57: Touch pocket	948	276	672	316	632
58: Handshaking	948	276	672	316	632
59: Walking towards	948	276	672	316	632
60: Walking apart	948	276	672	316	632

Apêndice B

Haralick Textural Features

In this appendix, we detail the definitions of the Haralick features proposed by Haralick et al. [1973], which can be extracted from each of the gray level co-occurrence matrices. The following equations define these features.

Notation:

$p(i,j)$ $(i,j)th$ entry in a normalized gray level co-occurrence matrix, $= P(i,j)/R$.

$p_x(i)$ ith entry in the marginal probability matrix generated by summing up the rows of $p(i,j)$, $= \Sigma_{j=1}^{N_g} P(i,j)$.

N_g Number of distinct gray levels in the quantized image.

$\sum\limits_i$ **and** $\sum\limits_j$ means $\sum\limits_{i=1}^{N_g}$ and $\sum\limits_{j=1}^{N_g}$, respectively.

$$p_y(j) \;= \sum_{i=1}^{N_g} p(i,j).$$

$$p_{x+y}(k) \;=\; \sum_{\substack{i=1\\ i+j=k}}^{N_g} \sum_{j=1}^{N_g} p(i,j), \qquad k = 2, 3, \cdots, 2N_g.$$

$$p_{x-y}(k) \;=\; \sum_{\substack{i=1\\ |i-j|=k}}^{N_g} \sum_{j=1}^{N_g} p(i,j), \qquad k = 0, 1, \cdots, N_g - 1.$$

Textural Features:

1. Angular Second Moment:
   $f_1 = \sum\limits_i \sum\limits_j \{p(i,j)\}^2$
2. Contrast:
   $$f_2 = \sum_{n=0}^{N_{g-1}} n^2 \left\{ \sum_{\substack{i=1\\ |i-j|=n}}^{N_g} \sum_{j=1}^{N_g} p(i,j) \right\}$$

3. Correlation:

   $f_3 = \frac{\sum\limits_i \sum\limits_j (ij)p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}$

   where $\mu_x, \mu_y, \sigma_x and \sigma_y$ are the means and the standard deviations of p_x and p_y.

4. Sum of Squares: Variance

   $f_4 = \sum\limits_i \sum\limits_j (i - \mu)^2 p(i,j)$

5. Inverse Different Moment:

   $f_5 = \sum\limits_i \sum\limits_j \frac{1}{1+(i-j)^2} p(i,j)$

6. Sum Average:

   $f_6 = \sum\limits_{i=2}^{2N_g} i p_{x+y}(i)$

7. Sum Variance:

   $f_7 = \sum\limits_{i=2}^{2N_g} (i - f_8)^2 p_{x+y}(i)$

8. Sum Entropy:

   $f_8 = -\sum\limits_{i=2}^{2N_g} p_{x+y}(i) \log\{p_{x+y}(i)\}$

   since some of the probabilities may be zero, and log(0) is not defined, it is recommended that the term $\log(p+\varepsilon)$ (ε an arbitrarily small positive constant) be used in place of $\log(p)$ in entropy computations.

9. Entropy:

   $f_9 = -\sum\limits_i \sum\limits_j p(i,j) \log(p(i,j))$

10. Difference Variance:

    f_{10} = variance of p_{x-y}

11. Difference Entropy:

    $f_{11} = -\sum\limits_{i=0}^{N_{g-1}} p_{x-y}(i) \log\{p_{x-y}(i)\}$

12. , 13. Information Measures of Correlation:

    $f_{12} = \frac{HXY - HXY1}{max\{HX, HY\}}$

    $f_{13} = (1 - \exp[-2.0(HXY2 - HXY)])^{1/2}$

    $HXY = -\sum\limits_i \sum\limits_j p(i,j) \log(p(i,j))$

    where HX and HY are entropies of p_x and p_y, and

    $HXY1 = -\sum\limits_i \sum\limits_j p(i,j) \log\{p_x(i) p_y(i)\}$

    $HXY2 = -\sum\limits_i \sum\limits_j p_x(i) p_y(j) \log\{p_x(i) p_y(i)\}$

13. Maximal Correlation Coefficient:

    f_{14} = (Second largest eigenvalue of $Q)^{1/2}$

    where $Q(i,j) = \sum\limits_k \frac{p(i,k)p(j,k)}{p_x(i)p_y(k)}$