FASTENSOR: A TENSOR FRAMEWORK FOR SPATIOTEMPORAL DESCRIPTION

VIRGÍNIA FERNANDES MOTA

FASTENSOR: A TENSOR FRAMEWORK FOR SPATIOTEMPORAL DESCRIPTION

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

Orientador: Prof. Dr. Arnaldo de Albuquerque Araújo
Coorientador: Prof. Dr. Jefersson Alex dos Santos

Belo Horizonte
Dezembro de 2018

VIRGÍNIA FERNANDES MOTA

FASTENSOR: A TENSOR FRAMEWORK FOR SPATIOTEMPORAL DESCRIPTION

Thesis presented to the Graduate Program in Ciência da Computação of the Universidade Federal de Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Ciência da Computação.

Advisor: Prof. Dr. Arnaldo de Albuquerque Araújo
Co-Advisor: Prof. Dr. Jefersson Alex dos Santos

Belo Horizonte
December 2018

Ficha catalográfica elaborada pela Biblioteca do ICEx - UFMG

Mota, Virgínia Fernandes.

M917f FASTensor: A tensor framework for spatiotemporal description / Virgínia Fernandes Mota — Belo Horizonte, 2018.
xxii, 96 p.: il.; 29 cm.

Tese (doutorado) - Universidade Federal de Minas Gerais – Departamento de Ciência da Computação.

Orientador: Arnaldo de Albuquerque Araújo
Coorientador: Jefersson Alex dos Santos

1. Computação – Teses. 2. Processamento de imagens. 3. Sistemas Multimídia. 4. Representação do conhecimento (Teoria da informação). I. Orientador. II. Coorientador. III. Título.

CDU 519.6*84(043)

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

FOLHA DE APROVAÇÃO

FASTensor: A tensor framework for spatiotemporal description

VIRGÍNIA FERNANDES MOTA

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. ARNALDO DE ALBUQUERQUE ARAÚJO - Orientador
Departamento de Ciência da Computação - UFMG

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

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

PROF. GABRIEL DE MORAIS COUTINHO
Departamento de Ciência da Computação - UFMG

PROF. SILVIO JAMIL FERZOLI GUIMARÃES
Departamento de Ciência da Computação - PUC/MG

Hélio Pedrini
PROF. HÉLIO PEDRINI
Instituto de Computação - UNICAMP

Belo Horizonte, 17 de dezembro de 2018.

Acknowledgments

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

First of all, I would like to start with a simple "thank you for being there". The Ph.D. process is not easy for anyone, but sometimes our lives play tricks on us. With everything that happened to me in the past years, I was at the bottom of my life, and you were there to throw me a rope and say "Climb up, miss! We are waiting for you".

I am deeply grateful to my family, their inspiration and encouragement stimulated me to pursue a Ph.D. My mother and grandmother, Natália and Anastácia, are the professors of my life, and my godfather, André, who was always there for me.

This thesis would not have been possible without continuous support and guidance from my conscientious supervisors from DCC/UFMG, Professor Arnaldo de Albuquerque Araújo and, recently, Professor Jefersson Alex dos Santos. Thank you for the patience, knowledge, encouragement.

I would like to thank the Brazilian Funding Agencies CAPES and CNPq for supporting me during my graduate school with scholarships. My thanks to NVIDIA for their support with GPUs.

For my friends, it is hard to mention one by one, but I will try. Tassio Knop, Camila Laranjeira (the love of my life), Hugo Oliveira, Alan Deivite, Julliana Gomes, Priscila Aleixo, Bianca Portes, Lucas Lattari, Sandra Avila, Alberto Pimentel you were the ones that took my hand and said: "You can do it". I love you for that. Jéssica, Alex, Mabel, Aninha, Glauber, Cícero, Juliana, Priscila, Bela, Kelly, I want to mention everyone, but it is not easy. So, just accept that I love all of you. NPDI, PATREO and SSIG, all those labs that I was part of, thanks for the support.

For my friends and colleagues from COLTEC, João Montandon, Humberto Honda, Márcio Fantini, Leandro Maia, Luciano Almeida and so on. Thanks for the patience, help, lunches and coffee.

Finally, besides the aforementioned people, I would like to thank all of those who have helped me complete my thesis no matter how.

“Inspiração é quando a gente não sabe de onde a ideia vem.
Na ciência é o contrário: é preciso explicar
o caminho que se tomou para chegar à ideia.
É esse caminho que tem o nome de método.
Seguindo o mesmo caminho,
qualquer outro cientista poderá chegar à mesma ideia.”
(Rubem Alves)

Abstract

Spatiotemporal representation is a research field with application in various areas such as video indexing, surveillance, human-computer interfaces, among others. Big Data problems in large databases are now being treated with Deep Learning tools, however we still have room for improvement in low-level description. Moreover, we still have problems that involve small data in which data augmentation and other techniques are not enough. Our main contribution is the development of a multipurpose framework for spatiotemporal representation using orientation tensors: Features As Spatiotemporal Tensors (FASTensor). This framework can be used in videos or multitemporal images. The first step of the proposed method is the low-level feature vector extraction. Then, the orientation tensor created from each feature vector will be accumulated for each image/frame. With the orientation tensor, we can capture not only what happens in this scene, but how we begin to deform the ellipsoid created from the accumulation so that it carries the whole tendency of the feature used. To validate our descriptors, we use three different applications: Human Action Recognition, Video Pornography Classification and Melanoma Cancer Cell classification, to which we contribute with a new dataset. The Melanoma Cancer Cell dataset is a small data that can not be artificially augmented due the difficulty of extraction and the nature of motion. Our experiments for this problem can be used in other cancer cell treatment analysis. The evaluation of our tensor framework consists in a classification task of these applications using an SVM classifier. In summary, our hypothesis is that orientation tensors can be used as compact spatiotemporal representations, enabling dimension reduction and invariance. Our experiments and evidences contribute to it, as the results were competitive, while also being computationally fast and simple to implement.

Resumo

A representação da informação espaço-temporal é um campo de pesquisa com aplicações em diversas áreas, como indexação de vídeos, vigilância, interfaces homem-computador, para citar alguns exemplos. Problemas de grandes massas de dados (*Big Data*) agora estão sendo tratados com ferramentas de *Deep Learning*, no entanto, ainda temos espaço para melhorias na descrição de baixo nível. Além disso, ainda temos problemas que envolvem pequena quantidade de dados nos quais o aumento de dados (*data augmentation*) e outras técnicas não são suficientes. Nossa principal contribuição é o desenvolvimento de um arcabouço para representação espaço-temporal usando tensores de orientação: *Features As Spatiotemporal Tensors* (FASTensor). Essa estrutura pode ser usada em vídeos ou imagens multitemporais. A primeira etapa do método proposto é a extração de vetores de características de baixo nível. Em seguida, o tensor de orientação criado a partir de cada vetor de características será acumulado para cada imagem/quadro. Com o tensor de orientação, podemos capturar não apenas as informações do vetor de características, mas como também toda a tendência da característica usada. Para validar nossos descritores, usamos três aplicações diferentes: Reconhecimento de Ações Humanas, Classificação Vídeos Pornográficos e Classificação de Células de Melanoma, para o qual contribuímos com uma nova base de imagens multitemporais. A base de dados Melanoma Cancer Cells é um conjunto pequeno de dados que não pode ser aumentado devido à dificuldade de extração e à natureza do movimento. Nossos experimentos para este problema podem ser usadas em outras análises de tratamento de células cancerígenas. A avaliação de nosso *framework* consiste em uma tarefa de classificação dessas aplicações usando um classificador SVM. Em resumo, nossa hipótese é de que os tensores de orientação podem ser usados como representação espaço-temporal compacta, possibilitando redução de dimensão e invariância, de acordo com a característica usada para criá-los. Nossos experimentos e provas contribuem para isso, já que os resultados foram competitivos, além de serem rápidos e simples de implementar, computacionalmente.

List of Figures

List of Tables

Contents

Acronyms

ACORDE Adult Content Recognition with Deep Neural Networks. 25

BoF Bag-of-Features. 4, 5, 21

BossaNova Bag Of Statistical Sampling Analysis. 22, 60

BoW Bag-of-Word. 21

CCA Canonical Correlation Analysis. 22

CCD charge-coupled device. 71

CNN Convolutional Neural Networks. 24, 25, 45, 60

CP CANDECOMP/PARAFAC. 23

DCS Divergence-Curl-Shear. 57

DFT Discrete Fourier Transform. 23

DL Deep Learning. 4, 21, 58

DNN Deep Neural Networks. 25, 60

FASTensor Features as Spatiotemporal Tensor. 45, 47, 59, 69

FBS Fetal Bovine Serum. 71

fps frames per second. 8

GAN Generative Adversarial Networks. 26

HOF Histrogram of Optical Flow. 5, 21, 45, 63, 73

HOG Histogram of Gradients. 5, 21, 43, 45, 63, 73

HOSVD High Order Single Value Decomposition. 23

MAP Mean Average Precision. 63

MCC Melanoma Cancer Cell dataset. 11, 12, 15

OF Optical Flow. 5, 18

RANSAC RANdom SAmple Consensus. 55

RNN Recurrent Neural Networks. 25

RPMI Roswell Park Memorial Institute. 71

SIFT Scale-Invariant Feature Transform. 5

SOM Self-Organizing Map (SOM). 24

STIP Spatio Temporal Interest Points. 5

SURF Speeded Up Robust Feature. 5, 22

SVM Support Vector Machines. 9, 25, 45, 47, 61, 69

TCCA Tensor Canonical Correlation Analysis. 22

TRoF Temporal Robust Features. 22, 60

VLAD Vector of Locally Aggregated Descriptor. 57

VLAT Vector of Locally Aggregated Tensor. 57

Chapter 1

Introduction

Spatiotemporal data usually contains the states of an object, an event or a position in space over a period of time. This data can be created from videos or multitemporal images (sequences of images that are combined depending on the purpose). An event in a spatiotemporal dataset describes a spatial and temporal phenomenon that may happen at a certain time and location.

In order to learn useful information regarding these events, computational systems generally use combinations of different features representing visual elements from the scene, such as color, texture, salient points, apparent motion, trajectories, *etc.* Those visual patterns provide information on the two-dimensional and/or three-dimensional structure of the scene, shape and trajectory of objects and the activity that is going on. Therefore, this visual information of still and moving images is the key for tasks such as:

- Video compression [Wiegand and Sullivan, 2007; Sze et al., 2014]: a task of reducing the size of video recordings by eliminating redundant and non-functional data from the original video file according to temporal coherence;

- Object tracking [Lan et al., 2018; Lan et al., 2018]: the process of locating a moving object (or multiple objects) over time in videos or multitemporal images. Object tracking can be used in other computer vision tasks as pre-processing or the main problem, such as in human-computer interface or video segmentation;

- Video segmentation [Grundmann et al., 2010; Souza et al., 2014]: refers to analyzing video frames and segmenting them into regions of interest. Typical examples are segment background over foreground, segment faces, segment moving objects, to name a few;

- Video surveillance [Prates and Schwartz, 2018]: a research field to interpret monitoring data, with goals such as face recognition, human pose estimation, suspicious behavior and anomalies;

- Video and multitemporal image classification [Sivic and Zisserman, 2003; Almeida et al., 2016]: a classification is a division or category in a system which divides things into groups or types. Video and multitemporal image classification aims to classify videos/multitemporal images into categories relating to their visual content;

- Cell shape classification [Kriegel et al., 2017]: cell shapes tend to change their behavior through time depending on the applied treatment. Cell shape classification tries to analyze cellular shapes in order to categorize groups of cells.

All those tasks that work with moving pictures need to be represented using not only spatial characteristics, but spatiotemporal description. It is a challenging application as we have continuous and discrete changes on the scene being influenced locally and globally, both in time and space. Let us take as an example two actions from the video dataset KTH [Schuldt et al., 2004]: Jogging and Walking. For both, we have a person doing the action in a homogeneous background that involves moving their feet. However, those actions are slightly different according to their velocities. So, we have to take into account the shape of the movement, the coherence through time, the velocity between frames. This exemplifies the challenges of extracting semantic information from elements in a scene that do not intrinsically possess semantic meaning, but instead are encoded as sequential numerical matrices with temporal variations.

In this thesis, we address the spatiotemporal feature representation problem applied to video and multitemporal image classification. Many works tackle this problem following three steps: handcrafted feature extraction, descriptor creation, and classification. Figure 1.1 shows the standard framework with the main steps of video classification. Regarding this framework, we are mainly interested in the first two steps: *feature extraction* and *descriptor creation.*

We classify the existing methods based on the type of features: Shallow and Deep Learning (DL) methods. Shallow methods are categorized into the following classes: *1.* Low-level approaches with handcrafted features; *2.* Bag-of-Features (BoF) representations or middle-level approaches. Deep Learning-based approaches share similar procedures: patch sampling, feature description/learning and classification [Hu et al., 2015]. Nowadays, image and video classification problems in large databases are being treated with Deep Learning tools [Karpathy et al., 2014; Wehrmann et al., 2018].

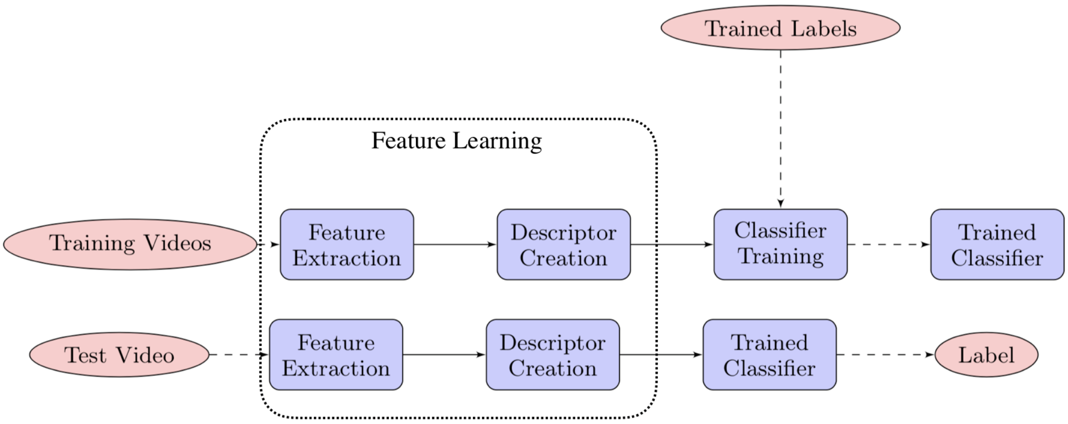


Figure 1.1: Standard framework for video classification with training and test example. First, a set of video descriptors is used to train a classifier. Then, a different set of video descriptors is used for the test and a label is given to them. This label indicates which action the video represents. The main difference between shallow and deep methods is in feature extraction and descriptor creation, deep learning methods use feature learning instead of those two different steps (Image done by the author).

However, deep architecture models suffer from over-fitting problems when there is a small amount of training data. There are methods to overcome this problem, such as data augmentation, transfer learning, data generation, among others [Pasupa and Sunhem, 2016; Sani et al., 2017]. But coherent approaches for moving pictures are still in their infancy as well as adding the temporal information on a deep architecture [Wang et al., 2018]. Hence, this is still an open problem in literature.

For the shallow methods, the handcrafted feature extraction starts by a preliminary dimension reduction since some point based motion indicator, usually intensity gradient, is coded in a compact form. Feature examples include Histogram of Gradients (HOG), Histrogram of Optical Flow (HOF), basis projections, and other Optical Flow (OF) based features [Dalal et al., 2006]. In most works of the literature, these features are associated with Scale-Invariant Feature Transform (SIFT) [Lowe, 1999], Speeded Up Robust Feature (SURF) [Bay et al., 2008] or Spatio Temporal Interest Points (STIP) [Laptev and Lindeberg, 2003] descriptors.

The description creation step uses the extracted features to provide the video signature, using a single type or a combination of features. The most used method for shallow methods is the canonical BoF [Sivic and Zisserman, 2003]. We discuss other methods to create the video signature. Using the idea of coding features into orientation tensors, we are able to aggregate them in order to represent the temporal evolution.

Different from the shallow methods, Deep Learning-based approaches do not usually work with handcrafted feature extraction. A deep feature is the consistent response of a unit within a hierarchical model to an input, where this response contributes to the model decision. A feature could be considered deeper than another depending on where the unit is positioned alongside the hierarchical structure of the model [Karpathy et al., 2014].

In this thesis, we work with handcrafted and deep feature extraction, thus the classification method follows a shallow approach. The video classification step is used to evaluate the descriptors created. We work with three spatiotemporal representation tasks: Human Action Recognition, Video Pornography classification and Cancer Cell classification.

1.1 Motivation

Orientation tensors are robust mathematical tools that perform information aggregation, which allows them to be applied to multivariate data [Mordohai and Medioni, 2007]. Temporal pixel-value variations in a scene (*i.e.*, occlusions, illumination settings, movement, *etc.*) are known to have multivariate components, which suggests that orientation tensors are an appropriate tool for the problem of aggregation and dimensionality reduction of visual data. These visual elements will be henceforth known as spatiotemporal features.

Those features with n dimensions can be coded into orientation tensors taking the form of a $n \times n$ symmetric matrix [Johansson et al., 2002]. This matrix will carry the covariance information from features. Here is interesting to note that we work only with vector and matrices calculation, which is easy to compute. Moreover, being a symmetric matrix, we can save only the triangular matrix above/below the main diagonal, hence, saving disk space. Moreover, by aggregating features through time, we can add temporal information maintaining the orientation tensor properties.

Thus, the main advantages are: orientation tensors are robust mathematical tools, easy to compute and save disk space.

Therefore, our main motivation to use tensor representation is **orientation tensors can capture uncertainties of visual elements from the scene, carrying more information than a feature vector**. Moreover, a good spatiotemporal representation can be used in many computer vision applications, as we saw above.

We justify the choice for using tensors in Chapters 3 and 5, in which we describe the basic mathematical framework needed to understand our proposed method, and in the following chapters we explore three spatiotemporal representation tasks: Human Action Recognition (Chapter 6), Video Pornography classification (Chapter 7) and Cancer Cell classification (Chapter 8), to which we contribute with a new dataset. Segments from this dataset (multitemporal images) are very difficult to capture and there are no other similar datasets, to the best of our knowledge. Therefore, deep learning techniques are not suitable for this dataset. This opens the discussion on how to work with datasets with a small number of samples.

1.2 Research Challenges

The problem addressed in this thesis is the study and development of a spatiotemporal feature representation capable of representing the patterns in moving images (video or multitemporal images), based only on gray-level variations due to temporal changes.

We assume that the descriptor may follow the assumptions bellow:

- In all image sequences, there is at least one important and representative motion: The representation must be discriminative. If we do not have a motion that outstands in a scene, we can not describe it using orientation tensors as they become isotropic (that is, no main direction or information is represented).

- In all image sequences, there is at least one important representative pattern: Different from the motion, a pattern is a global information from the scene. For the same reason, if we do not have a pattern that outstands in a scene, we can not describe it using orientation tensors as they become isotropic.

- The descriptor must be easy to compute: In this thesis, we want to study low complexity representations for video and multitemporal images, mainly exploring robust mathematical tools, such as orientation tensors.

Given the fact that tensors are able to capture uncertainties from temporal changes, we want to show how a spatiotemporal descriptor based on orientation tensors can be used from simple movements, such as cell motions, to complex movements such as human actions.

We explore three spatiotemporal tasks: Human Action Recognition, Video Pornography classification and Cancer Cell classification. Human Action Recognition aims to classify body movements into a set of known categories. Video Pornography classification, a more semantic task, can be defined as classification of videos in porn and non-porn classes. Finally, Cancer Cell classification aims to categorize cells into two classes: treated and non-treated.

Therefore, we will handle three main computational challenges: to work with *spatiotemporal semantic encoding*, the *compact representation* and the problem of learning from *small amount of data.*

Spatiotemporal semantic encoding The definition of a video according the Cambridge Dictionary[1] is: "a series of recorded images which are shown on television or viewed on a screen". In general, the recorded images are called frames and the video has a fixed number of frames per second (fps).

On the other hand, multitemporal images have a broader definition. They are also a series of recorded images, however, we do not have a fixed fps rate. Images can be captured in different times and then put together in order to form a video. This poses a great challenge, as the temporal coherence can be lost in capturing images. Moreover, for video or multitemporal images we work with their visual content, so we need to analyze their intrinsic semantic. In this context, semantic means what can be interpreted from the features extracted from the images or videos.

Why spatiotemporal tasks are so challenging? In general, video datasets have many intra and inter-class variations. For example, in Hollywood2 dataset [Marszałek et al., 2009] we have actions like *getting out of a car* and *answer phone.* A descriptor has to be able to represent those different actions. On the other hand, we also have many similar situations, such as Jogging and Walking from KTH dataset [Schuldt et al., 2004]. When we handle a more semantic context, such Pornography classification [Avila et al., 2013], the nuances are subtle. Though it certainly relates to nudity, pornography is a different concept: many activities which involve a high degree of body exposure (swimming,boxing, sunbathing, *etc.*) have nothing to do with it. We also have to take into account the recording conditions, the resolution, frames per second (mainly when we work with multitemporal images). All this could influence in the discriminality of the spatiotemporal representation.

Furthermore, multimedia data is always increasing. In 2005, Roger Magoulas from O'Reilly media coined the term Big Data to refer to a wide range of large data sets almost impossible to manage and process using traditional data management tools - due to their size, but also to their complexity[2]. Since then, more and more effort has been made in order to better describe and analyze those amount of data. Here is when Deep Learning gained terrain on literature.

[1] https://dictionary.cambridge.org/

[2] http://strata.oreilly.com/2010/01/roger-magoulas-on-big-data.html

Nowadays, image and video classification problems in large databases are being increasingly treated with Deep Learning tools [Karpathy et al., 2014; Wehrmann et al., 2018].

However, coherent approaches for moving pictures are still in their infancy as well as adding the temporal information to a deep architecture [Wang et al., 2018]. This is still an open a challenging problem in computer vision. We can add this temporality using handcrafted and/or deep features, that is, features extracted from shallow and/or DL approaches.

In this thesis, we work with both types of data. In Human Action Recognition (Chapter 6) and Pornography classification (Chapter 7) we use videos and for Cancer Cell classification (Chapter 8) we use multitemporal images.

Compact Descriptors The advantages to work with compact descriptors are to reduce time and space complexity, to create scalable methods and to provide broader description methods, that is, compact descriptors can be used from classification to compression [Duan et al., 2017].

Moreover, a high computational cost to create the representations could make it extremely hard to use in some real time applications [Caetano et al., 2014].

Shallow approaches have been working on compact descriptors of images and videos for a long time [Laptev et al., 2007]. The majority of works in literature used Support Vector Machines (SVM) as classifier. Yet, SVM does not work well with high dimensionality. That is why there was a great effort to work with dimension reduction and compact description in shallow approaches [Douze et al., 2010].

In deep learning, we can induce this throughout sparsity with regularization. However, it is not intrinsic to deep learning approaches. In general, the feature is dense and can not be compact [LeCun et al., 2015].

The idea in this thesis is to have an intrinsic compact representation, which is a computational challenge: we want to have a good balance between descriptor size, computer complexity and effectiveness.

Robustness in Small Data Scenarios It is a fact that we are in the era of Big Data. However, we can not neglect applications in which it is not possible to obtain large amounts of data. Drawing reliable conclusions from small datasets, like those from clinical trials for rare diseases or in studies of endangered species, remains one of the trickiest obstacles in machine learning [Altae-Tran et al., 2017]. Thus, let us take a look on this problem: Small Data.

Small Data as a topic of research is not yet defined. We can observe works that use terms like *small data*, *scarce data* to refer to the problem. We also have specific methods to work with small amounts of data, as one-shot learning [Fei-fei, 2006] and zero-shot learning [Larochelle et al., 2008]. One-shot learning aims to learn information about object categories from one, or only a few, training images. Zero-shot learning is used to construct recognition models for unseen target classes that were not labeled for training.

To sketch the emergence of Small Data in counterpoint to Big Data, we searched on Google Scholar[3] for the terms: Big Data + Classification, Small Data + Classification, Scarce Data + Classification, One-shot learning and Zero-shot learning. It should be noted that there are other phrases that might refer to the same concept, such as "large datasets" for Big Data or "little datasets" for Small Data. Moreover, we did not manually analyze each work found, since the focus of this discussion is to analyze the popularity of the terms in literature and show the difference between the number of works that address the problem of Big Data compared to Small Data approaches. Figure 1.2 shows the number of works from 2012 until 2018.

It is easy to see how Big Data has more works in the literature than any terms related to small data. Moreover, we see how learning from low data still is incipient. Thus, we need to put effort in this area to know how to represent, analyze and learn from Small Data with robustness.

Bradley Arsenault, Founder and CEO of Electric Brains[4], affirmed in a text about how Small Data is the future of Artificial Intelligence[5]: "For every dataset with one billion entries, there are 1,000 datasets with one million entries, and 1,000,000 datasets with only one thousand entries. So once the low-hanging fruit has been exhausted, the only possible way to move forward will be to climb the tree and build systems which can work with less and less data."

Therefore, in order to discuss this matter, we present a small dataset that can not be artificially augmented and transfer learning is not available: Melanoma Cancer Cell dataset (MCC) (Chapter 8).

Thus, in this thesis, we want to answer the following research questions. Table 1.1 resumes the research questions for each explored application. The marked cells represents in which application the discussion appears.

1. Can orientation tensors be used as compact spatiotemporal representations?

[3] http://scholar.google.com

[4] https://www.electricbrain.io

[5] https://towardsdatascience.com/why-small-data-is-the-future-of-ai-cb7d705b7f0a

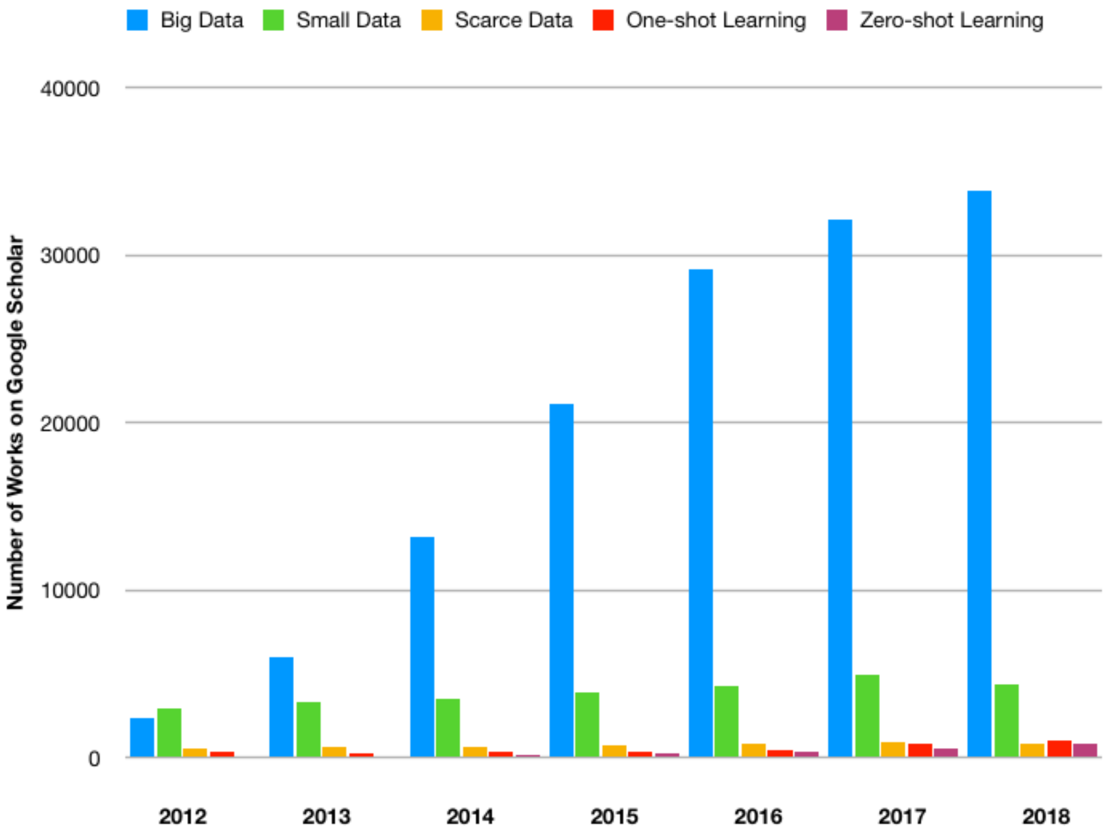


Figure 1.2: Popularity analysis of terms related to Big Data and Small Data searched from 2012 until 2018 on Google Scholar (Image made by the author).

2. Can the same representation be used in different classification tasks?
3. Can the same representation be used for video and for multitemporal images?
4. How do raw features from different natures (shallow and deep) behave in video description?
5. Is the orientation tensor suitable for supervised classification with small datasets?

Table 1.1: Research questions for each explored application. The marked cells represents in which application the discussion appears.

Question		Human Action Recognition	Pornography Classification	Cancer Cell Classification
1	Compact spatiotemporal description	▨	▨	▨
2	Different tasks	▨	▨	▨
3	Video vs Multitemporal images			▨
4	Shallow and Deep Features		▨	
5	Small Data			▨

1.3 Hypotheses

The hypotheses of this thesis are:

Hypothesis 1 (H.1) *Orientation tensors can be used as compact spatiotemporal representations, enabling dimensionality reduction and invariance, according to the feature used to built them.* These mathematical entities can capture the local, global orientation and dispersion of multivariate data, such as videos.

Hypothesis 2 (H.2) *The framework created (FASTensor) can be used in different spatiotemporal tasks, as Human Action Recognition, Video Pornography Classification and Cancer Cell Classification.* Each one of these tasks, even if they are spatiotemporal kind, have different elements. In Human Action Recognition, the same scene can represent several different actions (*e.g.*, Sit down while picking up the phone). In Video Pornography classification, the problem is more semantic than a mere movement. Finally, for Cancer Cell classification, the nature of motion is completely different from those human-based tasks.

Hypothesis 3 (H.3) *The framework created (FASTensor) can improve the accuracy of spatiotemporal tasks using handcrafted and deep features.* Automatically learning features at multiple levels of abstraction allows a system to learn complex functions mapping the input to the output directly from data, without depending so much on handcrafted features. However, each type of feature carries its importance and handcrafted ones are fast to compute. Thus, we need to study how the shallow and the deep features behave in our framework and if the FASTensor can improve results in both types of feature.

1.4 Contributions

The main contribution of this work is the development of a novel spatiotemporal description framework (Features As Spatiotemporal Tensors – *FASTensor*) using orientation tensors, enabling dimensionality reduction and invariance according to the feature. In order to evaluate the framework in other distinct and challenging scenarios than the traditional computer vision tasks, we also present a new open labeled dataset for melanoma cancer cell classification. It is a small dataset, called MCC, that cannot be artificially augmented due the inherent difficulties in the acquisition process and its particular nature. Furthermore, there are no similar open datasets in the literature, to the best of our knowledge. This opens the discussion on how we can we learn with small datasets. Our proposed method and experiments show that the method can be used in other cancer cell treatment analysis.

This research produced the following published papers as contribution to the literature in spatiotemporal representation:

- Journals
  - *Under Review*: Mota, V. F.; Oliveira, H.; Scalzo, S.; Dittz D.; Santos, R. J.; dos Santos, J. A.; Araújo, A. A. From video pornography to cancer cells: a tensor framework for spatiotemporal description. Multimedia Tools and Applications, 2018.
  - Mota, V. F.; Perez, E. A.; Maciel, L.; Vieira, M. B.; Gosselin, P. A tensor motion descriptor based on histogram of gradients and optical flow. Pattern Recognition Letters, 2013, v. 39, p. 85-91.
- Book Chapters
  - Mota, V. F.; Vieira, M. B.; Araújo, A. A. . Busca por Imagens e Vídeos com base no Conteúdo Visual: Uma Introdução. Anais da VII Escola Regional de Informática de Minas Gerais - ERI-MG, Juiz de Fora, 2012, v. 1, p. 1-24.
- Conferences
  - Mota, V. F.; Dias, G. D.; Santos, W. T.; Vieira, M. B.; Araújo, A. A. Tensor clustering for human action recognition. Workshop of Works in Progress/ Conference on Graphics, Patterns and Images (SIBGRAPI), 2015.
  - Mota, V. F.; Souza, J.; Araújo, A. A.; Vieira, M. B. Combining orientation tensors for human action recognition. Conference on Graphics, Patterns and Images (SIBGRAPI), 2013, p. 328-333.

This thesis also contributed to:

- Journals
  - Maia, H. A.; Figueiredo, A. M. O.; Oliveira, F. L. M.; Mota, V. F.; Vieira, M. B. A video tensor self-descriptor based on variable size block matching. Journal of Mobile Multimedia, 2015, v. 11, p. 90-102.
  - Oliveira, F. L. M.; Maia, H. A.; Mota, V.F.; Vieira, M. B.; Araújo, A. A. A variable size block matching based descriptor for human action recognition. Journal of Communication and Information Systems, 2015, v. 30, p. 90-99.

- Conferences
  - Figueiredo, A. M. O.; Caniato, M.; Mota, V. F.; Silva, R. L. S.; Vieira, M. B. A video self-descriptor based on sparse trajectory clustering. International Conference in Computer Science and its Applications (ICCSA), 2016, v. 9787, p. 571-583.
  - Lenzoni, C. S.; de Paula, G.; de Feiras, L. W.; Mota, V. F.; Pires, L.; Fernandes, N. M. S. Ferramenta de assistência médica para o estudo de declínio cognitivo em pacientes com doença renal crônica. Workshop of Works in Progress/Conference on Graphics, Patterns and Images (SIBGRAPI), 2016.
  - Figueiredo, A. M. O.; Maia, H. A.; Oliveira, F. L. M.; Mota, V. F.; Vieira, M. B. A Video tensor self-descriptor based on block matching. International Conference in Computer Science and its Applications (ICCSA), 2014, v. 8584, p. 401-414.
  - Oliveira, F. L.; Maia, H. A.; Mota, V. F.; Vieira, M. B.; Araújo, A. A. Video tensor self-descriptor based on variable size block matching. Workshop on Vision-based Human Activity Recognition/Conference on Graphics, Patterns and Images (SIBGRAPI), 2014.
  - Santos JR, C. E.; Souza, J. I. C.; Mota, V. F.; Sad, G.; Gorgulho, G.; Araújo, A. A. PanView: An extensible panoramic video viewer for the Web. 9th Latin American Web Congress (LAWEB), 2014, p. 109-113.
  - Sad, D.; Mota, V. F.; Maciel, L.; Vieira, M. B.; Araújo, A. A. A tensor motion descriptor based on multiple gradient estimators. Conference on Graphics, Patterns and Images (SIBGRAPI), 2013, p. 70-74.
- Summer School Participation
  - ENS/INRIA Visual Recognition and Machine Learning Summer School. Paris,France, 22-26 July 2013. Poster presentation - Combining gradient histograms using orientation tensors for human action recognition.

1.5 Structure of the Text

This thesis is structured in the following manner.

Chapter 2: Related Work This chapter reviews some state-of-the-art methods for spatiotemporal representations, focusing in video classification, using handcrafted features, the shallow approaches, and Deep Learning approaches. We could observe that the literature evolved from handcrafted features as HOG and HOF, to middle-level representations, as bag-of-visual-features, the shallow methods. Nowadays, the best results are achieved with deep learning-based approaches. However, deep learning methods are not suitable for all kinds of applications and still have several research open questions as such: coherent data augmentation methods for moving pictures are still in their infancy; adding temporal information in deep neural networks; train with small data could lead to an overfitted network. Therefore, there is still room for improvement in the field of spatiotemporal representation.

Chapter 3: Theoretic Fundamentals In this chapter, we present the mathematical background needed to understand our proposed method. In order to better comprehend the use of tensors, we also present different works in Computer Vision literature which uses this mathematical tool.

Chapter 4: Background for Tensor Representation This chapter presents the background for tensor representation in Human Action Recognition problem. The main hypothesis for all works described was that it was possible to create a simple descriptor using orientation tensors that could maintain balance between size, computer complexity and recognition rate. All those descriptors depend only on the video itself, not requiring any recomputation of the previously computed descriptors after the addition of new videos and/or new action categories to the dataset. The main contribution of these works was to show the power of orientation tensors as video descriptors.

Chapter 5: Proposed Framework In this chapter, we describe our proposed method *FASTensor* with the basic mathematical framework needed for its understanding. The Feature As Spatiotemporal Tensor framework was developed in order to show that orientation tensors can be used as compact spatiotemporal representations, enabling dimensionality reduction and invariance, according to the feature used to built them. That is, in this chapter we present contributions for our first hypothesis. In the sequence, we present our experiments for different applications.

Chapter 6: Human Action Recognition In this chapter, we describe our experiments with the *FASTensor* in the Human Action Recognition problem.

Chapter 7: Pornography Classification In this chapter, we present the experiments that we performed to validate our method for Video Pornography classification. Here, we worked with raw features from handcrafted and deep learning methods.

Chapter 8: Cancer Cell Classification In this chapter, we present our new small dataset, the MCC, and the experiments with the *FASTensor*.

Chapter 9: Conclusion Finally, this chapter presents our final remarks regarding the *FASTensor* and our experimental results in the three applications. We also present our future works. We were able to prove our three hypotheses. We could assert that FASTensors comprise the new state-of-the-art for video classification in the Pornography-800 dataset and for the Melanoma Cancer Cells dataset. For Human Action Recognition, we could also achieve competitive results. Therefore, orientation tensors carry more discriminative information than the feature vector itself, showing how robust is our method.

Chapter 2

Related Work

This chapter reviews related work in spatiotemporal representation. As we aforementioned, spatiotemporal description has several applications, such as, video compression [Sze et al., 2014], object tracking [Pernici and Del Bimbo, 2014], video segmentation [Grundmann et al., 2010; Souza et al., 2014], intelligent video surveillance [Prates and Schwartz, 2018], video and multitemporal image classification [Sivic and Zisserman, 2003; Almeida et al., 2016], cell shape classification [Kriegel et al., 2017].

We focus on review some state-of-the-art methods for spatiotemporal representations, focusing in video and multitemporal images classification, using handcrafted features, the shallow approaches, in Section 2.1 and Deep Learning approaches in Section 2.2. Figure 2.1 presents an example of descriptor creation for each technique.

Shallow methods are categorized into the following classes: *1.* Low-level approaches with handcrafted features; *2.* Bag-of-features representations or middle-level approaches. Some examples of handcrafted features are: HOG, HOF, basis projections, and other OF based features [Dalal et al., 2006]. In most works of the literature, these features are associated with SIFT [Lowe, 1999], SURF [Bay et al., 2008] or STIP [Laptev and Lindeberg, 2003] descriptors.

Deep Learning-based methods share similar procedures: patch sampling, feature description/learning and classification. However, all those steps involve neural networks. Some examples of deep architectures are: Convolutional Neural Networks (CNN) [LeCun, 1998], Recurrent Neural Networks (RNN) [Donahue et al., 2015] and Generative Adversarial Networks (GAN) Goodfellow et al. [2014].

The following sections will present the literature using all those approaches.

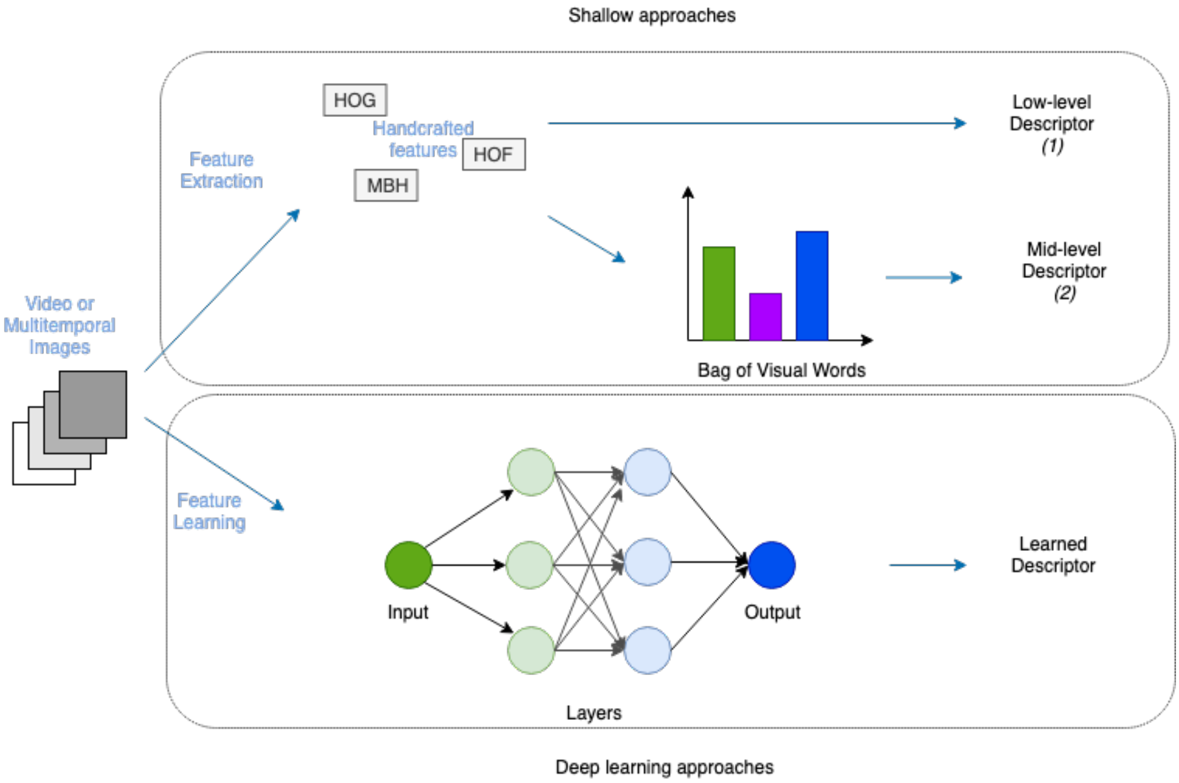


Figure 2.1: Example of descriptor creation. Shallow methods are categorized into the following classes: *1.* Low-level approaches with handcrafted features; *2.* Bag-of-features representations or middle-level approaches. Deep Learning-based methods share similar procedures: patch sampling, feature description/learning and classification. However, all those steps involve neural networks (Image made by the author).

2.1 Shallow Approaches

A handcrafted descriptor based on HOG was presented by Zelnik-manor and Irani [2001]. It is obtained by extracting multiple temporal scales through the construction of a temporal pyramid. To calculate this pyramid, they applied a low-pass filter to the video and sampled it. For each scale, the intensity of each pixel gradient is calculated. Then, a HOG is created for each video and compared with other histograms to classify the database. Later, Laptev et al. [2007] extended this descriptor applying it to other datasets in two different ways: using multiple temporal scales as the original and using multiple temporal and spatial scales.

Laptev et al. [2008] demonstrated the localization of drinking actions in movies by learning a cuboid classifier that combines a set of appearances and motion features. To avoid an exhaustive spatiotemporal search and to improve performance for action localization, the authors proposed to pre-filter possible action localization with a human key-pose detector, trained on key frames of the action. They combined HOG [Dalal and Triggs, 2005] and HOF [Dalal et al., 2006], creating a HOGHOF descriptor.

The BoF [Sivic and Zisserman, 2003] method, or bag-of-visual-feature, is a visual analog to the traditional Bag-of-Word (BoW) representations for text retrieval [Baeza-Yates and Ribeiro-Neto, 1999]. The main idea is to represent a histogram of word occurrences in order to provide a compact representation for the text.

Thus, for a bag-of-visual-feature technique we consider that an image/video is composed of *visual words*. A visual word is a local segment in an image, defined either by a region (image patch or blob) or by a reference point within its neighborhood. The analysis of visual word occurrences and configurations allows us to detect frequent occurrence patterns.

The standard process to create a feature vector with a BoF-based approach follows three steps: (*i*) low-level local descriptor extraction, (*ii*) coding, which performs a point wise transformation of the descriptors into a representation better adapted to the task and (*iii*) pooling, which summarizes the coded features over larger neighborhoods. Classification algorithms are then trained on the final obtained vectors.

An interesting extension of the BoF is the Bag Of Statistical Sampling Analysis (BossaNova) method. The BossaNova method was introduced by Avila et al. [2011] and was extended in Avila et al. [2013]. The main idea of this approach is to improve the pooling step of standard BoF method. The classical BoF has the drawback of compacting all information around a codeword in a single value. The BossaNova pooling improvement is obtained by estimating the distribution of the descriptors around each codeword and compacting this information into several values. The main difference remains on the pooling step. Instead of being represented by a single value in the final image representation, the cluster of codewords is represented by a histogram of distribution around the central codeword.

The BossaNova was extended to video description in Caetano et al. [2016]. The main idea of this work is to develop a mid-level representation based on binary features to reduce the computational cost. The authors compared five handcrafted binary descriptors in a standard bag-of-feature approach and with the BossaNova technique.

In order to add temporal information into video descriptor, Moreira et al. [2016] introduced Temporal Robust Features (TRoF), a spatiotemporal interest point detector and descriptor, which provides a speed compatible with real-time video processing and presents low-memory footprint. It is directly inspired by the still-image SURF detector [Bay et al., 2008].

In video classification, we see tensors in a small number of papers. They can be categorized into two types: those which use tensorial operations to help video analysis [Kim et al., 2007; Krausz and Bauckhage, 2010]; and those which use tensor as a descriptor [Jia et al., 2010; Zhang et al., 2018].

Kim et al. [2007] introduced a new method for gestures and action recognition called Tensor Canonical Correlation Analysis (TCCA), which is an extension of the classical Canonical Correlation Analysis (CCA) for multidimensional vectors. This method can extract flexible and descriptive correlation features of two videos in the joint space-time domain. The proposed statistical framework yields a compact set of pair-wise features. The proposed features were combined with the feature selection method and a nearest neighbor classifier.

According to Krausz and Bauckhage [2010], an action is a sequence of body positions that can be approximated by a linear combination of body parts. Thus, they presented a framework for action recognition that is based on the idea of non-negative tensor factorization. Using several training videos, a set of basis images that represent different parts of the human silhouette is determined. Since different linear combinations of these basis images encode different poses, a particular sequence of poses corresponds to a particular sequence of linear coefficients. By filtering the frames of a video with the basis images, we therefore obtain activation curves that are characteristic for the observed activity. Since non-negative tensor factorization yields basis images that are separable, the proposed filtering mechanism is highly efficient. Finally, the result of the action recognition is determined by using a voting scheme.

Through the use of spectral features of a tensor, Jia et al. [2010] proposed a method for action recognition based on multiresolution features. A series of silhouettes are transformed into an image called Serial-Frame from which are extracted features for the construction of an eigenvalues and eigenvector space called Serials-Frame Tensor. Analyzing this space, they can separate useful information to recognize different types of action. The video database of this work was created by the authors. Similar to Kim et al. [2007] and Krausz and Bauckhage [2010], the tensor used by Jia et al. [2010] is also composed using different modes. However, the idea is to combine five different modes: action, views, people, serials of frames and one-frame feature in a Serial-Frame image. High Order Single Value Decomposition (HOSVD) is applied to decompose the action, view angle and people on the scene.

Zhang et al. [2018] studied the potential for a range of applications towards computerized video classifications via tensor-based video descriptions. The authors proposed a tensor-based logistic regression learning algorithm, in which the weight parameter is regarded to be a tensor, calculated after the CANDECOMP/PARAFAC (CP) tensor decomposition. The CP decomposition decomposes a m-order weight tensor into m factor matrices or the sum of rank-1 tensors. The video classification is determined using this new regression method.

Zhang et al. [2018] studied the potential for a range of applications towards computerized video classifications via tensor-based video descriptions. The authors proposed a tensor-based logistic regression learning algorithm, in which the weight parameter is regarded to be a tensor, calculated after the CANDECOMP/PARAFAC tensor decomposition. The video classification is determined using this new regression method.

Also working with tensor decomposition, Zhang et al. [2017] used tensors to model image sets and proposed a transductive Tensor-driven Low-rank Discriminant Analysis (TLRDA) model on Grassmann manifold for image set classification, in which the tensor-driven low-rank approximation term and discriminant graph embedding term were jointly integrated to learn discriminant feature representation.

A different approach for using tensors in videos is proposed by Baburaj and Sudhish [2019]. Authors reweighted tensor decomposition in order to detect unwanted regions in the video. It is an inpainting technique to remove unwanted moving texts in videos. They also use a low-rank analysis as the method of Zhang et al. [2017].

Kriegel et al. [2017] presented a method based on Discrete Fourier Transform (DFT) to characterize different behaviors in human cells. Traditionally, cell behavior is characterized by tracking the cells movements. Cells can be classified according to their tracking behavior using all or a subset of these kinetic parameters. By carrying out a number of 3D-to-2D projections of surface-rendered cells, the applied method reduces the more complex 3D shape characterization to a series of 2D DFTs. The resulting shape factors are used to train a Self-Organizing Map (SOM) (SOM), which provides an unbiased estimate for the best clustering of the data, thereby characterizing groups of cells according to their shape. It is interesting to note that there are few works about video cell classification, human or non-human.

2.2 Deep Learning Approaches

Deep learning models are a class of machines that aim at learning feature hierarchies with features from higher levels of the hierarchy formed by the composition of lower level features [Bengio, 2009]. Automatically learning features at multiple levels of abstraction allows a system to learn complex functions mapping the input to the output directly from data, without depending so much on human-crafted features. The CNN [LeCun, 1998] are a type of deep models in which trainable filters and local neighborhood pooling operations are applied alternatively to the raw input images, resulting in a hierarchy of increasingly complex features.

In general, deep learning methods need to pre-train the neural network in order to have a better convergence of the model. After all, CNN's are high-performance classifiers and it is necessary to learn a great amount of parameters to converge [Huh et al., 2016].

As a class of attractive deep models for automated feature construction, CNNs have been primarily applied to 2D images in image classification. One natural way to encode spatiotemporal information in videos is to extend the convolution kernels in CNN from 2D to 3D and train a brand new 3D CNN [Qiu et al., 2017].

One of the first deep model technique to be applied for spatiotemporal representation was presented by Ji et al. [2013] for the problem of human action recognition. They developed a 3D CNN model. This model extracted features from both the spatial and the temporal dimensions by performing 3D convolutions, thereby capturing the motion information encoded in multiple adjacent frames. The developed model generated multiple channels of information from the input frames, and the final feature representation combines information from all channels. However, they did not surpass the baseline methods for in the wild datasets.

Later in [Tran et al., 2015], the authors devised a widely adopted 11-layer 3D CNN (C3D) for learning video representation over 16-frame video clips in the context of large-scale supervised video datasets, and temporal convolutions across longer clips (100 frames) are further exploited by Varol et al. [2018]. However, the capacity of existing 3D CNN architectures is extremely limited with expensive computational cost and memory demand, making it hard to train a very deep 3D CNN.

To overcome the timing consuming problem, Qiu et al. [2017] not only proposed the idea of simulating 3D convolutions with 2D spatial convolutions plus 1D temporal connections which is more economic, but also integrated this design into a deep residual learning framework for video representation learning.

Adult Content Recognition with Deep Neural Networks (ACORDE) [Wehrmann et al., 2018] are Deep Neural Networks (DNN) proposed for classification of adult content in videos. These architectures are composed of pretrained CNN [Krizhevsky et al., 2012] for image classification appended by RNN [Donahue et al., 2015]. The pretrained CNN performs deep feature extraction on the raw video frames, outputting a high semantic level representation of the images to the RNN, which handles the temporal coherence between frames. ACORDE was tested using both GoogLeNet [Szegedy et al., 2015] and Residual Network (ResNet) architectures [He et al., 2016] as feature extraction, CNNs and Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997] networks for temporal coherence.

Also using a CNN, Perez et al. [2017] proposed to add motion information to the deep architecture throughout an optical flow late fusion. The authors use convolutional neural networks along with static (image) and motion information (optical flow). In the static pipeline they start with a chosen sampling of the video frames and extract their features with a CNN. The motion information is extracted using the optical flow displacement fields from Brox et al. [2004]. Therefore, they have a CNN for motion, a CNN for static and finally a late fusion to obtain the best recognition rate. In the late fusion, each information is processed by a separate decision-making approach (*e.g.*, SVM classifier), generating independent classification scores that can then be combined later on a single score for the final classification.

We can observe that those models have been applied to deep architecture model and showed very impressive performance. Nowadays, image and video classification problems in large databases are being treated with Deep Learning tools [Karpathy et al., 2014; Wehrmann et al., 2018]. However, coherent approaches for moving pictures are still in their infancy as also adding the temporal information on a deep architecture [Wang et al., 2018].

Another problem for Deep Learning techniques is that deep architecture model suffers from overfitting problem when there is a small number of training data. Some strategies to overcome this problem are Transfer Learning, Data Augmentation, Data Generation and Hybrid Approaches.

Transfer Learning is a machine learning problem that focuses on storing knowledge gained while solving one problem and applying it to a different one. This is widely used in CNN's, when we pre-train the network in order to have better convergence, as described by Qiu et al. [2017]. The main problem of this technique is that it assumes a baseline data availability on which it builds [Huh et al., 2016].

Data Augmentation is a set of techniques where your data is augmented via a number of random transformations so that the deep learning model would never see the same picture twice. This also helps prevent over-fitting and helps the model generalize better. Techniques include rotation, flipping, zooming, width or height shift, *etc.* applied to an image in combination. The main problem of this technique is that not all domains are invariant to all those transformations [Ben-Cohen et al., 2018].

A very interesting approach is if we do not have the data, we can create the data, the Data Generation. GAN are a kind of generative models designed by Goodfellow et al. [2014]. The model consists of two networks that are trained in an adversarial process where one network generates fake images and the other network discriminates between real and fake images repeatedly. GANs have gained great popularity in the computer vision community and different variations of GANs were recently proposed for generating high quality realistic natural images. Recently, several medical imaging applications have applied the GAN framework [Frid-Adar et al., 2018]. However, for videos, this is still too incipient [Wang et al., 2018]. And even for medical images, we still can work only with very controlled environments like computed tomography, and magnetic resonance applications, where the subject is always in the same position and just the anomaly is different [Varghese et al., 2017].

It is also possible to combine shallow and deep methods to overcome the small number of data, creating a Hybrid Approach. One idea is to apply deep features to shallow methods, preventing the overfitting and using less layers for the neural network [Pasupa and Sunhem, 2016].

Thus, we can observe that depending on the data we are studying, there are no similar data available for transfer learning, there is no possible way to synthesize similar videos, data augmentation does not work and, yet, if we are working with small video datasets, the problem is also how to add coherent temporal information without having sufficient data to train. Hence, this is still an open problem in literature. Moreover, we can observe that deep learning is not suitable for all kind of classification problems.

We will discuss small data problems in Chapter 8, where we present a new small video dataset where those techniques do not work, therefore, deep learning-based approaches are not suitable.

Chapter 3

Theoretic Fundamentals

In this chapter, we present the mathematical background needed to understand our proposed method (Section 3.1). In order to better comprehend the use of tensors, we also present different works in Computer Vision literature which use this mathematical tool (Section 3.2).

3.1 Mathematical background

3.1.1 Basic Notations

The adopted notations follow common standards used in functional analysis, tensor theory and signal analysis [Santos, 2017; Westin, 1994].

Scalars A scalar is an element of a field which is used to define a vector space. In this work, it is denoted with italic lower case such as a.

Vectors A concept described by multiple scalars, such as one having both direction and magnitude, is called a vector. In this work, it is denoted with bold lower case as $\hat{\mathbf{v}}$.

Matrices A matrix, $m \times n$, denoted with capital case as M is a table of scalars disposed in m rows and n columns:

$$M = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}. \tag{3.1}$$

The transpose of a $m \times n$ matrix M is defined by a matrix $A = M^T$, $n \times m$.

The inverse of a $m \times n$ matrix M is defined by a matrix $B = M^{-1}$, $m \times n$. For a matrix be invertible it must be square, that is $m = n$, and its determinant must be different from zero.

The determinant of a square matrix M is a useful value computed from its inner elements and denoted $det(M)$ or $|M|$. The value of determinant of a matrix can be calculated by following procedure: For each element of first row or first column get cofactor of those elements and then multiply the element with the determinant of the corresponding cofactor, and finally add them with alternate signs. As a base case the value of determinant of a 1*1 matrix is the single value itself. Cofactor of an element, is a matrix which we can get by removing row and column of that element from that matrix.

Ellipse Analysis An ellipse is a curve in a plane surrounding two focal points such that the sum of the distances to the two focal points is constant for every point on the curve. As such, it is a generalization of a circle, which is a special type of an ellipse having both focal points at the same location.

Given the Cartesian coordinates such that the origin is the center of the ellipse, the x-axis is the major axis, the *foci* are the points $f_1 = (c, 0)$ and $f_2 = (-c, 0)$ and the vertices are $v_1 = (a, 0)$ and $v_2 = (-a, 0)$ (See Figure 3.1).

For an arbitrary point (x, y) the distance of to focus $(c, 0)$ is $\sqrt{(x-c)^2 + y^2}$ and to the second focus is $\sqrt{(x+c)^2 + y^2}$. By definition, the point (x, y) is on the ellipse if the condition $\sqrt{(x-c)^2 + y^2} + \sqrt{(x+c)^2 + y^2} = 2a$ is fulfilled. Therefore, using the relation $a^2 = b^2 + c^2$ we obtain the equation of the ellipse:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1. \tag{3.2}$$

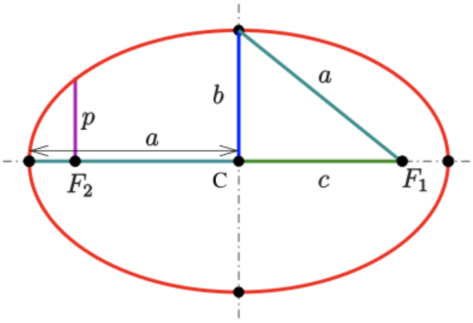


Figure 3.1: Ellipse in Cartesian coordinates with the shape parameters a semi-major axis, b semi-minor axis, c linear eccentricity, p semi-latus rectum (Image made by the author).

Moving the origin of the ellipse, we could rewrite this equation as:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = f, \tag{3.3}$$

where f is a function.

Using matrix notation, where $\hat{\mathbf{v}} = (x, y)$, we rewrite Equation 3.3 as:

$$\hat{\mathbf{v}} \begin{bmatrix} a & \frac{b}{2} \\ \frac{a}{2} & b \end{bmatrix} \hat{\mathbf{v}}^T = f\hat{\mathbf{v}}A\hat{\mathbf{v}}^T = f. \tag{3.4}$$

Therefore, any symmetric matrix created from $\hat{\mathbf{v}}A\hat{\mathbf{v}}^T$ can be interpreted as an ellipse. This notion can be extended to R^n.

3.1.2 Feature Vector

Suppose we are given a n dimensions feature extracted from a signal which components x_1, x_2, ..., x_n are independent normal random variables. Let these n random variables correspond to n indices t_1, ..., t_n.

This feature vector can be a dense sampling or grid extraction (handcrafted), or can be learned features, from a sequence of images, as depicted in Figure 3.2.

The distribution of a Gaussian process is the joint distribution of all those random variables, for any $n = 1, 2, \ldots$ and indices $t_1, \ldots, t_n$. That is, the joint probability density function is given by:

$$f_{x_1,\ldots,x_n}(x_1, \ldots, x_n) = \frac{|C_x^{-1}|^{\frac{1}{2}}}{(2\pi)^{\frac{n}{2}}} exp\left\{-\frac{[x-\mu_x]^T C_x^{-1}[x-\mu_x]}{2}\right\}, \tag{3.5}$$

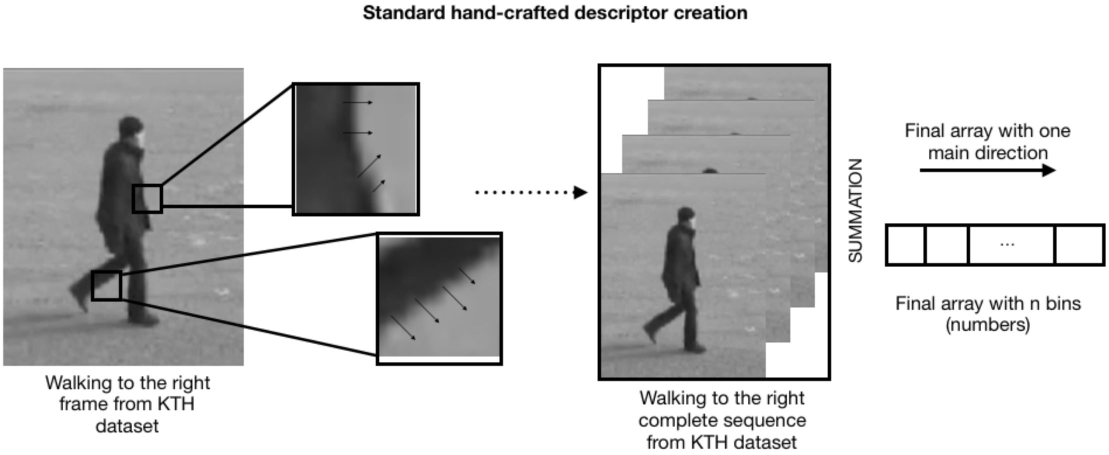

Standard hand-crafted descriptor creation
Walking to the right frame from KTH dataset
Walking to the right complete sequence from KTH dataset
SUMMATION
Final array with one main direction
...
Final array with n bins (numbers)

Figure 3.2: Geometry example of the feature vector $\hat{\mathbf{v}}$ (handcrafted) extraction and the final array creation to describe a video sequence. The image is an example of walking action from KTH dataset [Schuldt et al., 2004] (Image made by the author).

where μ_x is the components media.

Rewriting Equation 16, we have:

$$f_{x_1,\ldots,x_n}(x_1,\ldots,x_n) = \frac{exp\left\{-\frac{1}{2}[x-\mu_x]^T[C_x]^{-1}[x-\mu_x]\right\}}{\sqrt{(2\pi)^{\frac{n}{2}}|C_x|}}, \tag{3.6}$$

where $x-\mu_x$ and C_x are defined by:

$$x-\mu_x = \begin{bmatrix} x_1-\mu_{x_1} \\ x_2-\mu_{x_2} \\ \vdots \\ x_n-\mu_{x_n} \end{bmatrix}, \tag{3.7}$$

$$C_x = \begin{bmatrix} c_{11} & c_{12} & \ldots & c_{1n} \\ c_{21} & c_{22} & \ldots & c_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ c_{n1} & c_{n2} & \ldots & c_{nn} \end{bmatrix}, \tag{3.8}$$

$$C_{ij} = E[(x_i-\mu_{x_i})(x_j-\mu_{x_j})] = \left\{ \begin{array}{ll} \sigma^2_{x_i}, & i=j \\ C_{x_i x_j}, & i \neq j \end{array} \right\}. \tag{3.9}$$

The mean values μ_{x_i} of $x(t_i)$ are:

$$\overline{x_i} = E[x_i] = E[x(t_i)]. \tag{3.10}$$

The covariance matrix C_x elements are:

$$\begin{aligned} C_{ik} \quad = C_{x_i x_k} \quad &= E[(x_i - \mu_{x_i})(x_k - \mu_{x_k})] \\ &= E[\{x(t_i) - E[x(t_i)]\}\{x(t_k)E[x(t_k)]\}] \ . \\ &= C_{xx}(t_i, t_k) \end{aligned} \tag{3.11}$$

Assume that the components x_1, x_2, ..., x_n can be arranged into a feature vector $\hat{\mathbf{x}}$. Then, its covariance matrix $[C_{\hat{\mathbf{x}}}]$ (Equation 22) can be created by:

$$C_{\hat{\mathbf{x}}} = \mathrm{E}\left[(\hat{\mathbf{x}} - \mathrm{E}[\hat{\mathbf{x}}])\,(\hat{\mathbf{x}} - \mathrm{E}[\hat{\mathbf{x}}])^{\top}\right]. \tag{3.12}$$

3.1.3 Covariance Matrices and Orientation Tensors

The covariance matrix (also known as dispersion matrix or variance-covariance matrix) is a matrix whose element in the i, j position is the covariance between the i_{th} and j_{th} elements of a random vector. A covariance matrix C is symmetric, thus, from the spectral decomposition theorem it can be defined as:

$$C = \sum_{i=1}^{m} \hat{\mathbf{e}}_i \hat{\mathbf{e}}_i^T. \tag{3.13}$$

Therefore, the covariance matrix C can be considered an Orientation Tensor.

According to Westin [1994], the orientation tensor is a real and symmetric matrix, in this way it can be decomposed using the spectral decomposition theorem as follows:

$$T = \sum_{i=1}^{n} \lambda_i T_i, \tag{3.14}$$

where λ_i are eigenvalues of tensor T.

Projecting T over a space of dimension m, we have the following decomposition:

$$T_i = \sum_{s=1}^{m} \hat{\mathbf{e}}_s \hat{\mathbf{e}}_s^T, \tag{3.15}$$

where $\{\hat{\mathbf{e}}_1, ..., \hat{\mathbf{e}}_m\}$ is a basis from R^m.

An interesting decomposition of the orientation tensor T is given by

$$T = \lambda_n T_n + \sum_{i=1}^{n-1} (\lambda_i - \lambda_{i+1}) T_i, \tag{3.16}$$

where λ_i are eigenvalues corresponding to each eigenvector e_i. This decomposition is interesting because of its geometric interpretation. In fact, in R^3, the orientation tensor T decomposed using the Equation 3.16 can be represented by a spear, a plate and a sphere (Figure 3.3).

$$T = (\lambda_1 - \lambda_2)T_1 + (\lambda_2 - \lambda_3)T_2 + \lambda_3 T_3. \tag{3.17}$$

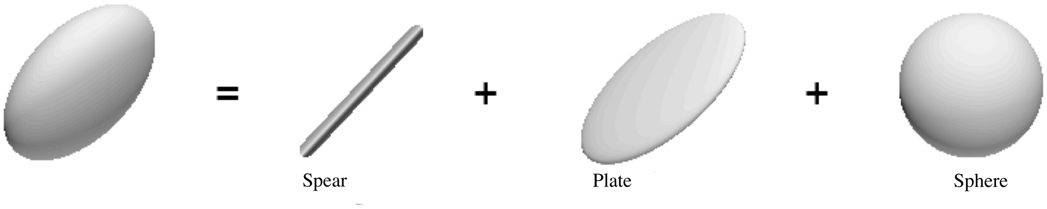


Figure 3.3: Decomposition of a tensor in R^3 in its components: spear, plate and sphere (Image made by the author).

The orientation tensor in R^3 decomposed as Equation 3.17, with eigenvalues $\lambda_i \geq 0$, $i = 1, 2, 3$, can be interpreted as:

- $\lambda_1 >> \lambda_2 \approx \lambda_3 \geq 0$ corresponds approximately to the linear tensor, with the dominant spear component.

- $\lambda_1 \approx \lambda_2 >> \lambda_3 \geq 0$ corresponds approximately to the flat tensor, with the dominant plate component.

- $\lambda_1 \approx \lambda_2 \approx \lambda_3 \geq 0$ corresponds approximately to the isotropic tensor, with the dominant sphere component and no dominant orientation.

An orientation tensor T can be created from a vector using its transpose as follows:

$$T = \hat{\mathbf{v}}\hat{\mathbf{v}}^T, \tag{3.18}$$

where T is the tensor created from the product between the vector $\hat{\mathbf{v}}$ and its transpose $\hat{\mathbf{v}}^T$. Then, the tensor T is a matrix $n \times n$, with n being the size of the vector $\hat{\mathbf{v}}$. Therefore, it is possible to consider the orientation tensor as a measure of covariance of the elements of the vector. It is important to note that the sums of orientation tensors are also orientation tensors.

3.2 Tensors in Computer Vision

In mathematics and physics, directional quantities are often represented as vectors, which are often triples (or n-tuples) of scalar numbers that behave according to a set of properties. A more complex structure is the tensor. Tensors extend the concept of vectors and matrices for larger orders. In tensorial terminology, vectors are first-order tensors and matrices are second-order tensors.

Depending on the transformation properties of a tensor, it will be categorized as being a covariant tensor, a contravariant tensor or a mixture of these two, *i.e.*, a mixed tensor [Westin, 1994]. The most common tensors used in image processing are the structure and orientation tensors. Note that the structure tensor is a specific case of orientation tensor [Johansson et al., 2002] and both can be categorized as covariant tensors.

Orientation tensors are powerful representations of local orientation. They take the form of an $m \times m$ real symmetric matrix for m-dimensional signals. Given a signal represented by a vector $\hat{\mathbf{v}}$ with m elements, it can be represented by the tensor $T = a\hat{\mathbf{v}}\hat{\mathbf{v}}^T$, where a is a scalar value that may encode information other than orientation.

For a non-simple signal, it is desired that the eigenvector with the largest eigenvalue of the tensor points out the dominant direction of the signal. This is called an anisotropic tensor. On the other hand, a signal with no dominant direction is preferably represented by an isotropic tensor. An extended mathematical explanation is depicted in Chapter 5.

One widely used application of tensorial analysis is velocity estimation in video sequences (or OF) [Lucas and Kanade, 1981a; Augereau et al., 2005]. In both works, structure tensor based on image gradients are used. It is interesting to note that this type of tensor is rotation equivariant for all types of signals [Johansson et al., 2002]. Another optical flow method is presented by Farnebäck [2001] using orientation tensors.

Given the image I constraint equation:

$$I(x_1, x_2, t) = I(x_1 + d_{x_1}, x_2 + d_{x_2}, t + 1), \tag{3.19}$$

optical flow methods try to find the velocity vector $\vec{v}$ in time t, where $\vec{v} = (v_{x_1}, v_{x_2}, 1)^T$, that satisfies the problem of optical flow constraint:

$$\nabla I . \vec{v} + I_t = 0. \tag{3.20}$$

Equation 3.20 only allows the parallel component of $\hat{v} = \frac{\nabla I}{||\nabla I||}$ to be obtained. This is known as the aperture problem.

Thus, to solve the optical flow constraint problem, Lucas and Kanade [1981a] minimize the energy function:

$$\int_{W(x)} (\nabla_{xt} I \ . \ \vec{v})^2 dx' = \int_{W(x)} \vec{v}^T (\nabla_{xt} I)(\nabla_{xt} I)^T \vec{v} dx', \tag{3.21}$$

where $W(x)$ is the neighborhood where the mean is calculated. Note that the tensor is present in $(\nabla_{xt} I)(\nabla_{xt} I)^T$ and represents the structure tensor of image gradients.

Augereau et al. [2005] used this same tensor in order to solve the optical flow constraint problem. However, the velocity vector $\vec{v}$ is found using the spectral elements of the tensor.

Let us call the structure tensor:

$$S = \nabla I \nabla I^T = \begin{pmatrix} I_{x_1}^2 & I_{x_1} I_{x_2} & I_{x_1} I_t \\ I_{x_1} I_{x_2} & I_{x_2}^2 & I_{x_2} I_t \\ I_{x_1} I_t & I_{x_2} I_t & I_t^2 \end{pmatrix}, \tag{3.22}$$

where $\nabla I = (\partial_{x_1} I, \partial_{x_2} I, \partial_{x_t} I)$. Its spectral elements are the eigenvalues λ_i:

$$\begin{aligned} \lambda_1^{(S)} &= I_{x_1}^2 + I_{x_2}^2 + I_t^2 \\ \lambda_2^{(S)} &= \lambda_3^{(S)} = 0. \end{aligned} \tag{3.23}$$

Thus, the eigenvector associated with $\lambda_1^{(S)}$ is the gradient $v_1^{(S)} = \nabla I$. In order to create an orthogonal basis, the other eigenvectors are:

$$\begin{aligned} v_1^{(S)} &= I_{x_1} I_{x_2} I_t, \\ v_2^{(S)} &= I_{x_2} - I_{x_1} 0, \\ v_3^{(S)} &= I_{x_1} I_t I_{x_2} I_t - [I_{x_1}^2 + I_{x_2}^2]. \end{aligned} \tag{3.24}$$

Finally, an optical flow approximation is obtained using the third eigenvector $v_3^{(S)}$.

For Farnebäck [2001], the parameters of the velocity model can be estimated directly from tensors in the region by using $\vec{v}^T T \vec{v}$ ($\vec{v}$ from Equation 3.20) as a cost function. T is an orientation tensor defined for an image I as

$$T = \tilde{T} - \lambda_{min} I, \tag{3.25}$$

where λ_{min} is the smallest eigenvalue of $\tilde{T}$ and $\tilde{T} = AA^T + \gamma bb^T$, with A and b computed by a Gaussian weighted least square approximation of the signal $f(x) \sim x^T A x + b^T x + c$; and γ is a non-negative weight factor.

As a part of the process of locating local image features, tensors are used to locate corners and junctions [Harris and Stephens, 1988; Förstner, 1994]. Structure tensors (Equation 3.22) presented for optical flow are also used in these approaches.

According to Harris and Stephens [1988], an interest point is characterized by a large variation in all directions of the gradient vector. By analyzing the eigenvalues of tensor S, an interest point should have two large eigenvalues. In some cases, one may wish to compute the location of a corner with subpixel accuracy. To achieve an approximate solution, Förstner [1994] algorithm uses least squares to compute the closest point to all the tangent lines of the corner in a given window, relying on the fact that tangent lines cross at a single point for an ideal corner.

Tensors are also applied to adaptive filtering [Granlund and Knutsson, 1995; Perona and Malik, 1990]. An adaptive filter is a filter that self-adjusts its transfer function according to an optimization algorithm driven by an error signal. The basic idea of these works is to use a filter that is locally adapted to the image and to estimate local orientation in terms of tensor T at each image point x. Then, use T_x to locally control the adaptive filter. There are two approaches for this filtering: Anisotropic Diffusion [Perona and Malik, 1990] and Image Enhancement [Granlund and Knutsson, 1995]. For the Anisotropic Diffusion, also called the Perona-Malik Diffusion, the filter for an image I is described by Equation 3.26, where D is a diffusion tensor controlled by T. For Image Enhancement filtering, filters are locally synthesized in the Fourier domain and are represented by Equation 3.27.

$$\frac{\partial I}{\partial s} = \frac{1}{2} \Delta(D \nabla I), \tag{3.26}$$

$$\hat{u}^T T \hat{u} = \langle \hat{u}\hat{u}^T | T \rangle. \tag{3.27}$$

Other image processing applications include representing multiresolution edges [Castro et al., 2009; Mota et al., 2009], detecting contours [Andaló et al., 2007], medical image filtering [Saha and Xu, 2010] and in biological image segmentation [Xu et al., 2012]. The last three use a method called tensor-scale, which takes into account three characteristics of a tensor in R^2: orientation, anisotropy and thickness.

To represent multiresolution edges, Castro et al. [2009] and Mota et al. [2009] used high frequency information extracted from wavelet decomposition. For each scale j, a vector $v_{j,p}$ is created:

$$v_{j,p} = (I.\psi^1_{j,p}, I.\psi^2_{j,p}, I.\psi^3_{j,p})^T, \tag{3.28}$$

where $v_{j,p}$ is the inner product of image I and the wavelet basis $\{\psi^1_{j,p}, \psi^2_{j,p}, \psi^3_{j,p}\}$. This vector contains the high frequency value at vertical, horizontal and diagonal directions of the image I at the position p and scale j.

Symmetric rank 2 tensors are then created $M_{j,p} = v_{j,p}v^T_{j,p}$ and the final tensor $M_{0,p}$ is computed for each pixel of the original image using

$$M_{0,p} = \sum_{j=1}^{n_j} k_j M_{j,p}, \tag{3.29}$$

where k_j is a weight assigned to each class based on the tensor trace.

The tensors are then decomposed and their eigenvalues are extracted. The values $\lambda_1 - \lambda_2$ are computed and normalized. They indicate the collinearity of the interpolated tensors. The authors argue that when the main direction is coincident with the variation of $\lambda_1 - \lambda_2$ there is a salient multiresolution region.

The tensor-scale often used in the literature [Andaló et al., 2007; Saha and Xu, 2010; Xu et al., 2012] of pixel p in a gray-level image can be defined as the largest ellipse within the same homogeneous region, centered in p. The homogeneous region is defined based on a predefined criterion and, for binary images, it is naturally defined by the object pixels. The tensor-scale ellipse is obtained from pairs of radially opposite sample lines, that are traced emerging from the center pixel. The method was originally proposed by Saha and Xu [2010], and extended/improved by Xu et al. [2012] and Andaló et al. [2007].

The next Chapter presents our early works using Orientation Tensor for Human Action Recognition.

Chapter 4

Background for Tensor Representation

In this chapter, we present the background for tensor representation in Human Action Recognition problem.

The first tensor descriptor used polynomials projections of optical flow [Mota et al., 2012]. Following the idea to code features into tensor, the authors developed a tensor descriptor based on histogram of gradients [Perez et al., 2012]. In order to improve the recognition and add more local information, authors improved the descriptor of Perez et al. [2012] in Oliveira et al. [2015]. As all approaches use differential operators, the authors asked which operator is more suitable for recognizing actions in videos [Sad et al., 2013]. Table 4.1 resumes this four different approaches.

Table 4.1: Four different approaches to use tensor descriptors for Human Action Recognition.

Method	Publication
Accumulating optical flow projections coded into orientation tensors	[Mota et al., 2012]
Accumulating histogram of gradients coded into orientation tensors	[Perez et al., 2012]
Variable size block matching algorithm associated with orientation tensors	[Oliveira et al., 2015]
Combining multiple gradient estimators	[Sad et al., 2013]

4.1 Tensor Accumulation of Optical Flow Projections

Mota et al. [2012] created a tensor descriptor in order to obtain simple descriptors with a good balance between their size and recognition rate. This work had two main hypothesis:

- The optical flow projection on Legendre polynomials could lead to an effective motion estimation and dimension reduction;

- Orientation tensors could capture the covariance information from coefficients extracted by the projection.

The descriptor proposed is obtained by combining polynomial coefficients calculated for each image in a video. The coefficients are found through the projection of the optical flow on Legendre polynomials, reducing the dimension of per frame motion estimations. The sequence of coefficients are then combined using orientation tensors.

In the feature extraction step, they approximate the optical flow through the projection into Legendre polynomials. The basic idea of a polynomial based model is to approximate a vector field with a linear combination of orthogonal polynomials [Druon, 2009; Kihl et al., 2010]. Let us define F on the domain Ω an optical flow:

$$F : \begin{array}{l} \Omega \subset R^2 \rightarrow R^2 \\ (x_1, x_2) \mapsto (V^1(x_1, x_2), V^2(x_1, x_2)) \end{array},$$

where the functions $V^1(x_1, x_2)$ and $V^2(x_1, x_2)$ correspond to the horizontal and vertical displacement of the point $(x_1, x_2) \in \Omega$.

The optical flow is computed by the method described in Augereau et al. [2005]. This method was chosen because we found experimentally that it computes a more regular optical flow than the one computed by the standard Lucas-Kanade [Lucas and Kanade, 1981a]. Thus, it is more suitable to model the motion from the frame using polynomials, which are continuous functions.

This optical flow is then approximated by projecting the displacement functions onto each polynomial $P_{i,j}$, which belong to an orthogonal basis, as such Legendre basis. It reduces the dimension of the optical flow field. We can express $\tilde{F} = (\tilde{V}^1(x_1, x_2), \tilde{V}^2(x_1, x_2))$, using a basis of degree g, as:

$$\left\{ \begin{array}{l} \tilde{V}^1(x_1, x_2) = \sum_{i=0}^{g} \sum_{j=0}^{g-1} \tilde{v}^1_{i,j} P_{i,j} \\ \tilde{V}^2(x_1, x_2) = \sum_{i=0}^{g} \sum_{j=0}^{g-1} \tilde{v}^2_{i,j} P_{i,j} \end{array} \right.,$$

where

$$\left\{ \begin{array}{l} \tilde{v}^1_{i,j} = \int \int_{\Omega} V^1(x_1, x_2) P_{i,j} \omega(x_1, x_2) dx_1 dx_2 \\ \tilde{v}^2_{i,j} = \int \int_{\Omega} V^2(x_1, x_2) P_{i,j} \omega(x_1, x_2) dx_1 dx_2 \end{array} \right., \tag{4.1}$$

It is important to note that the number of polynomials n which compose a basis of degree g is:

$$n_g = \frac{(g+1)(g+2)}{2}.$$

In order to capture the motion variation in time, it is possible to use both the polynomial coefficients $\tilde{v}^1_{i,j}$ and $\tilde{v}^2_{i,j}$ (Equation 4.1) and an approximation of their first temporal derivative $\partial\tilde{v}^q_{i,j} = \tilde{v}^q_{i,j}(f) - \tilde{v}^q_{i,j}(f-1)$ with $i+j < g$, to create a vector $\tilde{v_f}$ for each frame f of the video:

$$\tilde{v_f} = [\tilde{v}^1_{0,0}, ..., \tilde{v}^1_{g,0}, \quad \tilde{v}^2_{0,0}, ..., \tilde{v}^2_{g,0}, \quad \partial\tilde{v}^1_{0,0}, ..., \partial\tilde{v}^1_{g,0}, \quad \partial\tilde{v}^2_{0,0}, ..., \partial\tilde{v}^2_{g,0}].$$

Using the vector $\tilde{v_f}$, they generates an orientation tensor $T_f = \tilde{v_f}\tilde{v_f}^T$ for each frame f of the video, which is a $4n_g \times 4n_g$ matrix. This orientation tensor captures the covariance information between $\tilde{v}^1_{i,j}$ and $\tilde{v}^2_{i,j}$. It carries only the information of the polynomial of frame f and its rate of change in time.

The motion average of consecutive frames is expressed using a series of tensors. This can be achieved by $T^{OF} = \sum_a^b T_f$ using all video frames or an interval of interest. By normalizing T_f with a L_2 norm, they are able to compare different video clips or snapshots regardless of their length or image resolution.

Instead of using the entire optical flow of the video frames, it is also possible to use only the optical flow from a region with most representative motion. Therefore, they tested a sliding window with fixed dimensions placed around the subject doing the action. The center of mass of global optical flow gives the center of the window.

To validate this tensor descriptor, authors used the KTH video dataset [Schuldt et al., 2004].

4.2 Tensor Accumulation of Histogram of Gradients

Following the idea to explore other features that could be coded into orientation tensors, Perez et al. [2012] developed a tensor descriptor based on histogram of gradients (HOG). The histograms of gradients obtained per frame are combined into an orientation tensor, making it a simple, fast to compute and effective global descriptor. This work had two hypothesis:

- Histogram of gradients can be coded into orientation tensors in order to create a video descriptor.

- Orientation tensor accumulation can be used to reduce dimension of HOG features.

The feature extraction step of this method extracts gradients for each frame. The partial derivatives of the j-th image video frame I at point p

$$\vec{g}_t(p) = [dx\ dy\ dt] = \left[\frac{\partial I_j(p)}{\partial x}\ \frac{\partial I_j(p)}{\partial y}\ \frac{\partial I_j(p)}{\partial t}\right],$$

or, equivalently, in spherical coordinates $\vec{s}_t(p) = [\rho_p\ \ \theta_p\ \ \psi_p]$ with $\theta_p \in [0,\ \pi]$, $\psi_p \in [0,\ 2\pi)$ and $\rho_p = ||\vec{g}_t(p)||$, indicate brightness variation that might be the result of local motion.

The gradient of all n points of the image I_j can be compactly represented by a tridimensional histogram of gradients $\vec{h}_j = \{h_{k,l}\}$, $k \in [1, nb_\theta]$ and $l \in [1, nb_\psi]$, where nb_θ and nb_ψ are the number of cells for θ and ψ coordinates respectively. There are several methods for computing the HOG. They performed some experiments with the icosahedron discretization [Kläser et al., 2008], yet no significant enhancement was detected. Thus, authors chose a uniform subdivision of the angle intervals to populate the $nb_\theta \cdot nb_\psi$ bins (Equation 4.2), since it achieves good results and it is fast to compute.

$$h_{k,l} = \sum_p \rho_p \cdot w_p, \tag{4.2}$$

where $\{p \in I_j \mid k = 1 + \left\lfloor \frac{nb_\theta \cdot \theta_p}{\pi} \right\rfloor, l = 1 + \left\lfloor \frac{nb_\psi \cdot \psi_p}{2\pi} \right\rfloor\}$ are all points whose angles map to k and l bins, and w_p is a per pixel weighting factor which can be uniform or Gaussian as in Lowe [1999]. The whole gradient field is then represented by a vector $\vec{h}_j$ with $nb_\theta \cdot nb_\psi$ elements.

Analogously to the previous descriptor (Sec. 4.1), the HOGs with m bins $\vec{h}_j$, computed for j-th frames, can be combined in a tensor as follows:

$$T^{HOG} = \sum_j \vec{h}_j \vec{h}_j^T,$$

using all video frames or an interval of interest. By normalizing T^{HOG} with a L_2 norm, they are able to compare different video clips or snapshots regardless of their length or image resolution.

When the gradient histogram is computed using the whole image, the cells are populated with vectors regardless of their position in the image. This implies a loss in the correlation between the gradient vectors and their neighbors. As observed in several works [Lowe, 1999], the subdivision of the video into cubes of frames enhances the recognition rate, using a Gaussian weight for w_p. Suppose the video frame f is uniformly subdivided into $\vec{x}$ and $\vec{y}$ directions by a grid with n_x and n_y non-overlapping blocks. Each block can be viewed as a distinct video varying in time. The smaller images result in gradient histograms $\vec{h}_j^{c,r}$, $c \in [1, n_x]$ and $r \in [1, n_y]$, with better position correlation. The tensor for the frame j is then computed as the addition of all block tensors:

$$T_j = \sum_{c,r} \vec{h}_j^{c,r} \, \vec{h}_j^{c,r\,T} , \tag{4.3}$$

which captures the uncertainty of the direction of the m-dimensional vectors $\vec{h}_f^{a,b}$ in the frame j. This tensor is normalized using the L_2 norm. The image subdivision does not change the descriptor size, and the accumulation described above is the same. The global descriptor with image subdivision and histograms of gradients is then

$$T^{HOG} = \sum_{j=1}^{f} T_j .$$

Another improvement is to accumulate the tensor obtained with the video frame flipped horizontally. Therefore, the HOG3D is computed for each block, the final tensor is computed (Equation 4.3) and simply added to the original frame tensor. This flipped version enforces horizontal gradient symmetries that occur on the video, even those between multiple frames.In our experiments, all HOG descriptors were obtained using this improvement.

To validate this tensor descriptor, authors used the KTH video dataset [Schuldt et al., 2004] and the Hollywood2 dataset [Marszałek et al., 2009].

4.3 Tensor Descriptor Based on Variable Size Block Matching

Feature trajectories are based on spatial interest points tracked over time. The shape of trajectories encode the information about local motion patterns. Feature trajectories are typically extracted using Kanade-Lucas-Tomasi (KLT) tracker [Lucas and Kanade, 1981b] or matching Scale-Invariant Feature Transform (SIFT) [Lowe, 1999] descriptors between frames. Those approaches have a high computational complexity.

Oliveira et al. [2014] and Oliveira et al. [2015] argued that the Block Matching method runs fast and can potentially generate compact descriptors since it is widely used in video compression. It is a simple method compared to other approaches to extract motion information like 3D gradients and optical flow, because it yields a more coarse representation, with less vectors per frame. Moreover, using variable block sizes, the method is able to cover more homogeneous regions and to avoid redundancies. Authors had two main hypothesis:

- Variable size block matching algorithm could potentially generate simple and compact video descriptors;
- Orientation tensors could be used to code trajectories extracted from block matching algorithm.

Variable Size Block Matching algorithm (VSBMA) consists in dividing a so called "reference frame" into blocks of pixels and finding for each one a corresponding block in a so called "target frame" which minimizes a dissimilarity (or error) function. For each block, the algorithm outputs a displacement vector between the coordinates of a reference block and its corresponding target. When matching blocks, the blocks evaluated in the target frame are restricted to a close vicinity of the block coordinates in the reference frame. This vicinity is called a search window. The search window is established under the assumption that the movement of objects between frames is somewhat smooth, so it is not necessary to search the whole target frame. This greatly reduces the cost of the search, especially for higher resolution sequences. Note also that the size of the search window limits the magnitude of the displacement vectors.

Another noteworthy element of the algorithm is the search strategy. Even with a search window, evaluating all blocks within it is still too big an effort. To reduce this cost, this work uses a Four Step Search (4SS) as the search strategy. 4SS is a steepest descent strategy that compares at most 27 blocks to find the match, as opposed to evaluating all the blocks of the search window, which requires 225 comparisons.

All these elements are also present in a conventional Block Matching Algorithm (BMA). What differentiates VSBMA from BMA is that the sizes of the blocks in VSBMA change during the matching routine. All blocks have an initial size, but blocks that have a minimum matching error above a fixed error threshold are divided into smaller blocks and matched again. This process repeats until the error is below the threshold or the size of the blocks reach a fixed minimum size.

The proposed method is depicted in Figure 4.1 [Oliveira et al., 2015]. The first one is to calculate the displacement vectors with the variable size block matching algorithm. The second step is to convert these vectors into polar coordinates and build a histogram $\vec{h}$, where each bin represents an angle interval. The third step is to calculate an orientation tensor from $\vec{h}$, to serve as a condensed representation of the motion between a pair of frames. The tensors for each pair of frames and for each video sequence in the dataset are then accumulated and normalized with L2 Frobenius norm, so that it is possible to compare different video sequences regardless of their length or resolution.

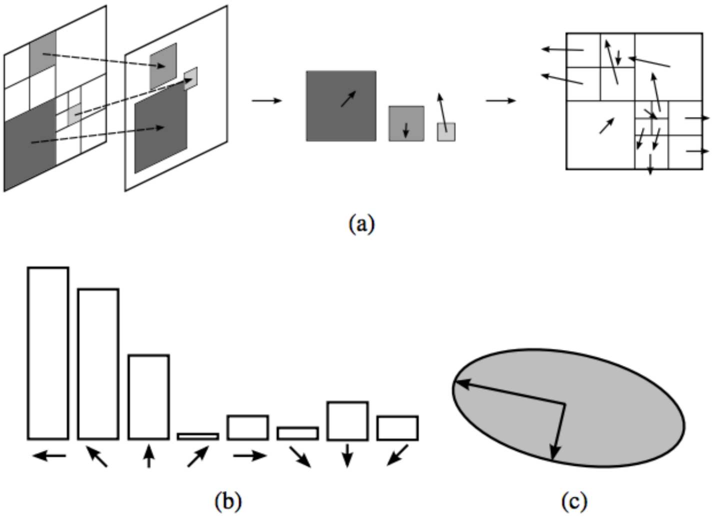


Figure 4.1: Tensor descriptor based on variable size block matching technique overview. (a) From left to right: Example of three blocks and their matches in the following frame yielded by the variable size block matcher. Displacement vectors obtained from the match. Vector map generated after matching all blocks in a frame. (b) Vectors accumulated into a histogram of directions. (c) Orientation tensor built based on such histogram. The ellipse is merely an illustration since generally tensor dimension is greater than two [Oliveira et al., 2015].

To validate this tensor descriptor, authors used the KTH video dataset [Schuldt et al., 2004].

4.4 Combination of Multiple Differential Operators

Different from the other approaches, Sad et al. [2013] presented an approach for motion description in videos using multiple band-pass filters which act as first order derivative estimator. Given that orientation tensors were promising to code features and describe videos, authors started to wonder if the band-pass filters used to extracted HOGs could have any impact in the quality of the final descriptor. The filter responses on each frame are coded into individual histograms of gradients to reduce their dimensionality. They are combined using orientation tensors. Motion description can even be enhanced using multiple filters with similar or overlapping frequency response.

The first step of this method is to apply a 5 × 5 Gaussian low-pass filter in all video frames. This is necessary to smooth the noisy highest frequencies. The noise reduction performed by the Gaussian filter proved itself to be relevant. However, in order to preserve significant motion information, the high frequency attenuation should not be too strong. Indeed, other filter sizes were used, resulting in lower performances. The subsequent gradient estimators are, thus, affected by this preprocessing.

A unidimensional filter is defined by a pair of impulse responses $(\mathrm{H}_a, \mathrm{G}_a)$, where $a \in \{1, 2, \cdots, f\}$ is the filter index, f is the number of filters for motion detection, G_a has high-pass frequency response, and H_a has low-pass response. Their multidimensional filter version is separable, having H_a and G_a as factors. To capture motion information, G_a is usually a derivative estimator with frequency response $\widetilde{\mathrm{G}}_a$. For multidimensional signals, H_a attenuates the noise on orthogonal directions. In this work, its frequency response $\widetilde{\mathrm{H}}_a$ is assumed to have some degree of complementarity in relation to $\widetilde{\mathrm{G}}_a$, in order to attenuate undesired correlated noise among the main axes.

The partial derivatives, or gradient, resulted from the application of a filter a on the j-th video frame I_j, at point p, are defined as the vector

$$\vec{g} = [dx_p^a \; dy_p^a \; dt_p^a]^T = \left[\frac{\partial I_j(p)}{\partial x} \; \frac{\partial I_j(p)}{\partial y} \; \frac{\partial I_j(p)}{\partial t}\right]^T,$$

or, equivalently, in spherical coordinates $\left[\rho_p^a \;\; \theta_p^a \;\; \psi_p^a\right]$ with $\theta_p^a \in [0,\ \pi]$, $\psi_p^a \in [0,\ 2\pi)$ and $\rho_p^a = ||\vec{g}||$. The derivatives indicate brightness variation that might be the result of local motion. The dx_p^a component is computed by firstly filtering the video in orthogonal directions Y and time using H_a, and afterwards in the main direction X using G_a. The same logic is used to obtain dy_p^a and dt_p^a.

Authors chose to apply wavelets as derivative estimators because of their widespread use and their behavior is well known. Note that the Gaussian low-pass filtering in image space followed by the application of a high-pass filter results in a band-pass frequency response.

Suppose the video frame j is uniformly subdivided into $\vec{x}$ and $\vec{y}$ directions by a grid with n_x and n_y non-overlapping blocks. Each block can be viewed as a distinct video varying in time. The smaller images result in gradient histograms $\vec{h}_j^a(c, r)$, $c \in [1, n_x]$ and $r \in [1, n_y]$, with better position correlation. The tensor for frame j, using derivative filter a, is then computed as the addition of all block tensors:

$$T_j^a(c,r) = \sum_{c,r} \vec{h}_j^a(c,r)\, \vec{h}_j^a(c,r)^T ,$$

which captures the uncertainty of the direction of the m-dimensional histogram vectors $\vec{h}_j^a(c,r)$ for the frame j. The tensor series becomes:

$$T^a = \sum_{j,c,r} \frac{T_j^a(c,r)}{||T_j^a(c,r)||},$$

where a is the derivative filter used, j is the frame index, and $(c,r) \in [1,n_x] \times [1,n_y]$ are the subimage coordinates.

The creation of the tensor follows the same idea explained above. The tensor of the frame I_j using the filter a is:

$$T_j^a = \vec{h}_j^a \vec{h}_j^{a\,T} ,$$

that carries the information of the gradient distribution of the j-th frame, computed using filter a. Individually, this tensor has the same information of $\vec{h}_j^a$.

The final video tensor descriptor for the derivative filter a is then $T^a/||T^a||$. It has the same number of elements of the version without image subdivision. The aggregation step captures the average motion of a video by $T^a = \sum_j T_j^a$.

To empirically reduce interframe brightness unbalance, the histogram of gradients $\vec{h}_j^a \in \mathbb{R}^{nb_\theta \cdot nb_\psi}$ might have all of its elements $h_{k,l}^a$ optionally adjusted to ${h_{k,l}^a}^\gamma$, with $\gamma = 0.5$. Here it was possible to see, again, the importance of a power normalization. This is applied only for the correlation filters to improve their performance.

Finally, it is proposed a concatenation of the individual tensors, computed for all the f filter pairs, to form the final descriptor for the input video:

$$T = \{T^1, T^2, \cdots, T^f\}.$$

The advantage of concatenation is that it preserves the motion information extracted by each filter, the same idea depicted in Sec. 6.2.2. The drawback is that the number of coefficients in the descriptor is multiplied by the number of filters f. In this work, the HOG has 128 bins yielding tensors with 8256 elements for a single filter. A video descriptor using four derivative filters has 33024 elements, slowing down SVM classification.

To validate this tensor descriptor, authors used three benchmark datasets: KTH

[Schuldt et al., 2004], UCF11 [Liu et al., 2009] and Hollywood2 [Marszałek et al., 2009]. They used the decomposition filter pairs of several wavelet families. The recognition rates differ slightly among them. The Daubechies wavelet family (db1, db2, db3, and so on) is adequate to separate the spectrum of dyadic bands, with an easy way to control the null moments.

To get even better results, the filters with better recognition rates were combined in a single linear filter. The proposal is to derive a pair $(\mathrm{H}_{f+1}, \mathrm{G}_{f+1})$ such that

$$|\widetilde{\mathrm{H}}_{f+1}(\omega)| = \sum_{a=1}^{f} |\widetilde{\mathrm{H}}_a(\omega)| \quad \text{and} \quad |\widetilde{\mathrm{G}}_{f+1}(\omega)| = \sum_{a=1}^{f} |\widetilde{\mathrm{G}}_a(\omega)|,$$

i.e., the magnitude response is the same as the sum of the magnitude of the $f > 1$ filters. Using db1, db3 and db7, for example, gives the $\mathrm{db}_{1,3,7}$ filter whose normalized high pass magnitude response is depicted in Figure 4.2. It alone gives 85.5% of recognition: slightly better than the average 85.1% of recognition of its components.

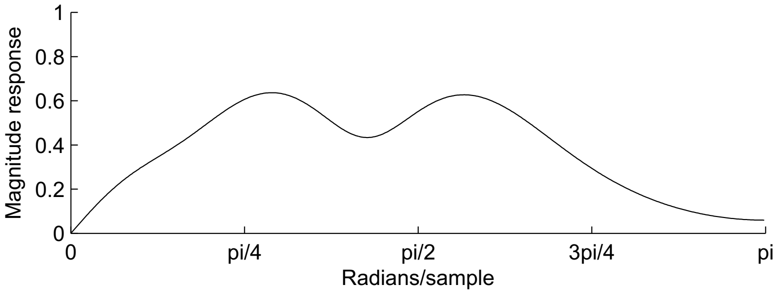


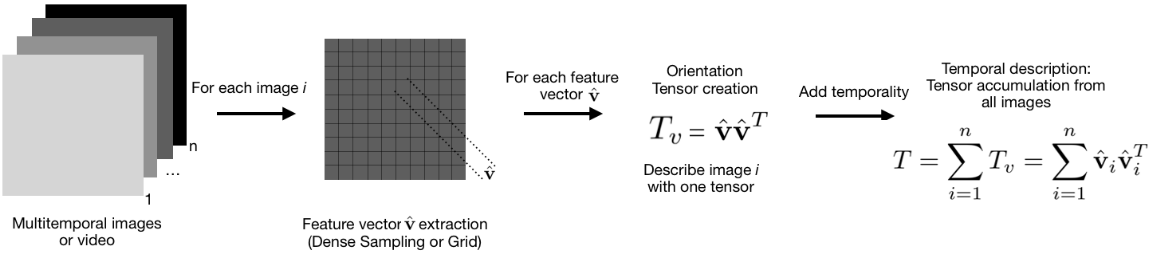
Figure 4.2: Transfer function of the correlation filter $\mathrm{db}_{1,3,7}$, modulated by a Gaussian filter.

Chapter 5

Proposed Framework

In this chapter, we describe our proposed method FASTensor: Features As Spatiotemporal Tensors.

Based on the mathematical framework presented in Chapter 3, an Orientation Tensor Framework for spatiotemporal description can be modeled as shown in Figure 5.1.



n
...
1
Multitemporal images or video
For each image i
Feature vector $\hat{\mathbf{v}}$ extraction (Dense Sampling or Grid)
$\hat{\mathbf{v}}$
For each feature vector $\hat{\mathbf{v}}$
Orientation Tensor creation
$T_v = \hat{\mathbf{v}}\hat{\mathbf{v}}^T$
Describe image i with one tensor
Add temporality
Temporal description: Tensor accumulation from all images
$T = \sum_{i=1}^{n} T_v = \sum_{i=1}^{n} \hat{\mathbf{v}}_i\hat{\mathbf{v}}_i^T$

Figure 5.1: An orientation tensor framework for temporal description created from the feature vector $\hat{\mathbf{v}}$ extracted with grid or dense sampling (Image made by the author).

FASTensor is composed by the following steps:

1. Feature vector extraction for each frame;

2. Orientation tensor creation for each frame;

3. Temporal description: tensor accumulation.

Section 5.1 presents the feature extraction step, comprehending shallow and deep features. Then, in section 5.2 we recap how we create an orientation tensor from a feature vector. Finally, in section 5.3 we summarize the proposed framework FASTensor.

5.1 Feature vector extraction

A feature vector is a representation of an image or a patch that simplifies the image by extracting useful information and throwing away extraneous information. Typically, a feature vector converts an image of size width × height × 3 (channels) to a feature vector / array of length n. This is the definition of handcrafted features.

In the context of deep learning a deep feature is the consistent response of a unit within a hierarchical model to an input, where this response contributes to the model's decision. A feature could be considered deeper than another if that unit depending on where the unit is positioned alongside the hierarchical structure of the model.

Handcrafted features used in our framework are presented in Section 5.1.1 and the deep features are presented in Section 5.1.2.

5.1.1 Handcrafted features

The experiments used three handcrafted features:

- Histogram of Gradients (HOG [Dalal and Triggs, 2005]);
- Optical Flow projected into Legendre Polynomials [Mota et al., 2012]; and,
- The concatenation of both of these features [Mota et al., 2014].

However, any feature could be used for this framework. Our choice was made in order to be able to compare with literature and state-of-the-art works.

5.1.2 Deep features

Transfer Learning using pretrained DNNs has also become a common practice for Computer Vision applications with the dawn of very large labeled datasets such as ImageNet[1] [Deng et al., 2009] and Pascal VOC[2] [Everingham et al., 2015]. Therefore we also tested the performance of tensors on the task of adding temporal consistency on feature vectors generated by activations in pretrained Convolutional Neural Networks (CNNs) [Krizhevsky et al., 2012], as these models are originally suited only for static images. Activations at the end of four distinct residual blocks in a pretrained ResNet were used as both raw features for classification and as inputs for FASTensors and compared at Section 7 and will be henceforth named as follows:

[1] `http://www.image-net.org/`

[2] `http://host.robots.ox.ac.uk/pascal/VOC/`

- ResNet-50 (1);
- ResNet-50 (2);
- ResNet-50 (3); and,
- ResNet-50 (4).

The pretrained DNNs were acquired and implemented using the PyTorch[3] framework and the torchvision pretrained model for the ResNet-50[4] [He et al., 2016].

5.2 Tensor creation

An orientation tensor T can be created from a vector using its transpose as follows:

$$T = (\hat{\mathbf{v}} - \mu)(\hat{\mathbf{v}} - \mu)^T, \tag{5.1}$$

where T is the tensor created from the product between the vector $\hat{\mathbf{v}}$ and its transpose $\hat{\mathbf{v}}^T$, with mean μ. Then, the tensor T is a matrix $n \times n$, with n being the size of the vector $\hat{\mathbf{v}}$. Therefore, it is possible to consider the orientation tensor as a measure of covariance of the elements of the vector. It is important to note that the sums of orientation tensors are also orientation tensors.

5.3 FASTensor: Features As Spatiotemporal Tensors

The framework can be used in videos or multi-temporal images with temporal dimension n. The orientation tensor T_v created from each feature vector $\hat{\mathbf{v}}$ will be accumulated for each image/frame i in order to represent the covariance of it, as in:

$$T = \sum_{i=1}^{n} T_v = \sum_{i=1}^{n} (\hat{\mathbf{v}}_i - \mu)(\hat{\mathbf{v}}_i - \mu)^T. \tag{5.2}$$

The features can be extracted with dense sampling or grid, or can also be learned. Then, the accumulation through time will provide the temporal description for the

[3] `https://pytorch.org/`

[4] `https://pytorch.org/docs/stable/torchvision/models.html`

video or for multi-temporal images. It is important to note that in each step, a normalization of the orientation tensor may be needed as the number of feature vectors from each image or frame could vary along time.

Figure 5.2 shows a two-dimensional example for the framework to better explain how the orientation tensor carries more information than the feature vector. Visually, instead of just having a vector representing the trend, we have the ellipsoid, carrying all the uncertainties and covariance measures of the features. Figure 5.2 shows an example of a movement tendency using a HOG feature of a person walking on a homogeneous background. With the orientation tensor, we can capture not only what happens in this scene, but how we begin to deform the ellipsoid so that it carries the whole tendency of the HOG. The geometric representation is made in three dimensions to facilitate the understanding of the problem. We will show in the following applications how this change can significantly improve video classification.

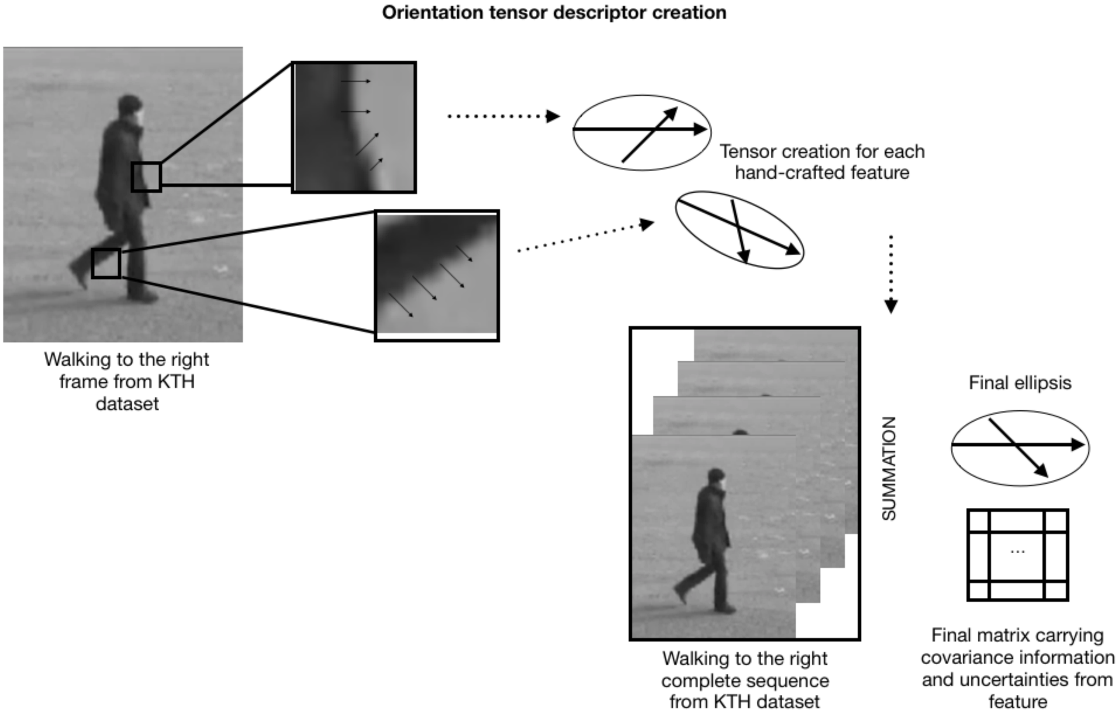

Orientation tensor descriptor creation
Tensor creation for each hand-crafted feature
Walking to the right frame from KTH dataset
SUMMATION
Final ellipsis
...
Walking to the right complete sequence from KTH dataset
Final matrix carrying covariance information and uncertainties from feature

Figure 5.2: Geometric example of the *FASTensor* framework for temporal description created from a feature vector $\hat{\mathbf{v}}$ (handcrafted) extracted with grid sampling. The final descriptor is a matrix $n \times n$, where n is the dimension of the feature, that carries the covariance and uncertainties from the features. The image is an example of walking action from KTH dataset [Schuldt et al., 2004] (Image made by the author).

Therefore, orientation tensors can be used as compact spatiotemporal representations, enabling dimension reduction and invariance according to the feature used to create them. They will capture the covariance information from the feature vector adding more information to the descriptor.

The limitations of this framework are that it carries only global information from each image and it is very dependable of the used feature. Thus, for describing several information from the same image sequences the method may not be eligible. For example, the ellipse depicted in Figure 5.2 can become a circle (or sphere in three dimensions), not carrying any main tendency information. That is, the tensor becomes isotropic.

5.3.1 Complexity Analysis

Given the mathematical framework presented in Section 3.1 and the *FASTensor* depicted in Figure 5.2, we can describe our proposed method with the following pseudo-algorithm:

```
(1)  Input: Video or Multi−temporal images
(2)  for each image i in Input:
(3)      // m features with dimension n
(4)      v[m] = feature_extraction();
(5)      for each feature vector j in v[m]:
(6)          T_j = matrix_multiplication(v[j]);
(7)          T_i = T_i + T_j;
(8)      normalize(T_i);
(9)      T_input = T_input + T_i;
(10) Output: T_input, a matrix of n x n
```

Figure 5.3: Pseudo-algorithm for the FASTensor framework.

The core of *FASTensor* is the computation of orientation tensor for each feature vector $\hat{\mathbf{v}}$. This is achieved with a matrix multiplication (line 6 of Figure 5.3), therefore, we have a complexity of $O(n^3)$, where n is the dimension of the feature vector $\hat{\mathbf{v}}$. In one frame we can have m feature vectors depending on the type of extraction. In the worst case, m is the number of pixels of the frame (dense sampling in lines 4 and 5 of Figure 5.3). Finally, the input has f frames (line 2 of Figure 5.3). Again, in the worst case, we use all frames from input.

The final complexity in the worst case for the *FASTensor* is $O(f \times m \times n^3)$, where f is the number of frames, m is the number of feature vector per frame and n is the dimension of the feature vector. In terms of time, as our method is feature dependent, we need to add the complexity of the feature extraction in $O(f \times m \times n^3)$.

Therefore, we have a complexity cubic growth in relation to the size of the feature, but a linear growth in relation to the number of features per frame and number of frames. So, we can reduce the computation time just by using less frames and other feature sampling instead of dense sampling.

5.4 Conclusion

In this chapter, we described the mathematical framework and presented a new method to spatiotemporal representation called *FASTensor* - Features As Spatiotemporal Tensors. With the mathematical framework we proved that orientation tensors can be used as compact spatiotemporal representations, enabling dimensionality reduction and invariance, according to the feature used to built them. That contributes to the first hypothesis of this dissertation.

In relation to computer complexity, the Features as Spatiotemporal Tensor (FASTensor) has a complexity of $O(f \times m \times n^3)$, where f is the number of frames, m is the number of feature vector per frame and n is the dimension of the feature vector. That is, we have a complexity cubic growth in relation to the size of the feature, but a linear growth in relation to the number of features per frame and number of frames.

The questions that remain are if the *FASTensor* can be used in different spatiotemporal tasks and if the framework can improve the accuracy of spatiotemporal tasks using handcrafted and deep features. This will be shown in the next chapters. We evaluated the *FASTensor* in three video classification applications: Human Action Recognition (Chapter 6), Video pornography (Chapter 7) and Cancer cell (Chapter 8). We chose three different types of video classification to validate the multipurpose of our framework and our three hypotheses presented in Chapter 1. The experiments used three handcrafted features: HOG, HOF and the concatenation of both (HOGHOF). We also used deep features extracted using CNN. We used SVM for classification and to compare the accuracy.

Chapter 6

Human Action Recognition

In this chapter, we present the experimental analysis we have conducted in order to evaluate the FASTensor in the Human Action Recognition problem. We have carried out experiments in order to address the following research questions:

- Can orientation tensors be used as compact spatiotemporal representations?
- Which is the best combination of orientation tensors to work with Human Action Recognition?

6.1 Experimental Setup

The experiments done in this chapter used three handcrafted features:

- Histogram of Gradients (HOG [Dalal and Triggs, 2005]);
- Optical Flow projected into Legendre Polynomials [Mota et al., 2012]; and,
- The concatenation of both of these features [Mota et al., 2014].

We used SVM as inference models for the classification tasks and compared it with the baselines using the accuracy metric. Feature extraction modules in this work were implemented using the skimage[1] framework, while SVM and validation procedure were coded using the sklearn[2] library. The core of the *FASTensor* approach uses the NumPy[3] and SciPy[4].

[1] https://scikit-image.org/

[2] http://scikit-learn.org/stable/

[3] http://www.numpy.org/

[4] https://www.scipy.org/

Our experiments used three benchmark datasets: KTH [Schuldt et al., 2004] (Figure 6.1), UCF11 [Liu et al., 2009] (Figure 6.2) and Hollywood2 [Marszałek et al., 2009] (Figure 6.3).

The KTH dataset [Schuldt et al., 2004] is a widely used standard dataset for human action recognition. It was introduced by Schuldt et al. [2004] and contains six types of human actions (walking, jogging, running, boxing, hand waving and hand clapping), which are performed by 25 actors in four different scenarios. All 2391 sequences have a spatial resolution of 160×120 pixels, a frame rate of 25 frames per second and about 4 seconds of duration. The background is static with some camera motion.

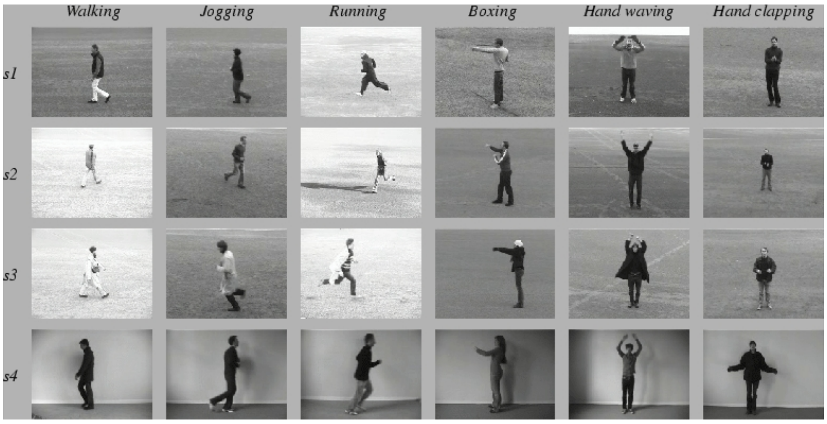


Figure 6.1: Action sample from KTH dataset [Schuldt et al., 2004].

The dataset was originally created to validate space-time features thus, it is simple compared with others as the background is static, the camera perspective is mostly frontal and remains still, although KTH dataset contains a certain degree of camera zooming. Therefore, the KTH dataset has been criticized for not being realistic sampling of actions in the real world [Liu et al., 2011]. Nevertheless, many researchers use them as a validation for newly proposed methods.

Liu et al. [2009] collected video sequences from YouTube and created a dataset of eleven actions, resulting in a total of 1168 sequences. The UCF11 (also known as UCF YouTube) dataset contains the following action categories: basketball shooting, biking/cycling, diving, golf swinging, horse back riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog.

The major criticism of KTH dataset was the lack of reality on their actions. In order to provide a more realistic benchmarking, Laptev and Pérez [2007] initiated an effort by creating a dataset with movie sequences. By extracting scenes from the movie "Coffee and Cigarettes", he provided a set of atomic actions, such as drinking and smoking. Each sequence is 30 to 200 frames long, with a mean of 70 frames.

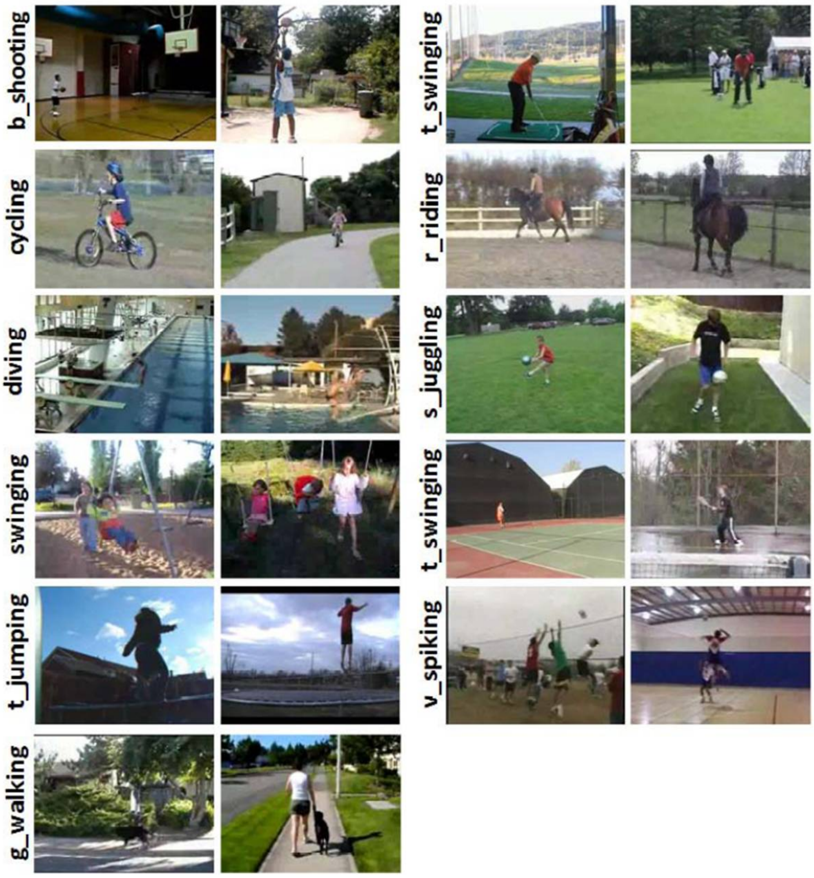


Figure 6.2: Action sample from UCF11 dataset [Liu et al., 2009].

Later, Laptev et al. [2008] created Hollywood1 dataset extracting eight different types of actions (answer phone, hug person, sit up, sit down, kiss, handshake, stand up and get out car) from several movies. Each sequence has about 50 to 200 frames with a spatial resolution of 240×500 pixels and a frame rate of 24 frames per second. Marszałek et al. [2009] subsequently created the Hollywood2 dataset by augmenting Hollywood1 with four more actions (drive car, eat, fight person and run).

Figure 6.3: Action sample from Hollywood2 dataset [Marszałek et al., 2009].

As the camera views are different from sequence to sequence, the background is cluttered, multiple subjects are present, occlusions occur very often, and the intra-class variability is large, making recognition hard [Liu et al., 2011].

The next sections describe our experiments with those three datasets.

6.2 Results and Discussion

Due to data and application specificities, several spatiotemporal content description approaches were proposed derived from the initial framework presented in Chapter 5. Those approaches are:

- Combining optical flow and histogram of gradients orientation tensors representation;
- Combining histogram of gradients tensors into blocks.

The following sections detail each one of these approaches.

6.2.1 Combining Optical Flow and Histogram of Gradients

In order to add more information to the tensor descriptor, in Mota et al. [2014], we proposed to concatenate the individual tensors, computed with the optical flow approximation (T^{OF}) and HOG (T^{HOG}), to form the final descriptor for the input video:

$$T = \{T^{OF}, T^{HOG}\}. \tag{6.1}$$

The hypothesis was that a simple concatenation could improve overall recognition. Despite the fact that other combination methods are possible, concatenation preserves the motion information extracted by each individual descriptor. The information of those descriptors are complementary and can improve the recognition rate.

It is important to note that the nature of these two tensors is different and, as such, needs to be equalized. One possible way is to use a power normalization in one of the descriptors. Experimentally, best results were obtained by normalizing the HOG tensor: the T^{HOG} descriptor in Equation 6.1 has all of its elements a_k adjusted to a_k^γ, $\gamma \ \in]0,1]$.

To validate this tensor descriptor, we used the KTH video dataset [Schuldt et al., 2004], UCF11 dataset [Liu et al., 2009] and the Hollywood2 dataset [Marszałek et al., 2009]. We followed the same protocol as the original paper [Schuldt et al., 2004; Liu et al., 2009; Marszałek et al., 2009]. Table 6.1 shows the recognition rates for several degrees for KTH dataset. The best recognition rate was 93.2% with polynomials of degree 5 (3670 elements) concatenated with a HOG of 128 bins (8256 elements).

Table 6.1: Recognition rates of KTH dataset for several degrees using a sliding window with dimensions 60×100 pixels concatenated with a HOG [Perez et al., 2012] of 128 bins with $\gamma = 1$ and confusion matrix for the best result (93.2%).

Degree	1	2	3	4	**5**	6	7	8	9	10
Rate (%)	92.6	92.1	92.5	92.7	**93.2**	92.1	92.0	91.3	91.8	90.6

Table 6.2 shows the recognition rates obtained with the proposed descriptor with a grid of 32×32 pixels and a HOG of 8×16 bins [Perez et al., 2012] for UCF11 dataset. A power normalization with $\gamma = 0.2$ was applied on the final HOG tensor. The best recognition rate was 72.7%, concatenating the HOG with polynomials of degree 13 (88410 elements).

Table 6.2: Concatenating the optical flow tensor descriptor with a grid of 32×32 pixels and a HOG of 8×16 bins [Perez et al., 2012] for UCF11 dataset. A power normalization was applied on the final HOG tensor with $\gamma = 0.2$.

Degree	1	2	3	4	5
Rate (%)	69.3	68.0	70.0	70.0	71.2
Degree	6	7	8	9	10
Rate (%)	70.7	71.8	71.5	71.4	72.2
Degree	11	12	**13**	14	15
Rate (%)	71.9	72.5	**72.7**	72.4	71.8

Table 6.3 shows the recognition rates for several degrees for Hollywood2 dataset. The best recognition rate was 40.3% concatenating the HOG with polynomials of degree 3 (820 elements).

Table 6.3: Concatenating the optical flow descriptor with a grid of 4×4 pixels and a HOG of 8×16 bins [Perez et al., 2012] for Hollywood2 dataset. A power normalization was applied on the final HOG tensor with $\gamma = 0.2$.

Degree	1	2	**3**	4	5	6	7	8	9	10
Rate (%)	39.5	39.9	**40.3**	40.2	40.3	39.8	40.1	39.7	39.9	40.3

6.2.2 Combining Multiple Histogram of Gradients

Due to the lose of locality information of the FASTensor when aggregating all frames, we proposed to analyze each tensor block individually.

Suppose the video frame f is uniformly subdivided in $\vec{x}$ and $\vec{y}$ directions by a grid with n_x and n_y non-overlapping blocks. Each block can be viewed as a distinct video varying in time. The smaller frames result in gradient histograms $\vec{h}_j^{i,j}$, $i \in [1, n_x]$ and $j \in [1, n_y]$, with better position correlation. The tensor for each block $b_{i,j}$ is then

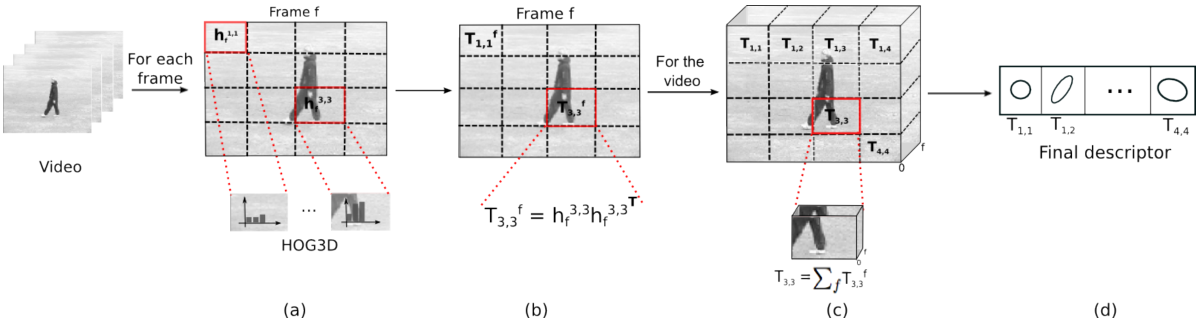


Figure 6.4: FASTensor modification. This is an example of a tensor descriptor obtained from a 4×4 pixels grid and each tensor block carries the information of the gradient distribution. The steps of this approach are: (a) Extract the histogram of gradients for each subdivision of each video frame; (b) Code HOG into orientation tensors for each block; (c) Accumulate the frame tensor from each block in order to model the temporal evolution of gradients; and (d) Concatenate each tensor block (Image made by the author).

computed as the addition of all block tensors throughout the video:

$$T_{i,j} = \sum_f \vec{h}_f^{i,j} \, \vec{h}_f^{i,j^T},$$

which captures the uncertainty of the direction of the m-dimensional vectors $\vec{h}_f^{i,j}$ for each block.

Thus, the final tensor descriptor is obtained by combining all blocks of the video. We proposed to concatenate the individual block tensors, to form the final descriptor for the input video:

$$T = \{T_{i,j}\}_{1 \leq i \leq n_x, 1 \leq j \leq n_y},$$

where the size of the final descriptor depends on the number of bins and subdivisions.

Figure 6.4 shows an example of a tensor descriptor obtained from a 4×4 pixels grid. For a better understanding of the method, the tensors are represented as an ellipsis (that is a second-order tensor). In our case, all tensors are m-order, where m is the number of bins of the histogram $(nb_\theta \cdot nb_\psi)$.

To validate this tensor descriptor, we used the KTH video dataset [Schuldt et al., 2004], UCF11 dataset [Liu et al., 2009] and the Hollywood2 dataset [Marszałek et al., 2009]. We followed the same protocol as the original papers [Schuldt et al., 2004; Liu et al., 2009; Marszałek et al., 2009]. The performance of this method on the KTH dataset is reported in Table 6.4, on the UCF11 dataset is reported in Table 6.5 and on the Hollywood2 dataset is reported in Table 6.6. For all datasets, the best configuration of parameters was Grid 4×4 pixels with a HOG of 8×16 bins.

Table 6.4: Recognition rate for KTH dataset for several parameter sets.

Parameters	Recognition Rate (%)
Grid 4×4 HOG 2×4	74.2
Grid 4×4 HOG 3×6	83.2
Grid 4×4 HOG 4×8	89.7
Grid 4×4 HOG 8×16	**92.5**
Grid 8×8 HOG 2×4	79.0
Grid 8×8 HOG 3×6	87.3
Grid 8×8 HOG 4×8	88.9

Table 6.5: Recognition rate for UCF11 dataset for several parameter sets.

Parameters	Recognition Rate (%)
Grid 4×4 HOG 2×4	57.3
Grid 4×4 HOG 3×6	63.9
Grid 4×4 HOG 4×8	70.1
Grid 4×4 HOG 8×16	**75.4**
Grid 8×8 HOG 2×4	58.9
Grid 8×8 HOG 3×6	64.7
Grid 8×8 HOG 4×8	71.0

Table 6.6: Recognition rate for Hollywood2 dataset for several parameter sets.

Parameters	Recognition Rate (%)
Grid 4×4 HOG 2×4	23.4
Grid 4×4 HOG 3×6	31.2
Grid 4×4 HOG 4×8	34.5
Grid 4×4 HOG 8×16	**40.3**
Grid 8×8 HOG 2×4	25.2
Grid 8×8 HOG 3×6	33.0
Grid 8×8 HOG 4×8	36.7

We explored two different approaches to use tensor descriptors:

- Combining optical flow and histogram of gradients orientation tensors representation;

- Combining histogram of gradients tensors into blocks.

We compared those approaches with the four approaches described in Chapter 4:

- Accumulating optical flow projections coded into orientation tensors;

- Accumulating histogram of gradients coded into orientation tensors;
- Variable size block matching algorithm associated with orientation tensors;
- Combining multiple gradient estimators.

A comparison of all works is presented in Table 6.7. We show the best results achieved by each work for three benchmark datasets: KTH, UCF11 and Hollywood2. We can see that the recognition rates for KTH dataset are not very discrepant, tending to be over 90.0%. Indeed, KTH dataset is the simplest with static background and a single view point, thus a simple descriptor could achieve good results. For more realistic datasets, the difference of recognition rates is more visible.

Table 6.7: A comparison of all works in three benchmark datasets: KTH, UCF11 and Hollywood2. Recognition rate in percentage for each of our works.

KTH		**UCF11**		**Hollywood2**	
Sad et al. [2013]	93.3	Mota et al. [2013]	75.4	Mota et al. [2013]	40.3
Mota et al. [2014]	93.2	Mota et al. [2014]	72.7	Mota et al. [2014]	40.3
Mota et al. [2013]	92.5	Perez et al. [2012]	68.9	Perez et al. [2012]	34.0
Perez et al. [2012]	92.0	Mota et al. [2012]	57.8	Mota et al. [2012]	15.0
Mota et al. [2012]	87.8				
Oliveira et al. [2015]	86.6				

With our first work [Mota et al., 2012] we were able to see how promising tensors were as motion descriptors. However, for a difficult dataset like Hollywood2, the descriptor failed. In fact, the summation of tensors will tend to be an isotropic tensor because there are a lot of different motions happening at the same time in the scenes. A single optical flow based descriptor is not enough to describe all important motion in the scenes.

The questions raised by this work led us to explore other features that could be coded into orientation tensors. We developed a tensor descriptor based on histogram of gradients [Perez et al., 2012]. We could see that HOG coded into orientation tensors carry richer information for the human action recognition problem. Moreover, it was possible to use orientation tensors for dimension reduction. An important question raised by this work was when normalizing the data. In subsequent works we always took it into account.

With Mota et al. [2014], it was possible to see that a concatenation of tensor descriptors improved overall recognition. Besides being very simple with a low-cost computation, concatenation preserves the motion information extracted by each individual descriptor. Other methods to combine feature remains an open question.

The results of Mota et al. [2013] indicated that the lost of locality of Perez et al. [2012] is prejudicial in more realistic datasets. We saw the work with individual tensors improved overall recognition. Then, we asked ourselves if other methods could use individual tensor.

The results of Sad et al. [2013] indicated that the use of multiple filters is promising for the problem of motion description. Further studies are needed to improve filter correlation. Moreover, an analysis in other datasets is needed.

We also compared our tensor approaches with the state-of-the-art shallow methods using three benchmark datasets: KTH, UCF11 and Hollywood2.

Table 6.8 summarizes the results for KTH dataset for global appearance and motion based descriptors.

Table 6.8: Recognition rates in percentage for KTH dataset. *Indicates leave-one-out protocol.

Flow/Gradient		Hybrid		Tensor	
Imtiaz et al. [2011]	97.0*	Schindler and Van Gool [2008]	92.5	Kim et al. [2007]	95.3
Solmaz et al. [2012]	92.0				
Rodriguez et al. [2008]	88.6				
Laptev et al. [2007]	72.0	Our approaches			
		Sad et al. [2013]	93.3		
		Mota et al. [2014]	93.2		
		Mota et al. [2013]	92.5		
		Perez et al. [2012]	92.0		
		Mota et al. [2012]	87.8		
		Oliveira et al. [2015]	86.6		

For KTH, we see that the best result is using optical flow and RANdom SAmple Consensus (RANSAC) achieving 97.0%. However, this result is reached with a leave-one-out protocol. Kim et al. [2007] achieves 95.3% by using tensor canonical correlation analysis. Thus, we cannot consider them as a fair comparison. The best result using the same protocol as the baseline by Schuldt et al. [2004] and a SVM classifier is 92.5% for Schindler and Van Gool [2008].

Tensor descriptor falls in the same category as the descriptors of these works, global appearance and motion based descriptors. Comparing to the best result that uses the same protocol as ours, only the optical flow based tensor descriptor [Mota et al., 2012] achieved a lower recognition rate. It is interesting to note that the hybrid approach [Mota et al., 2014] improved overall recognition rate. However, combining multiple differential estimators presented the best result and showed how promising this technique was [Sad et al., 2013].

Works about global descriptors do not provide results for difficult datasets such as UCF11 and Hoollywood2. Only the approach of Rodriguez et al. [2008] is evaluated on UCF Sports dataset, achieving 69.0%. We see that tensor HOG descriptor achieved a similar recognition rate (68.9%) as this work. Moreover, both combinations of tensors [Mota et al., 2014, 2013] improved this recognition rate. In order to explore the importance of multiple gradient estimators, we needed to further analyze Sad et al. [2013] in challenging datasets.

When compared to codebook based methods, the handcrafted descriptors tend to lose in more realistic datasets. Table 6.9 summarizes the best published results on KTH dataset for bag-of-feature based approaches. The best result is 98.2% [Sadanand and Corso, 2012] using an idea of semantic transfer. Using the classical framework, the best result is achieved by Kobayashi and Otsu [2012] (95.6%) and is very close to the result of Wang et al. [2013] (95.3%). The first uses co-occurrence histograms of the space-time dense sampled gradient. The second combines trajectories, HOG, HOF and MBH modeled along a space time grid. Just by changing the BoF framework to take into account SPD matrices, Faraki et al. [2014] were able to improve the recognition rate of Wang et al. [2013] achieving 97.4%. The common idea of all three is to sample features along space-time axis. It is interesting to note that the recognition rate for the tensor approach by Kihl et al. [2013] (94.2%) is very competitive.

Tensor descriptors presented lower recognition rates than these bag-of-features techniques. Nevertheless, it is interesting to note that the best result is competitive [Sad et al., 2013] using only histogram of gradients information, while the best codebook based methods combine various features. The tensor approach by Kihl et al. [2013] indicated the importance of local patch methods and showed us that tensors could really improve in this level.

Table 6.9: Recognition rates in percentage for KTH dataset using bag-of-feature based methods and our approaches. *Indicates leave-one-out protocol.

Local descriptors		Trajectories		Relationship Modeling	
Kobayashi and Otsu [2012]	95.6	Faraki et al. [2014]	97.4	Sadanand and Corso [2012]	98.2
Minhas et al. [2010]	94.8*	Wang et al. [2013]	95.3	Gilbert et al. [2011]	94.5
Le et al. [2011]	93.9	Wang et al. [2011]	94.2	Kovashka and Grauman [2010]	94.5
Shao and Gao [2010]	93.8				
Zhen and Shao [2012]	92.2				
Laptev et al. [2008]	91.8				
Kläser et al. [2008]	91.4	Tensor			
		Kihl et al. [2013]	94.2		
		Our approaches			
		Sad et al. [2013]	93.3		
		Mota et al. [2014]	93.2		
		Mota et al. [2013]	92.5		
		Perez et al. [2012]	92.0		
		Mota et al. [2012]	87.8		
		Oliveira et al. [2015]	86.6		

For UCF11, the best results are by Wang et al. [2013] with 85.4% and by Le et al. [2011] achieving 75.8%. We see that for more challenging datasets, the best results are still with Sadanand and Corso [2012] and Wang et al. [2013]. Note that our best result in UCF11 is 75.4% for Mota et al. [2013] which models the temporal evolution of HOG with orientation tensors. Thus, using only one type of feature, we achieved a recognition rate very close to a bag-of-feature technique.

A similar result is achieved for Hollywood2 dataset. Hollywood2 dataset is the most challenging, and has been collected from Hollywood movies. Table 6.10 summarizes the recognition rates for Hollywood2 dataset.

Table 6.10: Recognition rates in percentage for Hollywood2 dataset using bag-of-features based methods and our approaches.

Local descriptors		Trajectories		Relationship Modeling	
Le et al. [2011]	53.3	Jain et al. [2013]	62.5	Gilbert et al. [2011]	50.9
Kobayashi and Otsu [2012]	47.7	Wang et al. [2013]	59.9		
Laptev et al. [2008]	45.2				
Kläser et al. [2008]	43.7	Tensor			
		Vig et al. [2012]	59.5		
		Kihl et al. [2013]	57.6		
		Our approaches			
		Mota et al. [2013]	40.3		
		Mota et al. [2014]	40.3		
		Perez et al. [2012]	34.0		
		Mota et al. [2012]	15.0		

The best result is 62.5% found by Jain et al. [2013]. They used the same features as Wang et al. [2013] combined with a new motion descriptor called Divergence-Curl-Shear (DCS). Moreover, they did not use the classical BoF method, but rather the Vector of Locally Aggregated Descriptor (VLAD) method. Note that the result of its improvement, the Vector of Locally Aggregated Tensor (VLAT), achieved a competitive recognition rate (57.6%) using only two types of features. The other tensor approach by Vig et al. [2012] also achieved good recognition rate (59.5%).

The difference between our recognition rates and the state-of-the-art is very high, which indicates the power of bag-of-features based techniques and the importance of interest points (local descriptors). However, it is remarkable that our best results [Mota et al., 2013] and [Mota et al., 2014] (40.3%) are close to the combination of interest points, HOG and HOF [Kläser et al., 2008].

All those results were achieved with other shallow approaches. When compared to Deep Learning-based techniques, those three datasets are already deprecated. Table 6.11 shows the best results for KTH, UCF11 and Hollywood2 using state-of-the-art deep learning-based approaches.

Literature in Human Action Recognition has moved to more difficult datasets as HMDB51 [Kuehne et al., 2011] and Sports-1million [Karpathy et al., 2014]. Those

Table 6.11: Best results for KTH, UCF11 and Hollywood2 using state-of-the-art deep learning-based approaches.

Dataset	**Recognition Rate**
KTH [Zhou et al., 2016]	98.67%
UCF11 [Peng et al., 2014]	93.77%
Hollywood2 [Liu et al., 2017]	78.50%

datasets have more heterogeneous actions, more videos and even semantic context as smiling and laugh. Thus, for human action recognition we found a barrier and we are not able to compete with DL methods.

6.3 Conclusion

In this chapter, we described our tensor descriptors for Human Action Recognition, based on global appearance and motion, and compared with state-of-the-art shallow and deep learning methods.

The main hypothesis for all works described was that it was possible to create a simple descriptor using orientation tensors that could maintain balance between size, computer complexity and recognition rate. All those descriptors depend only on the video itself, not requiring any recomputation of the previously computed descriptors after the addition of new videos and/or new action categories to the dataset.

The main contribution of these works was to show the power of orientation tensors as video descriptors. By a simple coding of features into orientation tensors and accumulation of them, we were able to develop an efficient and simple temporal descriptor. Therefore, our three different approaches to use orientation tensors as video descriptors confirmed our first hypothesis.

However, for human action recognition we found a barrier. The big limitation of our method is the number of actions that can be performed in one scene. Furthermore, deep learning-based approaches is leading the human action recognition task nowadays and we are not able to compete with them.

Even so, we could consolidate the *FASTensor* as a framework for spatiotemporal description. Henceforth, we chose to use a more simple version of the framework to apply in other applications. Spatiotemporal representation is still an open problem, there are other applications that a method with low computer complexity and where deep learning is not suitable can be applied.

Chapter 7

Pornography Classification

In this chapter, we present the experiments that we performed to validate the FASTensor for Video Pornography classification. We have carried out experiments in order to address the following research questions:

- Can orientation tensors be used as compact spatiotemporal representations?
- Can the same representation be used in different classification tasks?
- How do raw features from different natures (shallow and deep) behave in video description?

Compared to Human Action Recognition, Pornography is less straightforward to define than it may seem at first, since it is a high-level semantic category, not easily translatable in terms of simple visual characteristics. Pornography can be defined as "any explicit sexual matter with the purpose of eliciting arousal" [Short et al., 2012]. Though it certainly relates to nudity, pornography is a different concept: many activities which involve a high degree of body exposure (swimming,boxing, sunbathing, *etc.*) have nothing to do with it. That is why systems based on skin detection often accuse false positives in contexts like beach shots or sports [Avila et al., 2013]. Moreover, few works tackle the problem using spatiotemporal representations.

We can divide works about pornography detection and/or classification in the main categories: Skin Detection, bag-of-feature (BoF) based approaches, Spatiotemporal features, Audio Features and Deep Learning-based approaches. As showed by Avila et al. [2013], skin detection is not a good technique for pornography detection as it accuses a great number of false positives. As we are mainly interested in visual information, we will focus on the image-based categories to compare *FASTensor* with the literature results in Pornography Classification.

In Chapter 2, we presented the BossaNova descriptor created by Avila et al. [2013] and its extension created by Caetano et al. [2016]. Both are BoF-based approaches. Moreira et al. [2016] introduced TRoF, a spatiotemporal descriptor that uses bag-of-feature associated with SURF [Bay et al., 2008]. Finally, Wehrmann et al. [2018] presented a CNN-based approach for pornography classification and Perez et al. [2017] added motion information to the CNN using a late fusion of optical flow information.

Even being a very semantic task, therefore, very complicated application, pornography classification is suitable for *FASTensor* as we have one main action occuring in the scene. Thus, the probability of the orientation tensor to become isotropic is inferior compared to the Human Action Recognition problem.

7.1 Experimental Setup

The experiments done for this application used three handcrafted features:

- Histogram of Gradients (HOG [Dalal and Triggs, 2005]);
- Histogram of Optical Flow (HOF [Dalal et al., 2006]); and,
- The concatenation of both of these features (HOGHOF [Laptev et al., 2008]).

Transfer Learning using pretrained DNN has also become a common practice for Computer Vision applications with the dawn of very large labeled datasets such as ImageNet[1] [Deng et al., 2009] and Pascal VOC[2] [Everingham et al., 2015]. Therefore we also tested the performance of tensors on the task of adding temporal consistency on feature vectors generated by activations in pretrained CNN [Krizhevsky et al., 2012], as these models are originally suited only for static images. Activations at the end of four distinct residual blocks in a pretrained ResNet were used as both raw features for classification and as inputs for *FASTensors*. Those deep features will be henceforth named as follows:

- ResNet-50 (1);
- ResNet-50 (2);
- ResNet-50 (3); and,
- ResNet-50 (4).

[1] `http://www.image-net.org/`

[2] `http://host.robots.ox.ac.uk/pascal/VOC/`

The pretrained DNNs were acquired and implemented using the PyTorch[3] framework and the torchvision pretrained model for the ResNet-50[4] [He et al., 2016].

We used SVM as inference models for the classification tasks and compared it with the baselines using the accuracy metric. Feature extraction modules in this work were implemented using the skimage[5] framework, while SVM and validation procedure were coded using the sklearn[6] library. The core of the *FASTensor* approach uses the NumPy[7] and SciPy[8] libraries.

We evaluated the *FASTensor* on the Pornography-800 Dataset, created by Avila et al. [2013]. This dataset contains nearly 80h of 400 pornographic and 400 non-pornographic videos. Concerning the pornographic material, the dataset is very assorted, including both professional and amateur content (Figure 7.1). Moreover, it depicts several genres of pornography, from cartoon to live action, with diverse behavior and ethnicity. With respect to non-pornographic content, they are general-purpose video networks, with difficult cases like sumo, swimming, beach scenarios (*i.e.*, words associated to skin exposure).

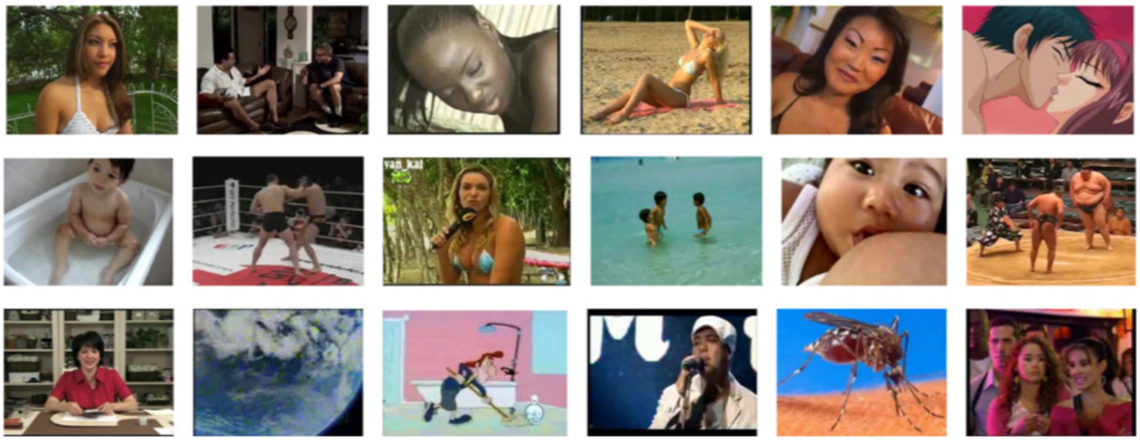


Figure 7.1: Examples from Pornography-800 dataset. The dataset comprises pornographic and non-pornographic videos, very assorted, including both professional and amateur content [Avila et al., 2013].

In order to transform the pornography classification into a more challenging problem, Moreira et al. [2016] extended the Pornography-800 dataset. The Pornography-2k dataset (Figure 7.2) comprises nearly 140 hours of 1,000 pornographic and 1,000 non-

[3] https://pytorch.org/
[4] https://pytorch.org/docs/stable/torchvision/models.html
[5] https://scikit-image.org/
[6] http://scikit-learn.org/stable/
[7] http://www.numpy.org/
[8] https://www.scipy.org/

pornographic videos, which varies from six seconds to 33 minutes. As we obtained very good results on the Pornography-800 dataset, we also evaluated the *FASTensor* on the Pornography-2k. Moreover, because the dataset has very long videos, we would like to test the robustness of the method in relation to the high number of frames.

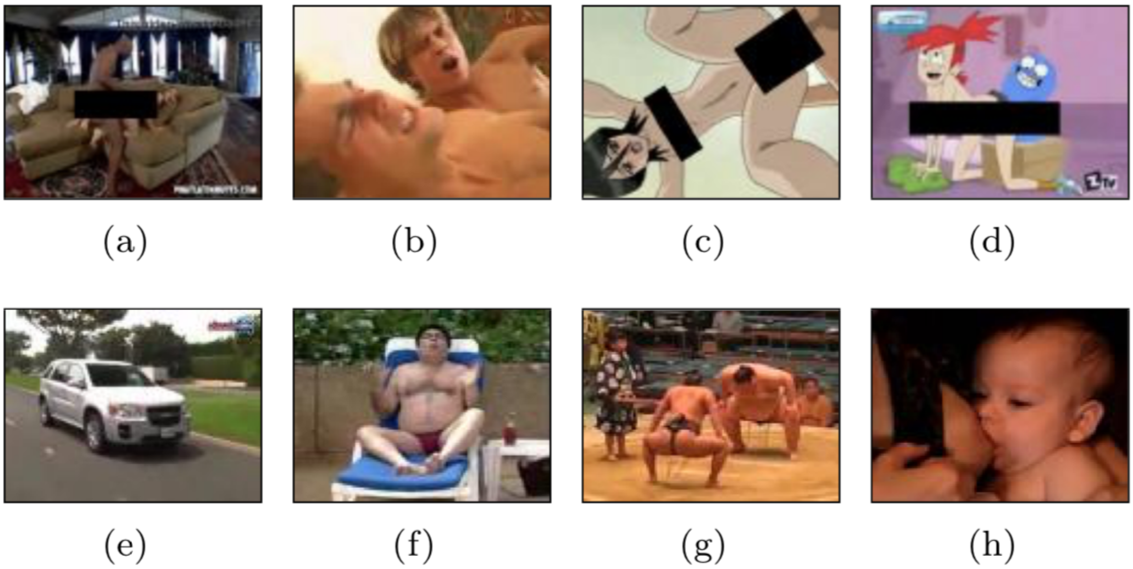


Figure 7.2: Examples from Pornography-2k dataset. The top row shows representative sensitive content, including pornographic cartoons. The black censor bars were added in the understanding that it is a very sensitive material. The bottom row shows non-pornographic content, emphasizing examples with non-sexual skin exposure [Moreira et al., 2016].

7.2 Results and Discussion

The baseline results for Pornography-800 dataset are presented in Table 7.1, extracted from Avila et al. [2013]. They preprocessed the dataset by segmenting videos into shots. On average there are 20 shots per video. A key frame (middle frame) is selected to summarize the content of the shot into a static image. As low-level local descriptor, they employed HueSIFT [Van de Sande et al., 2010], a SIFT variant including color information. The 165-dimensional HueSIFT descriptors are extracted densely every six pixels. The same vocabulary M constructed by k-means clustering algorithm, with M fixed as 256, is used for the standard BoF and the BossaNova method [Avila et al., 2013].

For classification, they used a 5-fold cross-validation to tune the best c parameter for a SVM classifier. The c parameter tells the SVM optimization how much you want to avoid misclassifying each training example. They reported the image classification performance by using the Mean Average Precision (MAP), and the video classification by accuracy rate, where the final video label is obtained by majority voting over the images. It is interesting to note that for both reported methods, the video classification scores are inferior to the image classification scores. That can be explained by the fact that some pornographic videos have the additional difficulty of having very few shots with pornographic content (typically one or two takes among several dialog shots or cut scenes).

As our framework is video based, we compared our results with the accuracy from the baseline. We used the same division protocol from the baseline as SVM protocol. We compared three handcrafted features vastly used in video description: HOG, HOF and the combination of both HOGHOF. We used a dense sampling extraction with the fixed number of bins, HOG with sixteen bins, HOF with eight bins, and the HOGHOF with twenty-four bins. The results comparing the baseline, handcrafted features and the *FASTensor* are depicted in Table 7.1.

We can observe that the best result from the baseline for accuracy, 89.5% ± 1 using BossaNova, is inferior compared to the best result from *FASTensor* 93.28% ± 0.36 using the HOGHOF coded into an orientation tensor. Even if we compare with HOG and HOF we see that the *FASTensor* achieved competitive results. When analyzing deep features, we see that for the third ResNet-50 layer it was possible to achieve 96.45% ± 0.24, a statistically higher accuracy than all baselines. Therefore, we see that our proposed method can be used for video description in pornography classification achieving high accuracy for the Pornography-800 dataset.

Here, it is interesting to discuss the failure cases of our descriptors. We first investigated the misclassified non-pornographic videos. Most parts correspond to *difficult* non-porn: Sequences of children being bathed, beach sequences and fights. The analysis of the pornographic videos revealed that descriptors have difficulty when the videos have very poor quality, when the video has few pornography sequences and when there are various camera angle changes in pornography sequences.

Given that the *FASTensor* is an interesting method for video description, we need to analyze how it improved the handcrafted features. Figure 7.3 shows the graph bar with the confidence interval of 95% to facilitate the visualization. It is easy to see that for all handcrafted features (HOG, HOF and HOGHOF), the difference in the obtained accuracy is statistically significant. While comparing with BossaNova, our results are competitive for HOG and HOF features and higher for HOGHOF *FASTensors*. Therefore, we have the proof of context for the framework, the *FASTensor* carries more information than the raw feature vector, producing a better and more discriminative

Table 7.1: Baseline for the Pornography-800 dataset using Standard Bag-of-features and BossaNova. Compared results from handcrafted features and the *FASTensor* for the Pornography-800 dataset. We used a dense sampling extraction with fixed number of bins, HOG with sixteen bins (eight for each frame in a pair), HOF with eight bins, and HOGHOF with twenty-four bins. Results for the *FASTensors* followed by † represent accuracies that were significantly improved by the proposed approaches in comparison with using Raw Features.

Method		Accuracy (%)
Baselines	**BoF**	83 ± 3
	BossaNova [Avila et al., 2013]	89.5 ± 1
	Caetano et al. [Caetano et al., 2016]	92.4 ± 1
	TRoF [Moreira et al., 2016]	95 ± **
	ACORDE-50* [Wehrmann et al., 2018]	94.8 ± 2
	ACORDE-101* [Wehrmann et al., 2018]	95.6 ± 1
	Perez et al. [Perez et al., 2017]	97.9 ± 1.5
Raw Features	**HOG**	82.16 ± 0.54
	HOF	77.20 ± 0.31
	HOGHOF	88.12 ± 0.56
	ResNet-50 (1)	91.34 ± 0.28
	ResNet-50 (2)	92.25 ± 0.14
	ResNet-50 (3)	94.73 ± 0.73
	ResNet-50 (4)	94.75 ± 0.38
FASTensors	**HOG**	85.32 ± 0.31 †
	HOF	84.18 ± 0.18 †
	HOGHOF	93.28 ± 0.36 †
	ResNet-50 (1)	93.50 ± 0.12 †
	ResNet-50 (2)	93.49 ± 0.14 †
	ResNet-50 (3)	**96.45 ± 0.24** †
	ResNet-50 (4)	96.25 ± 0.25 †

video descriptor. Moreover, with a feature richer in motion information, as the combination of HOGHOF, our framework can achieve better results with less computer work (it does not need any clustering or other techniques) and is very compact. BossaNova has 2560 codewords while the HOGHOF *FASTensor* is a 24 × 24 matrix.

Deep features extracted from the output of all four residual blocks of ResNet-50 [He et al., 2016] were significantly improved with the use of *FASTensors*, when compared to classification using simple raw activations from these networks, validating our method as a useful tool to insert temporal coherence in deep representations. ResNet-50 (3) and ResNet-50 (4) features coupled with *FASTensors* yielded better results than even the best shallow [Moreira et al., 2016] and deep [Wehrmann et al., 2018] baselines previously described in the literature. Wehrmann et al. [2018] added

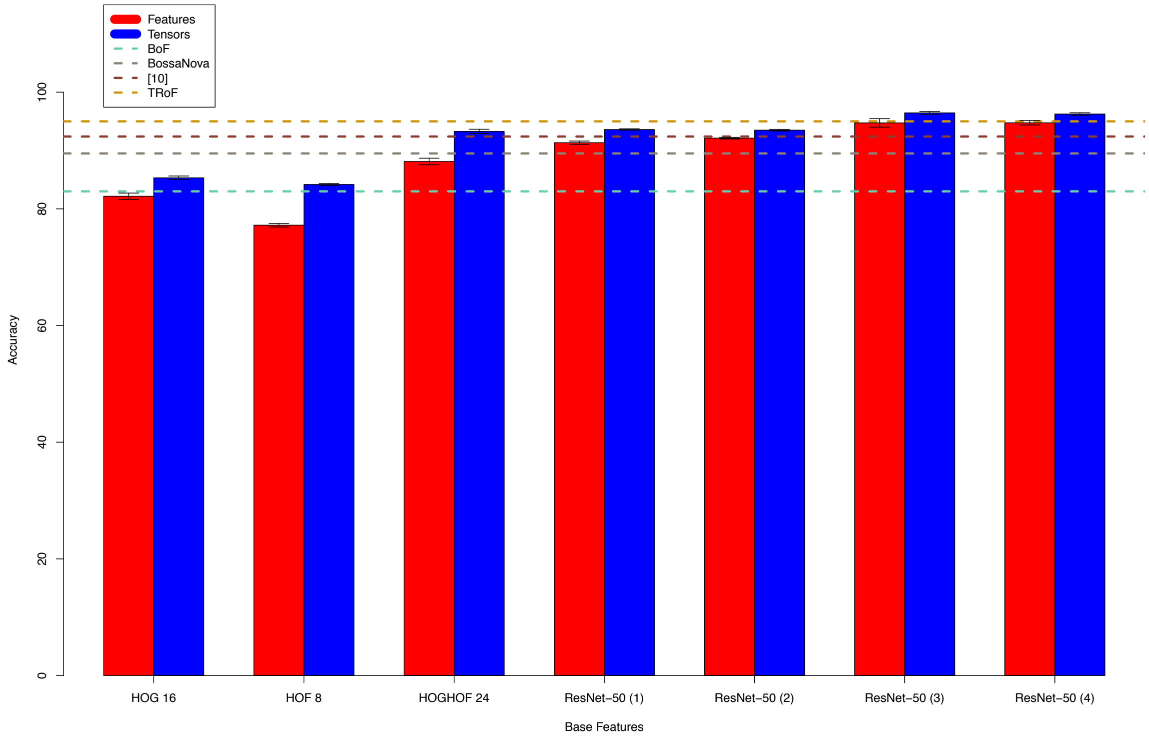


Figure 7.3: *FASTensor* comparison with handcrafted and deep features for the Pornography-800 dataset. Dashed lines show baseline accuracies in the dataset (Image made by the author).

temporal coherence to static image feature extraction from the video frames using an LSTM coupled with the ResNet feature extractors, which included the ResNet-50, ResNet-101 and ResNet-152. The best results were using LSTMs. With these results we can therefore confidently assert that *FASTensors* comprise the new state-of-the-art for video classification in the Pornography-800 dataset without using neural networks deep archtectures and is statistically equivalent to the state-of-the-art for the deep learning approach [Perez et al., 2017].

Given the best configuration for the Pornography-800 dataset, we evaluated the *FASTensor* on the Pornography-2k dataset using the handcrafted feature HOGHOF with twenty-four bins and all four deep features ResNet-50. Table 7.2 shows our accuracies compared with the baselines.

For the Pornography-2k, we can observe that the best result from the baseline for accuracy, 96.4% using two CNN's, is superior compared to the best result from *FASTensor* 93.36% ± 0.23 using the fourth ResNet-50 layer coded into an orientation tensor. However this is a very impressive result, as it is competitive, more simple and less time consuming. It is interesting to note that the state-of-the-art method,

Table 7.2: Baseline for the Pornography-2k dataset using Standard Bag-of-features and BossaNova. Compared results from handcrafted features and the *FASTensor* for the Pornography-2k dataset. We used a dense sampling extraction for a HOGHOF with twenty-four bins. Results for the *FASTensors* followed by † represent accuracies that were significantly improved by the proposed approaches in comparison with using Raw Features.

Method		Accuracy (%)
Baselines	**TRoF [Moreira et al., 2016]**	95.6 ± **
	Perez et al. [Perez et al., 2017]	**96.4 ± ****
Raw Features	**HOGHOF**	60.88 ± 0.44
	ResNet-50 (1)	82.7 ± 0.69
	ResNet-50 (2)	88.12 ± 0.08
	ResNet-50 (3)	91.59 ± 0.20
	ResNet-50 (4)	91.28 ± 0.15
FASTensors	**HOGHOF**	65.44 ± 0.31 †
	ResNet-50 (1)	89.35 ± 0.26 †
	ResNet-50 (2)	90.25 ± 0.03 †
	ResNet-50 (3)	93.01 ± 0.14 †
	ResNet-50 (4)	**93.36 ± 0.23** †

Perez et al. [2017] argued about the importance of adding motion information to the description. In fact, our proposed framework argues the same thing when talking about temporal information. Moreover, all *FASTensors* approaches were better than the raw features in this dataset. Therefore, we can conclude that orientation tensors can improve the accuracy of spatiotemporal tasks using handcrafted and deep features, which validates our hypothesis.

7.3 Conclusion

In this chapter, we evaluated the *FASTensor* for the Video Pornography classification task using two benchmark datasets: The Pornography-800 and the Pornography-2k.

We could evaluate the performance using handcrafted and deep features. Our results could assert that *FASTensor* comprises the new state-of-the-art for video classification in the Pornography-800 dataset without using neural networks deep archtectures and it is statistically equivalent to the state-of-the-art for the deep learning approach. For the Pornography-2k, *FASTensor* achieved competitive results using the fourth ResNet-50 layer compared to the state-of-the-art that uses two CNNs to achieve the best accuracy.

The main contribution of this application was to show that it was possible to validate the three hypotheses of this thesis: orientation tensors can be used as spatiotemporal representation, they improve the accuracy of raw features (handcrafted and deep) and are robust to diffent tasks, as we already show that it can also be used in Human Action Recognition.

The next step is to evaluate the *FASTensor* in a non-person task: The Cancer Cell classification that we discuss on the next chapter.

Chapter 8

Cancer Cell Classification

In this chapter, we present the new multitemporal image dataset Melanoma Cancer Cell (MCC) and the experiments using the FASTensor (Section 8.1).

We have carried out experiments in order to address the following research questions:

- Can orientation tensors be used as compact spatiotemporal representations?
- Can the same representation be used in different classification tasks?
- Is the *FASTensor* suitable for supervised classification with small datasets?

8.1 Experimental Setup

The experiments from this chapter used three handcrafted features:

- Histogram of Gradients (HOG [Dalal and Triggs, 2005]);
- Histogram of Optical Flow (HOF [Dalal et al., 2006]); and,
- The concatenation of both of these features (HOGHOF [Laptev et al., 2008]).

We used SVM as inference models for the classification tasks and compared it with the baselines using the accuracy metric. Feature extraction modules in this work were implemented using the skimage[1] framework, while SVM and validation procedure were coded using the sklearn[2] library. The core of the *FASTensor* approach uses the NumPy[3] and SciPy[4].

[1] `https://scikit-image.org/`

[2] `http://scikit-learn.org/stable/`

[3] `http://www.numpy.org/`

[4] `https://www.scipy.org/`

Even with technological advance, cancer still remains one of the most lethal diseases in the world [Bradbury, 2007]. The development of therapeutic strategies includes the discovery of new drugs through *in vitro* tests, preclinical studies in animals and clinical trials [Goodspeed et al., 2016].

In vitro assays with tumor-derived cell lines are performed outside the living organisms and have considerably increased the control and quantification of parameters in the cancer study, allowing the evaluation of even a single cell. A wide range of methodologies can be performed to study the effect of compounds as well as cell behavior, including viability testing, colony formation, cell-cell and matrix cell interactions, invasion and migration [Young, 2013]. These tests are routinely used for the discovery of new anti-cancer drugs since 1950, and thereafter, these studies are considered economically viable and translationally relevant [Gillet et al., 2013].

Melanoma is a highly aggressive type of cancer and therapy becomes a challenge once the cancer cells metastasize and colonize other tissues. The metastatic process includes cellular invasion through the extracellular matrix, intravasation into blood vessels and colonization of new sites after leaving the blood vessel [Martin et al., 2013; Villareal et al., 2018]. Here, we analyzed cell movement in B16F10 cell line, which is highly invasive subpopulation of a murine melanoma. Indeed, the great ability of these cells to colonize other sites comes from specific characteristics, among them those related to movement such as migration and invasion [Hart, 1979]. Movement analysis in cells exposed to a treatment may be directly related to its migration and invasion profile being of great importance during the development of new anti-cancer drugs.

Currently, there are some types of computational image processing techniques for cell migration analysis, combing live-cell microscopy and image processing algorithms [Masuzzo et al., 2016]. The most common method to quantify and characterize motion estimation is the Cell tracking, that has been performed manually, using the low-level preprocessing followed by segmentation, postprocessing of the candidate objects and feature extraction that supplies a latter stage of data analysis.

However, even for those methods that include image processing algorithms, the dataset is not open which makes it impossible to compare different techniques.

Therefore, one of the contributions of this work is a new open multitemporal image dataset: The Melanoma Cancer Cell dataset. This dataset provides better understanding of the cancer cell migration and anti-migration promoted by specific drugs [Decaestecker et al., 2007], classifying in treated and untreated cell, being possible to characterize phenotypic and morphologic drug effects [Ramnath and Creaven, 2004]. Therefore, allowing to elucidate some intrinsic biological mechanisms of cancer cell, particularly understanding the tissue invasion and metastases formation.

This dataset has two conditions of long-term culture of metastatic murine melanoma B16F10 cells in Roswell Park Memorial Institute (RPMI) medium (supplemented with 10% Fetal Bovine Serum (FBS), Streptomycin 10 mg/mL and Penicillin 10,000 Units/mL). First of all, B16F10 was plated (5x104 cells/mL) in a 35mm polystyrene dish and, after 24h, exposed to hydroxyurea (30mM) or only medium (control group). Then, cells were placed in BioStation IM-Q inverted microscope (Nikon)[5] and images from 69 fields were acquired over 24 hours by a high sensitivity cooled charge-coupled device (CCD) camera (40x objective). At the end, the final database resulted in 69 image sequences with 95 frames with a spatial resolution of 640x480 pixels and duration of one minute. Figure 8.1 presents a frame example with two marked cells to show what is the subject of this dataset. For this dataset, image sequences are multitemporal images.

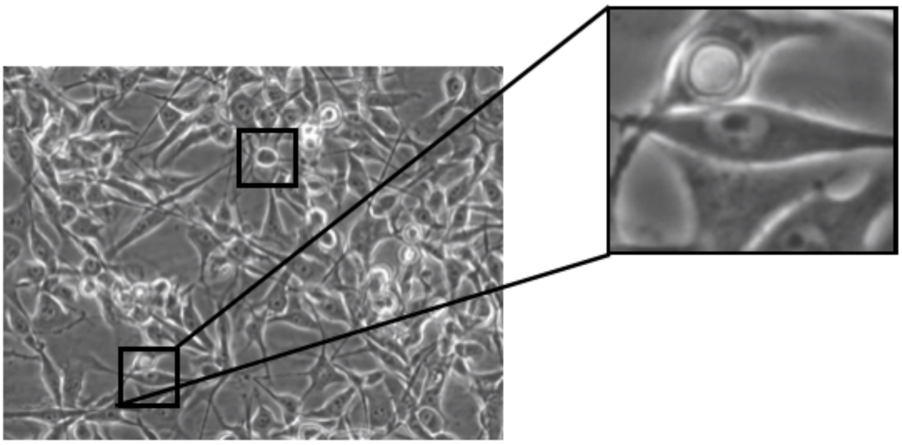


Figure 8.1: Example of cells from the melanoma cancer cell dataset. Two example cells are marked with a black bold square around its nucleoid. On the right we have a zoom on one of them (Image made by the author).

Hydroxyurea is a non-alkylating antineoplastic that selectively inhibits ribonucleoside diphosphate reductase, an enzyme required to convert ribonucleoside diphosphates into deoxyribonucleoside diphosphates, thereby preventing cells from leaving the G1/S phase of the cell cycle. In B16F10 cells, inhibition of migration by hydroxyurea starts from 1uM reaching maximum effect at 30uM without increasing cell death [Decaestecker et al., 2007].

In the control cell image sequences we see that cells increase the number and the velocity. When hydroxyurea is applied, the number of cells and the velocity decrease

[5] https://www.nikoninstruments.com/pr_BR/Produtos/Sistemas-de-triagem-de-celulas-vivas/BioStation-IM-Q

over time. Thus, it is interesting to analyze how a spatiotemporal descriptor can be used to discriminate the treated cells from the control cells in order to automate the process and help us better understand the phenomena. Figure 8.2 shows an example of a sequence from the dataset. On the left we see the evolution of melanoma cancer cells through time. On the right we see the cells treated with hydroxyurea. With the last frame, is easy to see how the number of cells increases without any treatment.

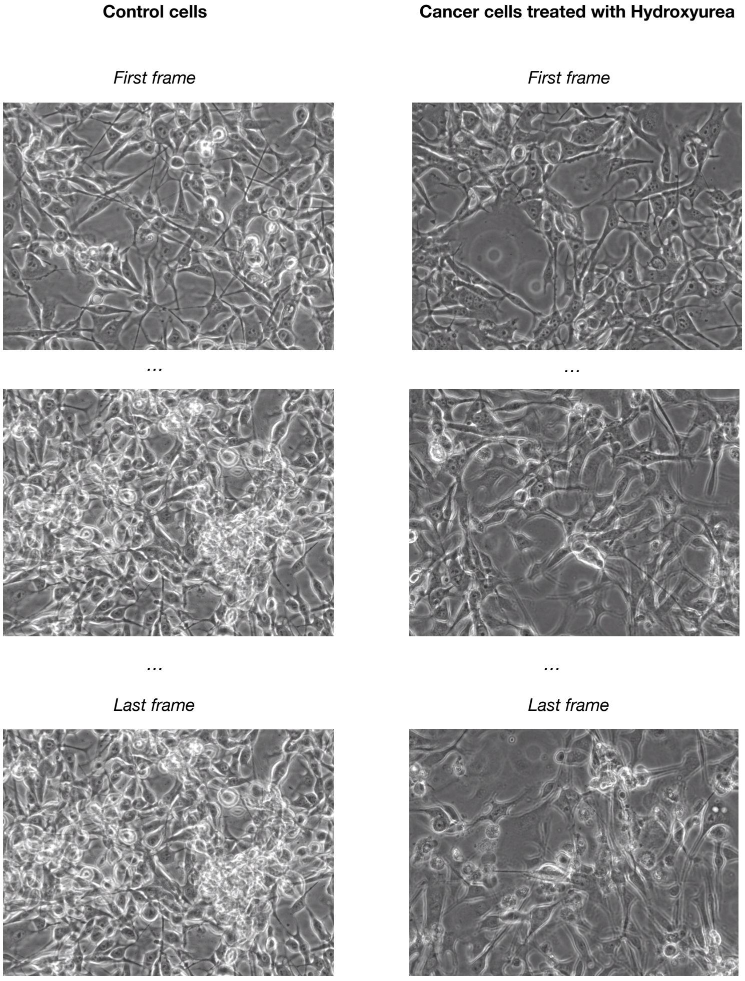


Figure 8.2: The melanoma cancer cell dataset composed by 69 image sequences of control melanoma cells and 69 image sequences for cells treated with hydroxyurea. On the left, we see the evolution of melanoma cancer cells through time. On the right, we see the cells treated with hydroxyurea. It is easy to see how the number of cells increases without any treatment (Image made by the author).

8.2 Results and Discussion

To evaluate the results of our experiments, we applied a 5x2-fold protocol. It consists of randomly splitting the MCC dataset five times into two folds, balanced by class. In each time, training and testing sets were switched and consequently five analysis for every model employed were conducted.

The baseline was computed with a dense extraction of the three handcrafted features HOG, HOF and HOGHOF. The results are depicted in Table 8.1. It can be observed that our assumption that a spatiotemporal descriptor could discriminate the control cells from the cancer cells is a fact, for all handcrafted features we achieved an accuracy greater than 80%.

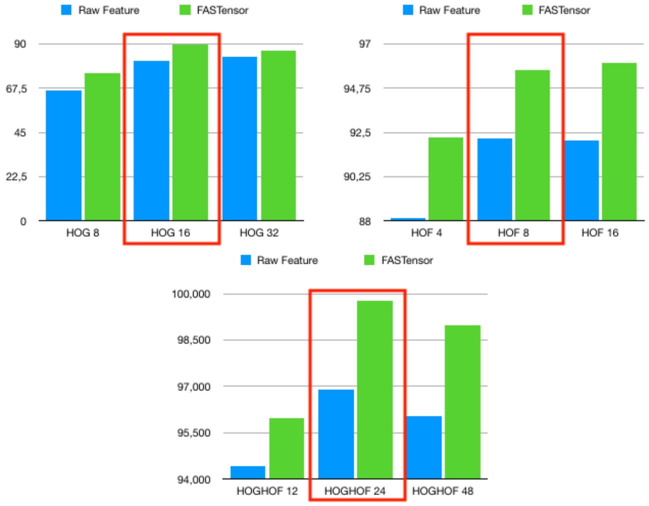


Figure 8.3: Compared results for different feature configurations and the *FASTensor* for the melanoma cancer cell dataset (Image made by the author).

Table 8.1: Baseline handcrafted features for the melanoma cancer cell dataset. We use a dense sampling extraction with the fixed number of bins, HOG with sixteen bins, HOF with eight bins, and the HOGHOF with twenty-four bins.

Method	Accuracy (%)
HOG 16 bins	81.22 ± 0.14
HOF 8 bins	92.2 ± 0.62
HOGHOF 24 bins	96.9 ± 0.24

Concerning our framework *FASTensor*, we can observe that all accuracies were higher than 89%. The results are depicted in Table 8.2. Our best result is also with the HOGHOF coded into the orientation tensor achieving 99.78% ± 0.34. To better visualize the difference of results between handcrafted features and *FASTensor*, Figure 8.4

Table 8.2: *FASTensor* results for the melanoma cancer cell dataset.

Method	**Accuracy (%)**
HOG 16 bins	89.58 ± 0.30
HOF 8 bins	95.69 ± 0.15
HOGHOF 24 bins	99.78 ± 0.34

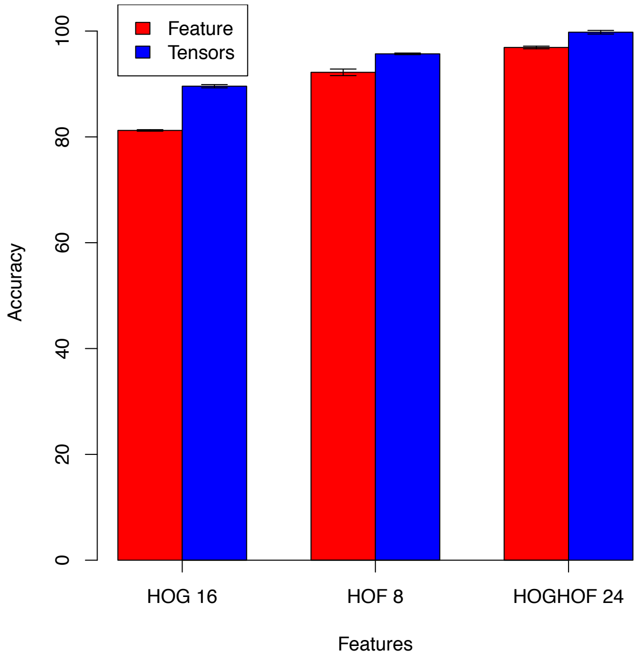


Figure 8.4: Compared results from handcrafted features and the *FASTensor* for the melanoma cancer cell dataset (Image made by the author).

shows each accuracy with the confidence interval of 95%. We see that our *FASTensor* iis also statistically better significant.

For the MCC dataset we did not perform deep features studies as we have a small amount of data and with our framework we already achieved an excellent result of 99.78% ± 0.34 with the HOGHOF *FASTensor*. Additionally, the ResNet-50 was pretrained on ImageNet [Deng et al., 2009], which is an RGB dataset with very distinct visual features compared to MCC.

8.3 Conclusion

In this chapter, we discussed the Small Data problem. We showed how literature on this matter is starting to grow, mainly working on how to learn feature with a small number of data. However, it was easy to see how Big Data has more works in literature than any terms related to Small Data. Moreover, we saw how learning from low data still is incipient. Thus, we need to put effort in this area to know how to represent, analyze and learn from Small Data.

In order to discuss on this matter, we proposed a small dataset called The Melanoma Cancer Cell dataset.

Although it was an easy task, as we saw by the high recognition rate achieved, we still can show that *FASTensor* can work on different spatiotemporal applications. Indeed, cells have very different shape an movement when compared to humans. Moreover, our multitemporal image classification opens the possibility of other drugs to be tested and analyzed.

Finally, we were able to validate all hypotheses for this application.

Chapter 9

Conclusion

In this thesis, we proposed an orientation tensor framework for video description called Features As Spatiotemporal Tensors (*FASTensor*). The orientation tensor created from each feature vector is accumulated for each image/frame. The accumulation through time provides the temporal description for the video or for multi-temporal images. We showed the mathematical proof and proof of context for the framework.

We evaluated the FASTensor in three different video and multitemporal image classification tasks: Human Action Recognition, Video Pornography classification and Melanoma Cancer Cell classification, to which we contribute with a new dataset.

Our experiments confirmed that the incorporation of covariance information from the features led to more effective video classification in different applications. This was shown with raw features HOG, HOF and HOGHOF, and deep features pretrained on a ResNet-50. In comparison with the state-of-the-art, our framework yielded better results.

For the Human Action Recognition task, it was possible to create a simple descriptor using orientation tensors that could maintain balance between size, computer complexity and recognition rate. However, the big limitation of our method is the number of actions that can be performed in one scene. Thus, for more complex video datasets we were not able to achieve competitive results, as the orientation tensor has a bigger tendency to become isotropic, that is, not have main direction information.

For the Video Pornography classification task, the FASTensor achieved the best results for the Pornography-800 and a competitive result for the Pornography-2k. In fact, this application is more suitable to work with orientation tensor, as the probability to become isotropic is inferior.

The Melanoma Cancer Cell (MCC) dataset provides better understanding of the cancer cell migration and anti-migration promoted by specific drugs, classifying in treated and untreated cell, being possible to characterize phenotypic and morphologic drug effects. This dataset showed that FASTensor can be used in very different applications. Moreover, the framework can be used in other cancer cells treatment analysis.

The MCC dataset could also start a discussion about the problem of Small Data. We showed how literature on this matter is starting to grow, mainly working on how to learn feature with a small number of data. However, it was easy to see how Big Data has more works in literature than any terms related to Small Data. Moreover, we see how learning from low data still is incipient. Thus, we need to put effort in this area to know how to represent, analyze and learn from Small Data.

With our results we can, therefore, confidently assert that FASTensors comprise the new state-of-the-art for video classification in the Pornography-800 dataset and for the Melanoma Cancer Cells dataset. For Human Action Recognition, we could also achieve competitive results. Therefore, orientation tensors carry more discriminative information than the feature vector itself, showing how robust is our method.

In resume, we started this thesis with three main hypotheses:

- Orientation tensors can be used as compact spatiotemporal representations, enabling dimensionality reduction and invariance, according to the feature used to built them;
- The framework created (FASTensor) can be used in different spatiotemporal tasks, as Human Action Recognition, Video Pornography Classification and Cancer Cell Classification; and,
- The framework created (FASTensor) can improve the accuracy of spatiotemporal tasks using handcrafted and deep features.

With our mathematical framework and our three applications using the FASTensor we were able to validate all those hypotheses. This thesis established the theoretical fundamentals for the orientation tensor framework, furnished a statistical analysis and was able to test the FASTensor in different applications.

As future work, we want to test other drugs in cancer cells and automate the analysis. We will investigate what more can be extracted with orientation tensors for this application, like motion tendency, cell density, among others.

We also want to analyze other applications that are suitable for FASTensors in medical imaging, remote sensing, surveillance, among other spatiotemporal tasks.

Furthermore, we will analyze the FASTensor as a descriptor creator not only for handcrafted features and deep learning features. We already saw the improvement for Pornography classification. We believe that we can improve the results by adding temporal information without the overhead of a very deep architecture for video classification. One idea is to add a layer in a CNN approach that creates tensors to add temporal information to the neural network.

Bibliography

Almeida, J., dos Santos, J. A., Alberton, B., Morellato, L. P. C., and da S. Torres, R. (2016). Phenological visual rhythms: Compact representations for fine-grained plant species identification. *Pattern Recognition Letters*, 81:90–100.

Altae-Tran, H., Ramsundar, B., Pappu, A. S., and Pande, V. (2017). Low data drug discovery with one-shot learning. *American Chemical Society (ACS) Central Science*, 3(4):283–293.

Andaló, F. A., Miranda, P. A. V., da Silva Torres, R., and Falcão, A. X. (2007). Detecting contour saliences using tensor scale. In *IEEE International Conference on Image Processing*, pages 349–352.

Augereau, B., Tremblais, B., and Fernandez-Maloigne, C. (2005). Vectorial computation of the optical flow in color image sequences. In *Thirteenth Color Imaging Conference*, pages 130–134.

Avila, S., Thome, N., Cord, M., Valle, E., and Araújo, A. A. (2013). Pooling in image representation: The visual codeword point of view. *Computer Vision and Image Understanding*, 117(5):453–465.

Avila, S., Thome, N., Cord, N., Valle, E., and Araújo, A. A. (2011). Bossa: Extended bow formalism for image classification. In *IEEE International Conference on Image Processing*, pages 2909–2912.

Baburaj, M. and Sudhish, N. (2019). Tensor based approach for inpainting of video containing sparse text. *Multimedia Tools and Applications*, 78(2):1805–1829.

Baeza-Yates, R. A. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval.* Addison-Wesley Longman Publishing Co., Inc.

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features. *Computer Vision and Image Understanding*, 110:346–359.

Ben-Cohen, A., Klang, E., Amitai, M. M., Goldberger, J., and Greenspan, H. (2018). Anatomical data augmentation for CNN based pixel-wise classification. In *International Symposium on Biomedical Imaging*, pages 1096–1099.

Bengio, Y. (2009). Learning deep architectures for ai. *Foundation Trends Machine Learning*, 2(1):1–127.

Bradbury, R. H. (2007). *Overview BT - Cancer*, pages 1–17. Springer.

Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision*, pages 25–36.

Caetano, C., Avila, S., Guimarães, S. J., and Araújo, A. A. (2014). Representing local binary descriptors with bossanova for visual recognition. In *29th Annual ACM Symposium on Applied Computing*, pages 49–54.

Caetano, C., Avila, S., Schwartz, W. R., Guimarães, S. J. F., and Araújo, A. A. (2016). A mid-level video representation based on binary descriptors: A case study for pornography detection. *Neurocomputing*, 213:102–114.

Castro, T. K., Almeida Perez, E., Mota, V. F., Chapiro, A., Vieira, M. B., and Freire, W. P. (2009). High frequency assessment from multiresolution analysis. In *International Conference on Computational Science*, pages 429–438.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition*, volume 1, pages 886–893.

Dalal, N., Triggs, B., and Schmid, C. (2006). Human detection using oriented histograms of flow and appearance. In *European Conference on Computer Vision*, pages 428–441.

Decaestecker, C., Debeir, O., Van Ham, P., and Kiss, R. (2007). Can anti-migratory drugs be screened in vitro? a review of 2d and 3d assays for the quantitative analysis of cell migration. *Medicinal Research Reviews*, 27(2):149–176.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition*, pages 248–255.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *Computer Vision and Pattern Recognition*, pages 2625–2634.

Douze, M., Jégou, H., Schmid, C., and Pérez, P. (2010). Compact video description for copy detection with precise temporal alignment. *Lecture Notes in Computer Science*, 6311:522–535.

Druon, M. (2009). *Modélisation du mouvement par polynômes orthogonaux : application à l'étude d'écoulements fluides*. PhD thesis, Université de Poitiers.

Duan, L., Chandrasekhar, V., Wang, S., Lou, Y., Lin, J., Bai, Y., Huang, T., Kot, A. C., and Gao, W. (2017). Compact descriptors for video analysis: the emerging mpeg standard. *CoRR*, abs/1704.08141.

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136.

Faraki, M., Palhang, M., and Sanderson, C. (2014). Log-euclidean bag of words for human action recognition. In *Institution of Engineering and Technology Computer Vision (IET-CV)*, volume 9, pages 331–339.

Farnebäck, G. (2001). Very high accuracy velocity estimation using orientation tensors, parametric motion, and simultaneous segmentation of the motion field. In *International Conference on Computer Vision*, pages 171–177.

Fei-fei, L. (2006). Knowledge transfer in learning to recognize visual object classes. In *International Conference on Development and Learning*, pages 1–51.

Förstner, W. (1994). A framework for low level feature extraction. In *European Conference on Computer Vision*, pages 383–394.

Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., and Greenspan, H. (2018). Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification. *ArXiv e-prints*.

Gilbert, A., Illingworth, J., and Bowden, R. (2011). Action recognition using mined hierarchical compound features. *Pattern Analysis and Machine Intelligence*, 33(5):883–897.

Gillet, J.-P., Varma, S., and Gottesman, M. M. (2013). The clinical relevance of cancer cell lines. *Journal of the National Cancer Institute*, 105:452–458.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Networks. *ArXiv e-prints.*

Goodspeed, A., Heiser, L., Gray, J., and Costello, J. (2016). Tumor-derived cell lines as molecular models of cancer pharmacogenomics. *Molecular Cancer Research*, 14(1):3–13.

Granlund, G. H. and Knutsson, H. (1995). *Signal Processing for Computer Vision.* Kluwer Academic Publishers.

Grundmann, M., Kwatra, V., Han, M., and Essa, I. (2010). Efficient hierarchical graph-based video segmentation. In *Computer Vision and Pattern Recognition*, pages 2141–2148.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Fourth Alvey Vision Conference*, pages 147–151.

Hart, I. R. (1979). The selection and characterization of an invasive variant of the b16 melanoma. *The American Journal of Pathology*, 97:587–600.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*, pages 770–778.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Hu, J., Xia, G.-S., Hu, F., and Zhang, L. (2015). Dense v.s. sparse: A comparative study of sampling analysis in scene classification of high-resolution remote sensing imagery. *ArXiv e-prints.*

Huh, M., Agrawal, P., and Efros, A. A. (2016). What makes imagenet good for transfer learning? *ArXiv e-prints.*

Imtiaz, H., Mahbub, U., and Ahad, M. (2011). Action recognition algorithm based on optical flow and ransac in frequency domain. In *Annual Conference of Society of Instrument and Control Engineers*, pages 1627 –1631.

Jain, M., Jégou, H., and Bouthemy, P. (2013). Better exploiting motion for better action recognition. In *Computer Vision and Pattern Recognition*, pages 2555–2562.

Ji, S., Xu, W., Yang, M., and Yu, K. (2013). 3d convolutional neural networks for human action recognition. *Pattern Analysis and Machine Intelligence*, 35(1):221–231.

Jia, C., Wang, S., Xu, X., Zhou, C., and Zhang, L. (2010). Tensor analysis and multi-scale features based multi-view human action recognition. In *International Conference on Computer Engineering and Technology*, pages 60–64.

Johansson, B., Farnebäck, G., and Ack, G. F. (2002). A theoretical comparison of different orientation tensors. In *Symposium on Image Analysis*, pages 69–73.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *Computer Vision and Pattern Recognition*, pages 1725–1732.

Kihl, O., Picard, D., and Gosselin, P.-H. (2013). A unified formalism for video descriptor. In *IEEE International Conference on Image Processing*, pages 2416–2419.

Kihl, O., Tremblais, B., Augereau, B., and Khoudeir, M. (2010). Human activities discrimination with motion approximation in polynomial bases. In *IEEE International Conference on Image Processing*, pages 2469–2472.

Kim, T., Wong, S., and Cipolla, R. (2007). R.: Tensor canonical correlation analysis for action classification. In *Computer Vision and Pattern Recognition*, pages 1–8.

Kläser, A., Marszałek, M., and Schmid, C. (2008). A spatio-temporal descriptor based on 3d-gradients. In *British Machine Vision Conference*, pages 995–1004.

Kobayashi, T. and Otsu, N. (2012). Motion recognition using local auto-correlation of spacetime gradients. *Pattern Recognition Letters*, 33(9):1188–1195.

Kovashka, A. and Grauman, K. (2010). Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *Computer Vision and Pattern Recognition*, pages 2046–2053.

Krausz, B. and Bauckhage, C. (2010). Action recognition in videos using nonnegative tensor factorization. *International Conference on Pattern Recognition*, 0:1763–1766.

Kriegel, F., Köhler, R., Bayat-Sarmadi, J., Bayerl, S., E. Hauser, A., Niesner, R., Luch, A., and Cseresnyés, Z. (2017). Cell shape characterization and classification with discrete fourier transforms and self-organizing maps. *International Society for Advancement of Cytometry*, 93:323–333.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, pages 1097–1105.

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., and Serre, T. (2011). Hmdb: a large video database for human motion recognition. In *International Conference on Computer Vision*, pages 2556–2563.

Lan, X., Ye, M., Zhang, S., Zhou, H., and Yuen, P. C. (2018). Modality correlation aware sparse representation for rgb-infrared object tracking. *Pattern Recognition Letters*. To appear.

Lan, X., Zhang, S., Yuen, P. C., and Chellappa, R. (2018). Learning common and feature-specific patterns: A novel multiple-sparse-representation-based tracker. *IEEE Transactions on Image Processing*, 27(4):2022–2037.

Laptev, I., Caputo, B., Schuldt, C., and Lindeberg, T. (2007). Local velocity-adapted motion events for spatio-temporal recognition. *Computer Vision and Image Understanding*, 108:207–229.

Laptev, I. and Lindeberg, T. (2003). Space-time interest points. In *International Conference on Computer Vision*, pages 432–439.

Laptev, I., Marszałek, M., Schmid, C., and Rozenfeld, B. (2008). Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition*, pages 1–8.

Laptev, I. and Pérez, P. (2007). Retrieving actions in movies. In *International Conference on Computer Vision*, pages 1–8.

Larochelle, H., Erhan, D., and Bengio, Y. (2008). Zero-data learning of new tasks. In *Association for the Advancement of Artificial Intelligence*, pages 646–651.

Le, Q. V., Zou, W. Y., Yeung, S. Y., and Ng, A. Y. (2011). Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Computer Vision and Pattern Recognition*, pages 3361–3368.

LeCun, Y. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pages 2278–2324.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521:436–444.

Liu, A., Su, Y., Nie, W., and Kankanhalli, M. (2017). Hierarchical clustering multi-task learning for joint human action grouping and recognition. *Pattern Analysis and Machine Intelligence*, 39(1):102–114.

Liu, H., Feris, R., and Sun, M.-T. (2011). Benchmarking datasets for human activity recognition. In *Visual Analysis of Humans*, pages 411–427.

Liu, J., Luo, J., and Shah, M. (2009). Recognizing realistic actions from videos in the wild. In *Computer Vision and Pattern Recognition*, pages 1996–2003.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, volume 2, pages 1150–1157.

Lucas, B. and Kanade, T. (1981a). An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679.

Lucas, B. D. and Kanade, T. (1981b). An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679.

Marszałek, M., Laptev, I., and Schmid, C. (2009). Actions in context. In *Computer Vision and Pattern Recognition*, pages 2929–2936.

Martin, T. A., Ye, L., Sanders, A. J., Lane, J., and Jiang, W. G. (2013). Cancer invasion and metastasis: Molecular and cellular perspective. *Metastatic Cancer: Clinical and Biological Perspectives*, pages 135–168.

Masuzzo, P., Van Troys, M., Ampe, C., and Martens, L. (2016). Taking aim at moving targets in computational cell migration. *Trends in Cell Biology*, 26:88–110.

Minhas, R., Baradarani, A., Seifzadeh, S., and Jonathan Wu, Q. M. (2010). Human action recognition using extreme learning machine based on visual vocabularies. *Neurocomputing*, 73(10-12):1906–1917.

Mordohai, P. and Medioni, G. G. (2007). *Tensor Voting: A perceptual Organization Approach to Computer Vision and Machine Learning*. Morgan and Claypool Publishers.

Moreira, D., Avila, S., Perez, M., Moraes, D., Testoni, V., Valle, E., Goldenstein, S., and Rocha, A. (2016). Pornography classification: The hidden clues in video space-time. *Forensic Science International*, 268:46–61.

Mota, V., Souza, J., Araújo, A. A., and Vieira, M. B. (2013). Combining orientation tensors for human action recognition. In *Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 328–333.

Mota, V. F., De Almeida Perez, E., De Castro, T. K., Chapiro, A., and Bernardes Vieira, M. (2009). Detection of high frequency regions in multiresolution. In *IEEE International Conference on Image Processing*, pages 2141–2144.

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

Mota, V. F., Perez, E. A., Vieira, M. B., Maciel, L. M., Precioso, F., and Gosselin, P.-H. (2012). A tensor based on optical flow for global description of motion in videos. In *Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 298–301.

Oliveira, F., Maia, H., Mota, V. F., Vieira, M., and Araújo, A. A. (2014). Video tensor self-descriptor based on variable size block matching. In *Workshop on Vision-based Human Activity Recognition - Conference on Graphics, Patterns and Images (SIBGRAPI)*.

Oliveira, F. L. M., Maia, H., Mota, V. F., Vieira, M. B., and Araújo, A. A. (2015). A variable size block matching based descriptor for human action recognition. *Journal of Communication and Information Systems*, 30(1):90–99.

Pasupa, K. and Sunhem, W. (2016). A comparison between shallow and deep architecture classifiers on small dataset. In *International Conference on Information Technology and Electrical Engineering (ICITEE)*, pages 1–6.

Peng, X., Zou, C., Qiao, Y., and Peng, Q. (2014). Action recognition with stacked fisher vectors. In *European Conference on Computer Vision*, pages 581–595.

Perez, E. A., Mota, V. F., Maciel, L. M., Sad, D., and Vieira, M. B. (2012). Combining gradient histograms using orientation tensors for human action recognition. In *International Conference on Pattern Recognition*, pages 3460–3463.

Perez, M., Avila, S., Moreira, D., Moraes, D., Testoni, V., Valle, E., Goldenstein, S., and Rocha, A. (2017). Video pornography detection through deep learning techniques and motion information. *Neurocomputing*, 230(C):279–293.

Pernici, F. and Del Bimbo, A. (2014). Object Tracking by Oversampling Local Features. *Pattern Analysis and Machine Intelligence*, 36(12):2538–2551.

Perona, P. and Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *Pattern Analysis and Machine Intelligence*, 12:629–639.

Prates, R. and Schwartz, W. R. (2018). Kernel multiblock partial least squares for a scalable and multicamera person reidentification system. *Journal of Electronic Imaging*, 27(3):1–33.

Qiu, Z., Yao, T., and Mei, T. (2017). Learning spatio-temporal representation with pseudo-3d residual networks. In *International Conference on Computer Vision*, pages 5534–5542.

Ramnath, N. and Creaven, P. (2004). Matrix metalloproteinase inhibitors. *Current Oncology*, 6:96–102.

Rodriguez, M. D., Ahmed, J., and Shah, M. (2008). Action mach: a spatio-temporal maximum average correlation height filter for action recognition. In *Computer Vision and Pattern Recognition*, pages 1–8.

Sad, D., Mota, V. F., Maciel, L., Vieira, M. B., and Araújo, A. A. (2013). A tensor motion descriptor based on multiple gradient estimators. In *Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 70–74.

Sadanand, S. and Corso, J. J. (2012). Action bank: A high-level representation of activity in video. In *Computer Vision and Pattern Recognition*, pages 1234–1241.

Saha, P. K. and Xu, Z. (2010). An analytic approach to tensor scale with an efficient algorithm and applications to image filtering. In *International Conference on Digital Image Computing Techniques and Applications*, pages 429–434.

Sani, S., Massie, S., Wiratunga, N., and Cooper, K. (2017). Learning deep and shallow features for human activity recognition. *ArXiv e-prints*, pages 469–482.

Santos, R. J. (2017). *Matrizes, Vetores e Geometria Analítica.* Imprensa Universitária da UFMG.

Schindler, K. and Van Gool, L. (2008). Action Snippets: How many frames does human action recognition require? In *Computer Vision and Pattern Recognition*, pages 1–8.

Schuldt, C., Laptev, I., and Caputo, B. (2004). Recognizing human actions: A local svm approach. In *International Conference on Pattern Recognition*, pages 32–36.

Shao, L. and Gao, R. (2010). A wavelet based local descriptor for human action recognition. In *British Machine Vision Conference*, pages 72.1–10.

Short, M., Black, L., Smith, A., Wetterneck, C., and E Wells, D. (2012). A review of internet pornography use research: Methodology and content from the past 10 years. In *Cyberpsychology, behavior and social networking*, volume 15, pages 13–23.

Sivic, J. and Zisserman, A. (2003). Video google: a text retrieval approach to object matching in videos. In *International Conference on Computer Vision*, pages 1470–1477 vol.2.

Solmaz, B., Assari, S. M., and Shah, M. (2012). Classifying web videos using a global video descriptor. *Machine Vision and Applications*, pages 1–13.

Souza, K., Araújo, A. A., Patrocínio Jr, Z., and Guimarães, S. (2014). Graph-based hierarchical video segmentation based on a simple dissimilarity measure. *Pattern Recognition Letters*, 47:85–92.

Sze, V., Budagavi, M., and Sullivan, G. J. (2014). *High Efficiency Video Coding: Algorithms and Architectures*. Springer.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Computer Vision and Pattern Recognition*, pages 1–9.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In *International Conference on Computer Vision*, pages 4489–4497.

Van de Sande, K. E. A., Gevers, T., and Snoek, C. G. M. (2010). Evaluating color descriptors for object and scene recognition. *Pattern Analysis and Machine Intelligence*, 32(9):1582–1596.

Varghese, A., Mohammed, P. S. K., Chennamsetty, S. S., and Krishnamurthi, G. (2017). Generative adversarial networks for brain lesion detection. In *Medical Imaging: Image Processing*, volume 10133, pages 1–9.

Varol, G., Laptev, I., and Schmid, C. (2018). Long-term temporal convolutions for action recognition. *Pattern Analysis and Machine Intelligence*, 40(6):1510–1517.

Vig, E., Dorr, M., and Cox, D. D. (2012). Saliency-based selection of sparse descriptors for action recognition. *IEEE International Conference on Image Processing*, pages 1405–1408.

Villareal, M. O., Sato, Y., Matsuyama, K., and Isoda, H. (2018). Daphnane diterpenes inhibit the metastatic potential of b16f10 murine melanoma cells in vitro and in vivo. *BMC Cancer*, 18:856.

Wang, H., Kläser, A., Schmid, C., and Cheng-Lin, L. (2011). Action Recognition by Dense Trajectories. In *Conference on Computer Vision and Pattern Recognition*, pages 3169–3176.

Wang, H., Kläser, A., Schmid, C., and Liu, C.-L. (2013). Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103:60–79.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Liu, G., Tao, A., Kautz, J., and Catanzaro, B. (2018). Video-to-video synthesis. In *Advances in Neural Information Processing Systems*, volume 31, pages 1144–1156.

Wehrmann, J., Simões, G. S., Barros, R. C., and Cavalcante, V. F. (2018). Adult content detection in videos with convolutional and recurrent neural networks. *Neurocomputing*, 272:432–438.

Westin, C.-F. (1994). *A Tensor Framework for Multidimensional Signal Processing.* PhD thesis, Linköping University, Sweden. Dissertation No 348, ISBN 91-7871-421-4.

Wiegand, T. and Sullivan, G. J. (2007). The h. 264/avc video coding standard [standards in a nutshell]. *IEEE Signal Processing Magazine*, 24(2):148–153.

Xu, Z., Gao, Z., Hoffman, E. A., and Saha, P. K. (2012). Tensor scale-based anisotropic region growing for segmentation of elongated biological structures. In *International Symposium on Biomedical Imaging*, pages 1032–1035.

Young, E. W. K. (2013). Cells, tissues, and organs on chips: challenges and opportunities for the cancer tumor microenvironment. In *Integrative Biology*, volume 5, pages 1096–1109.

Zelnik-manor, L. and Irani, M. (2001). Event-based analysis of video. In *Computer Vision and Pattern Recognition*, pages 123–130.

Zhang, J., Li, Z., Jing, P., Liu, Y., and Su, Y. (2017). Tensor-driven low-rank discriminant analysis for image set classification. *Multimedia Tools and Applications*, 78:4001–4020.

Zhang, J., Liu, Y., and Jiang, J. (2018). Tensor learning and automated rank selection for regression-based video classification. *Multimedia Tools and Applications*, 77:29213–29230.

Zhen, X. and Shao, L. (2012). A local descriptor based on laplacian pyramid coding for action recognition. *Pattern Recognition Letters*, 34:1899–1905.

Zhou, T., Li, N., Cheng, X., Xu, Q., Zhou, L., and Wu, Z. (2016). Learning semantic context feature-tree for action recognition via nearest neighbor fusion. *Neurocomputing*, 201:1–11.