



Universidade Estadual de Campinas
Instituto de Computação



Darwin Ttito Concha

Multi-Stream Convolutional Neural Networks
for Action Recognition in Video Sequences
Based on Spatio-Temporal Information

Redes Neurais Convolucionais de Múltiplos Canais
para Reconhecimento de Ações em Sequências
de Vídeos Baseado em Informações Espaço-Temporais

CAMPINAS
2019

Darwin Ttito Concha

**Multi-Stream Convolutional Neural Networks
for Action Recognition in Video Sequences
Based on Spatio-Temporal Information**

**Redes Neurais Convolucionais de Múltiplos Canais
para Reconhecimento de Ações em Sequências
de Vídeos Baseado em Informações Espaço-Temporais**

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

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

Supervisor/Orientador: Prof. Dr. Hélio Pedrini

Este exemplar corresponde à versão final da Dissertação defendida por Darwin Ttito Concha e orientada pelo Prof. Dr. Hélio Pedrini.

CAMPINAS
2019

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

T789m Ttito Concha, Darwin, 1995-
Multi-stream convolutional neural networks for action recognition in video sequences based on spatio-temporal information / Darwin Ttito Concha. – Campinas, SP : [s.n.], 2019.

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

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

Informações para Biblioteca Digital

Título em outro idioma: Redes neurais convolucionais de múltiplos canais para reconhecimento de ações em sequências de vídeos baseado em informações espaço-temporais

Palavras-chave em inglês:

Convolutional neural networks

Computer vision

Machine learning

Área de concentração: Ciência da Computação

Titulação: Mestre em Ciência da Computação

Banca examinadora:

Hélio Pedrini [Orientador]

Ricardo Cerri

Esther Luna Colombini

Data de defesa: 04-04-2019

Programa de Pós-Graduação: Ciência da Computação

Identificação e informações acadêmicas do(a) aluno(a)

- ORCID do autor: <https://orcid.org/0000-0001-5539-3313>

- Currículo Lattes do autor: <http://lattes.cnpq.br/1283742349945922>



Universidade Estadual de Campinas
Instituto de Computação



Darwin Ttito Concha

**Multi-Stream Convolutional Neural Networks
for Action Recognition in Video Sequences
Based on Spatio-Temporal Information**

**Redes Neurais Convolucionais de Múltiplos Canais
para Reconhecimento de Ações em Sequências
de Vídeos Baseado em Informações Espaço-Temporais**

Banca Examinadora:

- Prof. Dr. Hélio Pedrini
IC/UNICAMP
- Prof. Dr. Ricardo Cerri
DC/UFSCar
- Profa. Dra. Esther Luna Colombini
IC/UNICAMP

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 04 de abril de 2019

Acknowledgements

I am thankful to my whole family, to my mother Marina because she is my inspiration and the reason to fight for my goals, to my father Donato for his hard work to allow me to be here, to my brother Kleiber because, despite his disinterest, he made everything look easy, and to my little sister Sayda for being the person with whom I return to be a child and forget all problems that afflict me. Thank you for all the love, understanding and support you give me.

I am grateful to all the members of the Visual Informatics Laboratory (LIV) in the Institute of Computing (IC) at the University of Campinas (UNICAMP), for their advices and teaching. They were like my second family. I would like to thank Anderson for being the laboratory administrator and being there at any time of the day to solve all types of problems even though this is not his main job, as well as Rodolfo and Jose Luis for being like my brothers and sharing an endless number of experiences together.

I am very grateful to my advisor Hélio Pedrini for his patience, teaching and constant advice throughout the Master's degree and research process, without his help none of this would be possible.

I am thankful to the entire UNICAMP family: people from the UNICAMP's Registrar (DAC) and IC's graduate secretary for making it seem that there are no bureaucratic issues, particularly to Wilson for his great ability to solve any type of request. To the professors of the IC for sharing their knowledge and providing me with quality education.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Resumo

Avanços na tecnologia digital aumentaram as capacidades de reconhecimento de eventos por meio do desenvolvimento de dispositivos com alta resolução, pequenas dimensões físicas e altas taxas de amostragem. O reconhecimento de eventos complexos em vídeos possui várias aplicações relevantes, particularmente devido à grande disponibilidade de câmeras digitais em ambientes como aeroportos, bancos, estradas, entre outros. A grande quantidade de dados produzidos é o cenário ideal para o desenvolvimento de métodos automáticos baseados em aprendizado de máquina profundo. Apesar do progresso significativo alcançado com as redes neurais profundas aplicadas a imagens, a compreensão do conteúdo de vídeos ainda enfrenta desafios na modelagem de relações espaço-temporais. Nesta dissertação, o problema do reconhecimento de ações humanas em vídeos foi investigada. Uma rede de múltiplos canais é a arquitetura de escolha para incorporar informações temporais, uma vez que se pode beneficiar de redes profundas pré-treinadas para imagens e de características tradicionais para inicialização. Além disso, seu custo de treinamento é geralmente menor do que o das redes neurais para vídeos. Imagens de ritmo visual são exploradas, pois codificam informações de longo prazo quando comparadas a quadros estáticos e fluxo ótico. Um novo método baseado em rastreamento de pontos é desenvolvido para decidir a melhor direção do ritmo visual para cada vídeo. Além disso, redes neurais recorrentes foram treinadas a partir das características extraídas dos canais da arquitetura proposta. Experimentos conduzidos nas desafiadoras bases de dados públicas UCF101 e HMDB51 mostraram que a abordagem é capaz de melhorar o desempenho da rede, alcançando taxas de acurácia comparáveis aos métodos da literatura. Embora os ritmos visuais sejam originalmente criados a partir de imagens RGB, outros tipos de fontes e estratégias para sua criação são explorados e discutidos, tais como fluxo ótico, gradientes de imagem e histogramas de cores.

Abstract

Advances in digital technology have increased event recognition capabilities through the development of devices with high resolution, small physical dimensions and high sampling rates. The recognition of complex events in videos has several relevant applications, particularly due to the large availability of digital cameras in environments such as airports, banks, roads, among others. The large amount of data produced is the ideal scenario for the development of automatic methods based on deep learning. Despite the significant progress achieved through image-based deep neural networks, video content understanding still faces challenges in modeling spatio-temporal relations. In this dissertation, we address the problem of human action recognition in videos. A multi-stream network is our architecture of choice to incorporate temporal information, since it may benefit from pre-trained deep networks for images and from hand-crafted features for initialization. Furthermore, its training cost is usually lower than video-based networks. We explore visual rhythm images since they encode longer-term information when compared to still frames and optical flow. We propose a novel method based on point tracking for deciding the best visual rhythm direction for each video. In addition, we experimented with recurrent neural networks trained from the features extracted from the streams of the previous architecture. Experiments conducted on the challenging UCF101 and HMDB51 public datasets demonstrated that our approach is able to improve network performance, achieving accuracy rates comparable to the state-of-the-art methods. Even though the visual rhythms are originally created from RGB images, other types of source and strategies for their creation are explored and discussed, such as optical flow, image gradients and color histograms.

List of Figures

| | | |
|------|--|----|
| 1.1 | Caption for LOF | 15 |
| 2.1 | Representation of a video sequence | 19 |
| 2.2 | XT - YT slices | 19 |
| 2.3 | Example of visual rhythm image generated for WallPushups class from UCF101. The central row of each frame becomes a slice in the resulting image. Extracted from [17]. | 21 |
| 2.4 | Modified spatio-temporal slice: the horizontal-mean/vertical-mean slice from a given frame contains the average of the columns/rows. The slices were resized for illustration purposes. Extracted from [17]. | 22 |
| 2.5 | Optical flow of a video sequence | 23 |
| 2.6 | Transfer Learning | 25 |
| 2.7 | Convolutional Neural Network architecture | 26 |
| 2.8 | Convolutional Neural Network architecture | 27 |
| 2.9 | Recurrent Neural Networks architectures | 28 |
| 2.10 | Bi-Directional Recurrent Neural Networks architecture | 29 |
| 3.1 | Frames of some videos from the UCF101 dataset. Extracted from [70]. . . | 34 |
| 3.2 | Frames of some videos from the HMDB dataset. Extracted from [40]. . . | 35 |
| 4.1 | Examples of visual rhythm construction using the tracking a certain path, such as (a) vertical, (b) horizontal and (c) circular. | 38 |
| 4.2 | Examples of visual rhythm construction using the diagonal pixels and compression of the information through the mean operation. | 38 |
| 4.3 | Horizontal-mean and vertical-mean visual rhythm images as a unique dataset. . . | 39 |
| 4.4 | Examples of modified visual rhythms for a white background video and a black square figure that moves according to the direction of the arrows drawn next to it. | 40 |
| 4.5 | A moving object considering two consecutive frames and horizontal-mean slices. Parallel movement is better captured in the slice. | 40 |
| 4.6 | Examples of modified visual rhythms for TrampolineJumping and WallPushups classes from UCF101 dataset. Red arrows indicate the predominant direction of the action. Extracted from [17]. | 41 |
| 4.7 | Construction process of the adaptive visual rhythm. Extracted from [17]. . | 41 |
| 4.8 | Overview of our three-stream proposal for action recognition. Extracted from [17]. | 43 |
| 4.9 | Fragments from the UCF101 that show the significant variation in appearance in the two halves of the videos. | 44 |
| 4.10 | Overview of our improved spatial stream versus the spatial stream of the literature. | 44 |

| | | |
|------|--|----|
| 4.11 | Overview of the temporal stream of the literature. | 45 |
| 4.12 | Overview of our spatial-temporal stream. | 45 |
| 4.13 | Overview of our CNN-RNN multi-stream proposal for action recognition. . | 46 |
| 4.14 | Overview of our innovative LSTM stream. | 47 |
| 4.15 | Overview of our weighted average fusion technique. | 48 |
| 5.1 | First row, from top to bottom, shows some videos of the UCF101 dataset that belong to the Typing class. The second, third and fourth row exhibit visual rhythm images obtained from the horizontal-mean, vertical-mean and diagonal strategies. | 50 |
| 5.2 | Videos of the UCF101 dataset with their respective optical flow images below each of them. | 51 |
| 5.3 | Bar graph for the accuracy obtained for each class of the HMDB51 dataset using our adaptive visual rhythm. | 52 |
| 5.4 | Bar graph for the accuracy obtained for each class of the UCF101 dataset using our adaptive visual rhythm. | 53 |
| 5.5 | Bar graph for the accuracy obtained for each class of the HMDB51 dataset using our three-stream approach 1. | 55 |
| 5.6 | Bar graph for the accuracy obtained for each class of the UCF101 dataset using our three-stream approach 1. | 56 |
| 5.7 | Bar graph for the accuracy obtained for each class of the HMDB51 dataset using our three-stream approach 2. | 58 |
| 5.8 | Bar graph for the accuracy obtained for each class of the UCF101 dataset using our three-stream approach 2. | 58 |

List of Tables

| | | |
|------|--|----|
| 3.1 | Most used datasets in the human action recognition problem in videos. . . | 33 |
| 5.1 | Individual results for ResNet152. | 50 |
| 5.2 | Individual results for Inception V3. | 51 |
| 5.3 | Results and comparison of different approaches used to create the visual rhythms. | 52 |
| 5.4 | Individual results for ResNet152. | 54 |
| 5.5 | Individual results for Inception V3. | 54 |
| 5.6 | Results and comparison of the individual results (streams). | 54 |
| 5.7 | Results for RGB*, optical flow and adaptive visual rhythm stream fusion for ResNet152. | 54 |
| 5.8 | Results for RGB*, optical flow and adaptive visual rhythm stream fusion for Inception V3. | 55 |
| 5.9 | Results for RGB*, optical flow and adaptive visual rhythm stream fusion. . | 55 |
| 5.10 | Results for stream combination using the ResNet152. | 56 |
| 5.11 | Results for stream combination using the Inception V3. | 57 |
| 5.12 | Results for RGB*, optical flow and adaptive visual rhythm stream fusion. . | 57 |
| 5.13 | Comparison of accuracy rates (%) for UCF101 and HMDB51 datasets. Cells on bold represents the overall highest accuracies, whereas underlined cells consist of the best results using only ImageNet to pre-train the network. | 59 |

List of Abbreviations and Acronyms

| | |
|---------|------------------------------------|
| 2D | Two-Dimensional |
| 3D | Three-Dimensional |
| AVR | Adaptive Visual Rhythm |
| BoW | Bag-of-Words |
| CEC | Constant Error Carousel |
| CNN | Convolutional Neural Network |
| ConvNet | Convolutional Networks |
| DC | Direct Current |
| FPS | Frames per Second |
| GPU | Graphics Processing Unit |
| GRU | Gated Recurrent Unit |
| HMDB51 | Human Motion DataBase |
| HOF | Histogram of Optical Flow |
| HOG | Histogram of Oriented Orientations |
| I3D | Inflated 3D ConvNet |
| IC | Institute of Computing |
| KNN | K-Nearest Neighbors |
| LBP | Local Binary Pattern |
| LIV | Laboratory of Visual Informatics |
| LSTM | Long Short-Term Memory |
| MBH | Motion Boundary Histogram |
| MEI | Motion Energy Image |
| MHI | Motion History Image |
| MPEG | Moving Picture Expert Group |
| PoTion | Pose Motion |
| RGB | Red-Green-Blue |
| RNN | Recurrent Neural Networks |
| SVM | Support Vector Machine |
| UCF101 | University of Central Florida |
| UNICAMP | University of Campinas |
| VGG | Visual Geometry Group |
| VLBP | Volume Local Binary Pattern |
| VR | Visual Rhythm |

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 14 |
| 1.1 | Problem and Motivation | 15 |
| 1.2 | Research Questions | 16 |
| 1.3 | Objectives | 16 |
| 1.4 | Contributions | 17 |
| 1.5 | Text Organization | 17 |
| 2 | Background | 18 |
| 2.1 | Video Sequence Understanding and XT-YT slices | 18 |
| 2.2 | Visual Rhythm | 20 |
| 2.2.1 | Literature Review | 20 |
| 2.2.2 | Construction | 21 |
| 2.3 | Optical Flow | 22 |
| 2.3.1 | Lucas-Kanade Method | 24 |
| 2.4 | Deep Learning | 24 |
| 2.4.1 | Transfer Learning | 25 |
| 2.4.2 | Convolutional Neural Networks | 25 |
| 2.4.3 | Recurrent Neural Networks | 28 |
| 2.5 | Human Action Recognition | 29 |
| 2.5.1 | Problem Definition | 29 |
| 2.5.2 | Literature Review | 30 |
| 3 | Materials | 33 |
| 3.1 | Action Datasets | 33 |
| 3.1.1 | UCF101 Dataset | 34 |
| 3.1.2 | HMDB51 Dataset | 34 |
| 3.2 | Computational Resources | 34 |
| 4 | Proposed Action Recognition Method | 36 |
| 4.1 | Methodology Overview | 36 |
| 4.2 | Visual Rhythms | 37 |
| 4.2.1 | Visual Rhythm Construction | 37 |
| 4.2.2 | Resizing of Visual Rhythm Images | 37 |
| 4.2.3 | Analysis of Vertical-Mean and Horizontal-Mean Visual Rhythms | 39 |
| 4.2.4 | Adaptive Visual Rhythm | 41 |
| 4.3 | Multi-Stream Convolutional Neural Network | 43 |
| 4.3.1 | Improved Spatial Stream | 43 |
| 4.3.2 | Temporal Stream | 45 |
| 4.3.3 | Spatio-Temporal Stream | 45 |

| | | |
|----------|---|-----------|
| 4.4 | Multi-Stream Convolutional and Recurrent Neural Network | 46 |
| 4.4.1 | LSTM Stream (RGB + Optical Flow) | 46 |
| 4.4.2 | Weighted Average Fusion | 47 |
| 5 | Experiments | 49 |
| 5.1 | Visual Rhythms | 49 |
| 5.2 | Multi-Stream Architectures | 52 |
| 5.2.1 | Approach 1 | 53 |
| 5.2.2 | Approach 2 | 55 |
| 5.3 | State-of-the-Art Comparison | 57 |
| 6 | Conclusions and Future Work | 60 |
| | Bibliography | 62 |

Chapter 1

Introduction

In recent years, one of the main sources of new data has been video cameras. This type of devices is widely available in all places presented as mobile devices, robotics and video surveillance. Due to the significant growth of this type of data and rapid technological advances, many video datasets have become available, allowing the research and development of several applications oriented to video analysis in public, private and restricted areas such as streets, banks and radioactive places, respectively. Therefore, automatic procedures are needed to extract useful information from videos (spatial and temporal) to analyze and make the best decisions.

The problem addressed in this work is the recognition of human actions in video sequences [8, 16, 45, 58, 73, 80, 91], which aims to detect and identify actions of one or more agents. It is a challenging task since the same action may vary according to the actor and the scene may present difficult conditions, such as occlusions, background clutter and camera motion. This problem has several relevant applications, such as intelligent surveillance [33], human-computer interaction [27, 63] and healthy monitoring [13].

Based on the taxonomy proposed by Goodfellow et al. [26], the approaches to this problem can be categorized into two groups: (i) traditional methods [6, 56, 77, 92], where the action representation is explicitly chosen and the action recognition is defined under conventional machine learning algorithms, and (ii) representation-learning strategies that explore machine learning techniques for both tasks. The latter includes shallow approaches, such as dictionary-based methods [44, 55, 57, 83], and deep learning strategies [33, 34, 35, 50, 59, 67].

The majority of current approaches that address this problem employ deep learning, since it has shown to be a useful tool to generalize data in complex scenarios, achieving impressive results in different computer vision problems (for instance, image classification). However, the inclusion of temporal information may increase the number of parameters in the network, leading to a significant increase in the training cost. Moreover, designing spatio-temporal models brings a major issue: choosing a proper temporal extension that encloses every possible action without compromising the computational cost. For this reason, many recent deep learning proposals have explored hand-crafted inputs, such as optical flow images, in order to encode action dynamics. Image networks and fusion techniques are used to process these inputs and capture temporal evolution [34, 35, 50, 59, 67, 84, 88].

In this work, we propose a three-stream architecture based on the two-stream one [67]

that explores complementary modalities to recognize the action: RGB (Red-Green-Blue) frames (spatial) and optical flow images (temporal). In our architecture, a third modality called visual rhythm is used to provide dynamic information of the entire video for the network. In addition, we modify the original spatial and temporal streams to incorporate a temporal extension using Long Short-Term Memory (LSTM) networks.

1.1 Problem and Motivation

Video recognition involves its semantic understanding, which consists of labeling all objects, people and their events. In other words, it contains levels that are responsible for particular objectives, such as object-level understanding (location of people and objects), tracking (trajectories of objects), pose (parts of the human body) and activity (recognition of human actions and events). These levels are of great importance due to the variety of applications that each one offers. For example, the detection and tracking of objects can be adapted to applications related to the analysis and behavior of pedestrians, which is to detect the human agents present in a given video and analyze their movement patterns. On the other hand, action recognition in real time with the help of video surveillance cameras plays a very important role in the prevention and detection of actions that go against the rules of certain places [2, 31].



Figure 1.1: Frame captured by a video-surveillance camera that shows two people in a discussion. Extracted from [94].

Automatic video analysis is a challenging task. Similarly to image analysis, it requires previous stages for the extraction and processing of its features by means of classification techniques [21]. However, since a video is a sequence of several images, the two mentioned tasks demand more sophisticated algorithms and approaches that are not only based on the analysis of spatial information, but also use the temporal information contained in each sequence of frames [31].

In the last few decades, in order to obtain effective predictions of human actions in videos, many approaches and frameworks oriented to the construction of deep architectures have been proposed, leaving aside the hand-crafted representations and traditional video processing and classification techniques [31].

Motivated by the previous premise and inspired by a successful framework known as

multi-stream convolutional neural networks [24, 87, 88], we investigated a temporal information representation and its influence as extra source of information in videos. The visual rhythm, as described in the following chapters, is a way to represent a video sequence through an image, compacting the video information in order to be more representative and easier to process it.

1.2 Research Questions

We drive our research through some investigative questions, considering the problem of human action recognition in video sequences. Our main research questions are the following.

- Is the visual rhythm representation a useful data source to train a deep learning architecture?
- Is the spatio-temporal information extracted from the visual rhythm method useful for the action recognition problem?
- Only one RGB frame per video is sufficient to train a spatial stream?
- Is the visual rhythm stream more/less discriminative than the optical flow and RGB streams?

1.3 Objectives

This work aims to propose, implement and analyze the use of visual rhythms for human action recognition in videos sequences. Based on the followed strategy and the need for a large amount of data, we conducted experiments on two challenging datasets.

To be consistent with our goals, the following guidelines represent the focus of this work:

- Evaluation of different methods for extracting visual rhythms.
- Investigation of spatio-temporal features from the visual rhythm data.
- Investigation of spatio and temporal information from RGB and optical flow images, respectively.
- Evaluation of each individual stream¹ trained with RGB, optical flow and visual rhythm data and their contribution to the final result.
- Classification of human actions in video sequences.
- Comparison of the obtained results to state-of-the-art approaches.

¹The word stream here is referred to as a convolutional neural network.

1.4 Contributions

Even though the visual rhythm has been previously used in some image and video classification problems using hand-crafted representations (see Chapter 2 for more details), this representation has not been yet explored with deep architectures.

In this work, we demonstrate the importance of visual rhythms as a source of spatial-temporal information, achieving competitive results compared to the state of the art. In this sense, we propose an innovative and robust visual rhythm method based on the tracking of interest points of the video frames. In addition, we experiment with the combination of convolutional neural network (CNN) and recurrent neural network (RNN) architectures as feature extractor and classifier, respectively.

1.5 Text Organization

This text is organized as follows. In Chapter 2, we describe some relevant concepts and approaches related to the topic of human action recognition in videos. In Chapter 3, we present a brief description of the datasets used in our experiments, as well as the hardware and software resources used in the development of the project. In Chapter 4, we describe the proposed action recognition approaches. In Chapter 5, we report the experimental results and a comparison against the state of the art. In Chapter 6, we conclude the work with some final remarks and directions for future work. Finally, some bibliographic references associated with the problem investigated in this dissertation are presented.

Chapter 2

Background

This chapter presents some relevant concepts related to visual rhythm, optical flow, deep learning and the problem under investigation in this dissertation. Due to the importance of the visual rhythm representation in our work, the first section is dedicated exclusively to its revision, based on its antecedents, previous works, origins and techniques used for its construction. In the remainder of the chapter, we describe and discuss some concepts, techniques and applications associated with optical flow (tracking of interest points in videos), deep learning (types, architectures and transfer learning) and state-of-the-art approaches to human action recognition in video sequences as a computer vision task.

2.1 Video Sequence Understanding and XT-YT slices

Nowadays, there are robust frameworks that allow computers to reach high levels of precision, even better than human performance in the image recognition task, such as standard images that are presented in the ImageNet dataset (vehicles, musical instruments, flowers, animals, among others) [19, 29]. Furthermore, there are approaches that are even capable of recognizing distorted images, demonstrating that some problems related to this task can be better executed by a computer than by a human [20].

However, video recognition is a more complex task compared to the aforementioned one, which requires an understanding of concepts such as three-dimensional geometry. Therefore, a common way to represent a video sequence is through a 3D object (Figure 2.1(a)), varying in X , Y and T , where X and Y correspond to the spatial dimensions, and T to the temporal dimension (number of frames). Nevertheless, based on previous information, a video sequence will be defined in this work as a set of frames related to each other (Figure 2.1(b)). This last part is very important because a video sequence can only have a meaning or message as long as the spatial information of the frame i and $i + 1$ are related, where $i \in \{0, 1, \dots, T\}$ (see two types of video sequences in Figure 2.1).

Similar to the previous idea, let X, Y and T be planes of the 3D object or video sequence. Then, the XT and YT slices are defined as a portion of this volume for a certain z value in the Y or X plane for XT and YT slice, respectively. That is, the XT plane would have X along the T axis for a value $z \in Y$ (Figure 2.2(b)), whereas the YT plane would have Y along the T axis for a value $z \in X$ (Figure 2.2(c)). Therefore, we

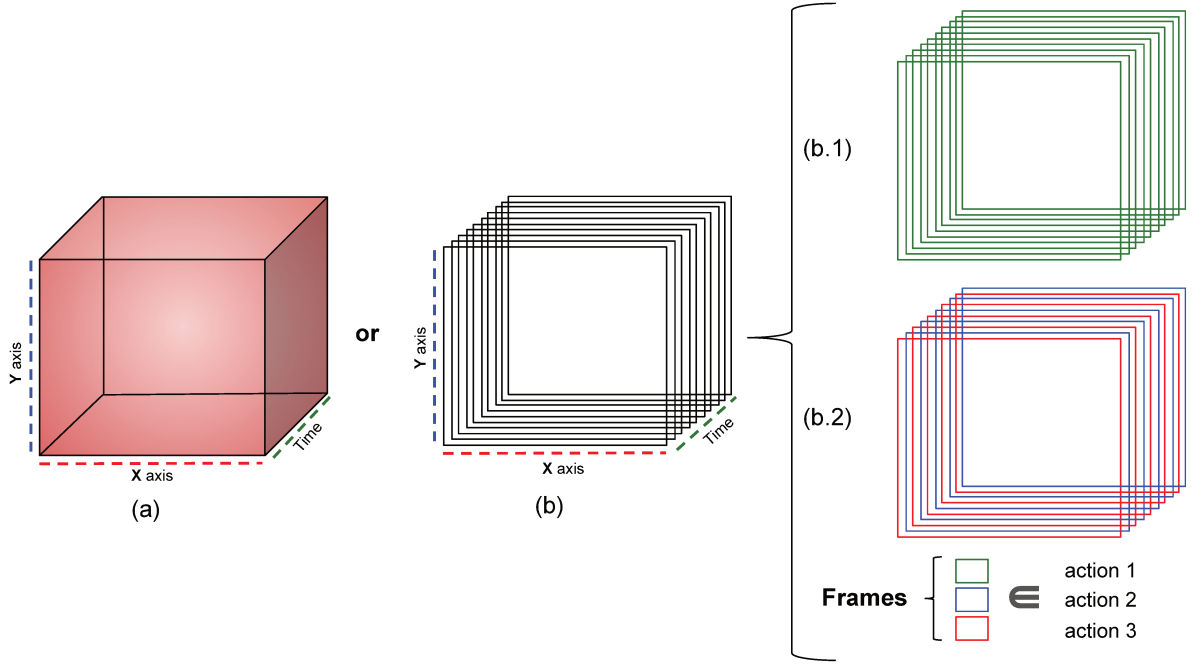


Figure 2.1: A video sequence can be considered as: (a) 3D object or (b) set of frames. In the last one, assume that (b.1) and (b.2) are two video sequences of actions that contain frames of red, green and blue types. Therefore, (b.1) has a set of frames related to each other, because they belong to the same action (green type), thus generating a video sequence with meaning or message, whereas (b.2) generates a meaningless video due to its randomness of frames (red and blue types).

are able to create slices that contain spatio-temporal information in a two-dimensional plane [77].

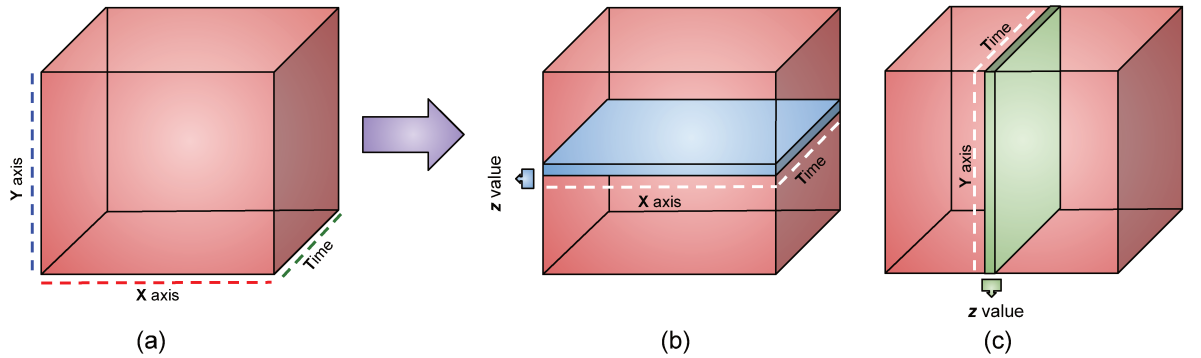


Figure 2.2: XT and YT slices are the compression of a 3D volume in a 2D plane. (a) 3D object, (b) XT slice with $z \in Y$ axis and (c) YT slice with $z \in X$ axis.

Even though the temporal slices are a proper strategy for solving certain tasks related to video analysis, such as human actions recognition (with static camera), abnormal event detection, gesture recognition and face spoofing detection [66, 77], Torres et al. [77] highlighted that it is necessary to satisfy two special assumptions in the video recording: (i) static scenario and a moving camera and (ii) static camera recording objects in movement.

In this work, we demonstrated that, even going against such requirements, the spatio-

temporal information provided by this technique can significantly benefit human action recognition tasks in video sequences.

2.2 Visual Rhythm

This section briefly reviews some concepts, related approaches and techniques used for the construction of two visual rhythm representations.

2.2.1 Literature Review

Since the beginning of the 1990s, there have been works that focused on the extraction of spatial-temporal features from videos, looking for the compression of this to reduce computational costs in its processing and analysis. A great variety of techniques have been proposed for the aforementioned purpose, for instance, the extraction of slices of a volume (video), also known as slices XT - YT . Over the years, however, researches have renamed this type of technique as visual rhythm due to the type of patterns that it presents [52, 66, 77, 93].

The concept of spatio-temporal slice was introduced by Ngo et al. [51, 52]. It consists of a set of predefined pixels sampled from a frame and arranged in a 1D image, e.g., a fixed row or column per frame. The result of this technique is a 2D image obtained from the concatenation of the slices over time. The authors proposed a method for locating video transitions and classifying them as cut, wipe or dissolve, through the analysis of vertical, horizontal and diagonal slices. This is possible because transitions in videos generally results in boundary lines in the 2D image. The shots (i.e. a video segment between two transition frames) are further subdivided according to the camera motion also based on patterns found in spatio-temporal slices, but using horizontal and vertical slices [53]. This way, by choosing proper slices, the resulting image may contain rich patterns to detect and classify events in videos. The authors also argued that, compared to other spatio-temporal features, the slices have the advantage of providing long-term information instead of encoding only a few frames.

Yeo et al. [93] is one of the first works that described a way to reduce 3D volumes to 2D, creating direct current (DC) images from motion compensated P-frames and B-frames of Moving Picture Expert Group (MPEG) compressed video.

Ngo et al. [52] proposed to use the temporal slices analysis in the detection of gradual transitions, that is, for the detection of camera cuts, wipes and dissolves, reducing a video segmentation problem to a image segmentation problem.

Almeida et al. [4] investigated a strategy for extracting visual rhythms to address the task of video caption detection. They proposed to scan each frame through a certain curve to produce a slice, demonstrating that their choice is simple and effective for detecting captions in arbitrary orientations.

Pinto et al. [66] extracted the already known XT and YT slices taking as a z value the center of each pixel on the Y and X axes, respectively, to address the face spoofing detection. They worked with the Fourier spectrum instead of directly handling the images

in the spatial domain. Therefore, the visual rhythms contained data from the frequency domain.

Almeida et al. [3] studied the fine-grained plant species identification based on encoding time series as a visual rhythm, showing that their representation is compact and suitable for long-term series.

Torres et al. [77] explored a methodology to extract a descriptor of features from visual rhythms. They evaluated their proposal in three different tasks: abnormal event detection, human action classification, and gesture recognition. It is worth mentioning that they explored different strategies for obtaining visual rhythms, such as horizontal, vertical, circular, zig-zag and random paths.

Although there is a considerable amount of researches that use of the analysis of spatial-temporal features obtained through the extraction of processing of visual rhythms, none of them used this source of information in deep architectures or deep learning in general.

2.2.2 Construction

Let $V = \{F_1, F_2, \dots, F_t\}$ be a video with t frames F_i , where each frame is an $h \times w$ matrix, and $P = \{p_1, \dots, p_n\}$ a set of 2D image coordinates. A spatio-temporal slice i is given by the $n \times 1$ column vector $S_i = [F_i(p_1) \ F_i(p_2) \ \dots \ F_i(p_n)]^T$, with $F_i(p_j)$ representing the RGB value of the point p_j in the frame F_i . Then, the visual rhythm for the entire video V following P is given by the $n \times t$ matrix:

$$VR_P = [S_1 \ S_2 \ \dots \ S_t].$$

Figure 2.3 shows an example of visual rhythm construction, where each slice corresponds to the central row of a frame. Considering the video as volume XYT , the resulting image can be seen as a plane parallel to XT .

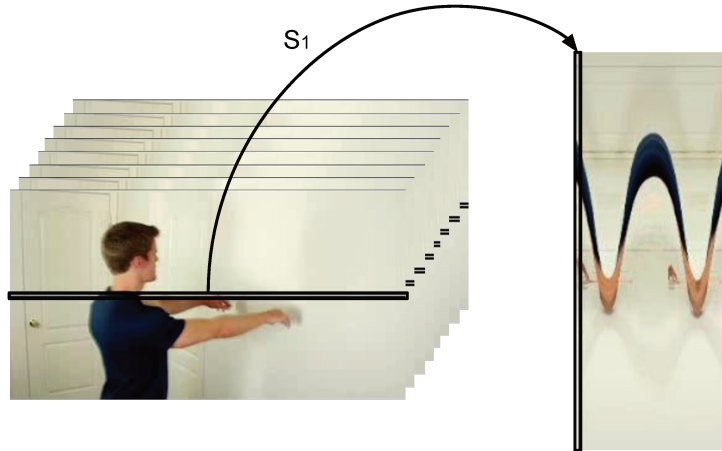


Figure 2.3: Example of visual rhythm image generated for WallPushups class from UCF101. The central row of each frame becomes a slice in the resulting image. Extracted from [17].

For encoding the entire information contained in the frames, we compute a visual

rhythm image using the following modification [71]. Let $p_j = (y_j, x_j)$ be a point in P . We define F'_i as:

$$F'_i(p_j) = \frac{\sum_y F_i(y, x_j)}{h} \quad (2.1)$$

for horizontal slices and

$$F'_i(p_j) = \frac{\sum_x F_i(y_j, x)}{w} \quad (2.2)$$

for vertical ones. In other words, $F'_i(p_j)$ is the mean intensity of the column (horizontal) or row (vertical) corresponding to p_j . Then, the modified slice S_i becomes $S_i = [F'_i(p_1) F'_i(p_2) \cdots F'_i(p_n)]^T$, as illustrated in Figure 2.4. Henceforth, we refer to these rhythms using the mean intensity as horizontal-mean and vertical-mean visual rhythms. For simplicity and for better visualization, we maintain the direction of the slices in the horizontal-mean rhythm, this way the corresponding image will have the $t \times n$ dimension instead of $n \times t$.



Figure 2.4: Modified spatio-temporal slice: the horizontal-mean/vertical-mean slice from a given frame contains the average of the columns/rows. The slices were resized for illustration purposes. Extracted from [17].

2.3 Optical Flow

The optical flow is the pattern of motion originated by the object or the camera, that is, the relative motion between an observer and a scene (see Figure 2.5). This method seeks to calculate this pattern between two consecutive frames through partial derivatives with respect to spatial and temporal coordinates, that is, it is based on local approximations of the Taylor series of the image signal [7, 22].

The optical flow technique operates under certain assumptions, such as:

1. the pixel intensities of an object do not change between consecutive frames.
2. neighboring pixels have similar motion.

Let $I(x, y, t)$ a pixel in first frame, where t described the current frame. It moves by distance $(\Delta x, \Delta y)$ in the next frame after Δt time. Since the value of the pixel is the

figs/optical_flow.pdf

Figure 2.5: Assume that images (a), (b), (c) and (d) video frames at times t , $t + 1$, $t + 2$ and $t + 3$, respectively; (d) optical flow vector of the movement of an object in the video sequence formed by the frames described previously (a ball in 4 consecutive frames).

same because it belongs to the same object (it was moved), we can define the value of pixel I as Equation 2.3.

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \quad (2.3)$$

Assuming that the movement will be small, we develop the Taylor series on the frame I to obtain Equation 2.4. From this expression, Equation 2.5 is obtained, which is reduced to Equation 2.6.

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\delta I}{\delta x} \Delta x + \frac{\delta I}{\delta y} \Delta y + \frac{\delta I}{\delta t} \Delta t \quad (2.4)$$

$$\frac{\delta I}{\delta x} \frac{\Delta x}{\Delta t} + \frac{\delta I}{\delta y} \frac{\Delta y}{\Delta t} + \frac{\delta I}{\delta t} \frac{\Delta t}{\Delta t} = 0 \quad (2.5)$$

$$f_x u + f_y v + f_t = 0 \quad (2.6)$$

where f_x , f_y , f_t are $\frac{\delta I}{\delta x}$, $\frac{\delta I}{\delta y}$ and $\frac{\delta I}{\delta t}$ respectively (derivatives of the frame I) and $\frac{\Delta x}{\Delta t}$, $\frac{\Delta y}{\Delta t}$ are the x and y components of the velocity or optical flow of I .

The concept of optical flow has been applied to various fields, such as structure from motion, video compression, video stabilization and object tracking. Thus, due to its broad field of action, several methods based on partial derivatives of the image signal have been proposed in the literature, however, the five most relevant are: Lucas-Kanade [48], Horn-Schunck [97], Buxton-Buxton [32], Black-Jepson [7] and General variational methods [23]. The first one is used for the purpose of this work, therefore, it will be explored in the following subsections in more details.

2.3.1 Lucas-Kanade Method

Lucas et al. [48] developed a differential method for optical flow estimation. It assumes that, for local neighboring pixels, the flow is essentially constant (have similar motion) and solves the optical flow equations, shown in the previous section, for the entire neighborhood using the least square criterion. For the first purpose, it is taken a 3×3 patch around each point, obtaining 9 points (f_x, f_y, f_r) with the same motion. Thus, the problem is reduced to solving 9 equations with 2 variables (u, v) .

Equation 2.7 shows the final solution obtained from Equation 2.6. Another important detail is that this method is supported with the Harris corner detector, since corners are interesting points to be tracked (see the inverse matrix in Equation 2.7).

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_i f_{xi}^2 & \sum_i f_{xi} f_{yi} \\ \sum_i f_{xi} f_{yi} & \sum_i f_{yi}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i f_{xi} f_{ti} \\ -\sum_i f_{yi} f_{ti} \end{bmatrix} \quad (2.7)$$

2.4 Deep Learning

Deep learning is a particular subset of machine learning methods using artificial neural networks. This approach is inspired by the behavior and structure of neurons in the human brain. Informally, the word *deep* refers to a large number of layers in a neural network architecture, however, this meaning has changed over time. Many references consider a network as deep without the need to use many layers.

Currently, there are many deep neural network architectures [47]. However, for the purpose of this work, we have considered two of them: convolutional neural networks (CNN) and recurrent neural networks (RNN) [39, 96].

The basic concepts and mathematical representation of the CNN and RNN architectures are described in the following subsections.

2.4.1 Transfer Learning

Many works related to deep learning do not perform the training stage from scratch, because the number of data is rarely sufficient. Instead, it is common to use pre-trained networks with large amounts of data, such as ImageNet (it contains 1.2 million images with 1000 categories) [62].

In summary, transfer learning is a very powerful deep learning technique that consists in using prior knowledge of some previously trained network. This technique has many applications in different domains [54].

Figure 2.6 illustrates two best known transfer learning strategies. The first one is to *freeze the weights* of certain layers of the network and leaving the rest open to be retrained. The second one is known as *fine tune* and consists of retraining the entire network by making a simple modification on the last layer (softmax), because the number of classes are not always the same.

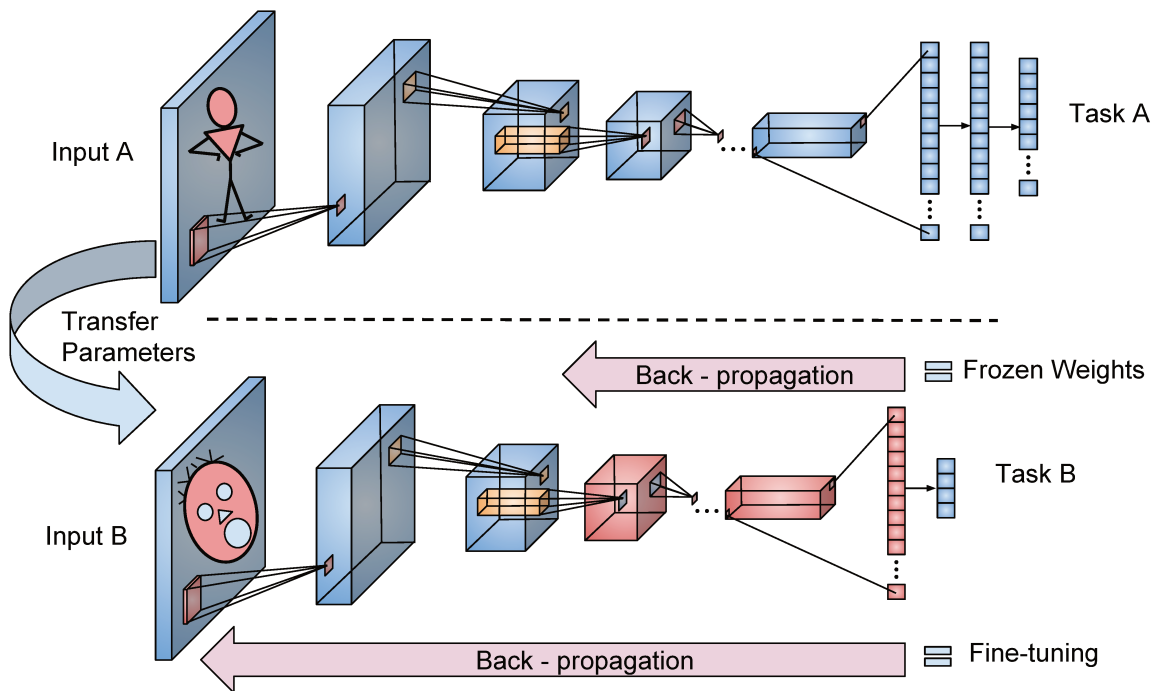


Figure 2.6: Transfer learning between two CNN architectures. The first one designed to person detection, whereas the second is designed to face detection. For the first approach, the back-propagation is performed only on the red boxes (layers). However, for the fine-tuning approach, the entire network is retraining.

2.4.2 Convolutional Neural Networks

Convolutional Neural Network (CNN) is a popular deep learning technique inspired by the organization of the animal visual cortex. Similarly to all deep learning techniques, CNN is very dependent on the size and quality of the training data [39].

A CNN consists of one or more convolutional layers with nonlinear activation functions, pooling layers and one or more fully connected layers as in a standard multilayer neural

network. The task performed by each layer is described as follows:

- *convolution layer* performs most of the heavy computational lifting. This layer is responsible for the extraction of features in images or videos through filters and convolution operators.
- *pooling layer* is a form of non-linear downsampling. This layer serves to progressively reduce the spatial size of the representation in order to decrease the number of parameters and amount of computation in the network and thus control the overfitting.
- *fully connected layer* performs the high-level reasoning of the neural network.

CNNs usually use little pre-processing of data in compared to other approaches. This means that the CNNs learn the filters, while other algorithms are hand-engineered.

To obtain a CNN architecture that is capable of dealing with input videos, a straightforward process is to simply replace 2D convolutions by 3D ones. These types of network are strongly used to perform the extraction of features from images and videos, however, the processing of these types of data is very expensive and, therefore, requires great computing power [33].

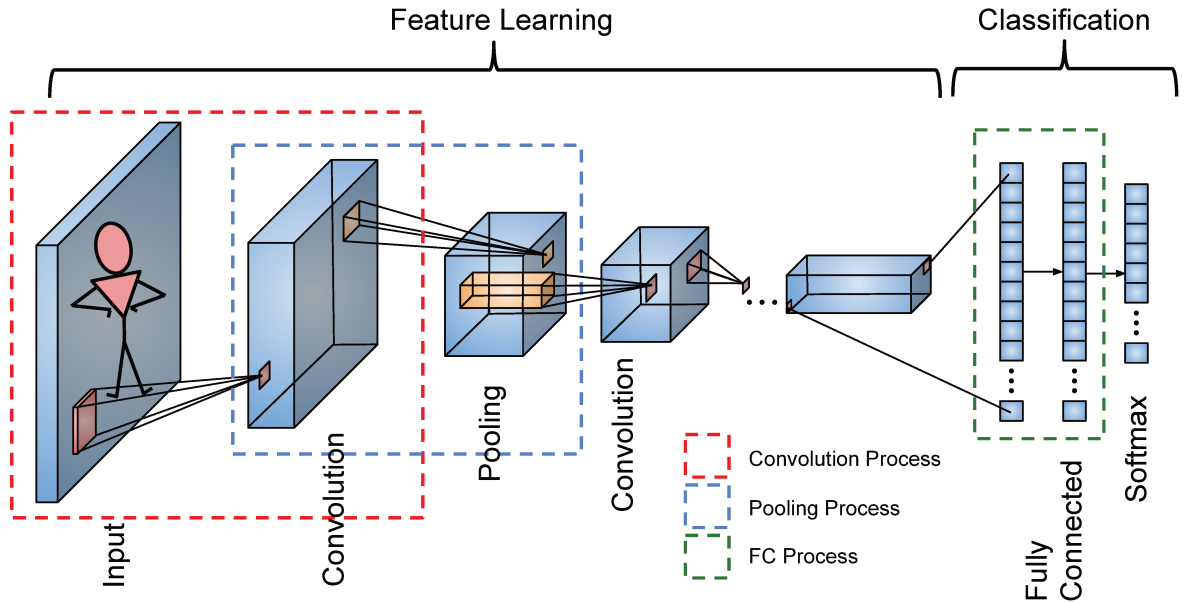


Figure 2.7: The three most important processes of a CNN architecture: the red and blue dotted boxes show the process of convolution and sub-sampling, respectively (both belong to the feature learning phase), whereas the green box involves part of the classification process (fully connected layers).

Two-Stream Architecture

In the two-stream network developed by Simonyan and Zisserman [67], a single frame is used to train the spatial stream and 10 pairs of consecutive optical flow images to train the temporal one. Although dynamic information is relevant for action recognition, static and context information such as actor poses, the objects involved and standard scenarios may help distinguish the classes. A green grass field, for instance, may be a clue for actions related to soccer games; a horse may help to recognize a horse riding action. For this reason, even using a single video frame, the spatial stream alone is capable of achieving good results.

Since spatial stream works with the same modality as image networks for classification and has a comparable goal (appearance recognition), it is reasonable that it can be pre-trained using image datasets such as ImageNet [62], followed by fine-tuning on the desired video dataset. Surprisingly, experiments indicate that the same pre-training process may be applied to the temporal stream [87].

The original network is based on AlexNet [39] (Figure 2.8). However, Wang et al. [87] argued that deeper networks, such as VGG [68] and GoogLeNet [74], are preferable to address our target problem, since the concept of action is more complex than object. Here, we explore even deeper networks: ResNet152 [30] and Inception V3 [75]. For both streams, the training data is augmented using random cropping, horizontal flipping and RGB jittering. To avoid overfitting in very deep CNNs, two additional data augmentation techniques were proposed by Wang et al. [87]: corner and multiscale cropping.

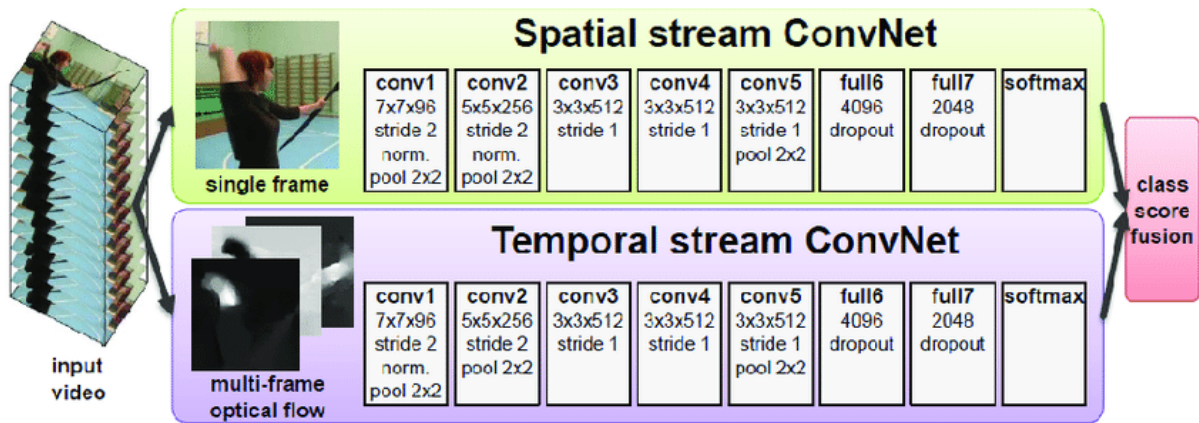


Figure 2.8: Two-stream architecture for video classification proposed by Simonyan and Zisserman [67].

For testing, 25 frames/stacks of optical flow images are selected from each video and used to produce 10 new samples per frame/stack by cropping and flipping techniques. Each sample is individually tested in the corresponding stream. Finally, the class scores computed in each CNN (softmax scores) are combined through a weighted average.

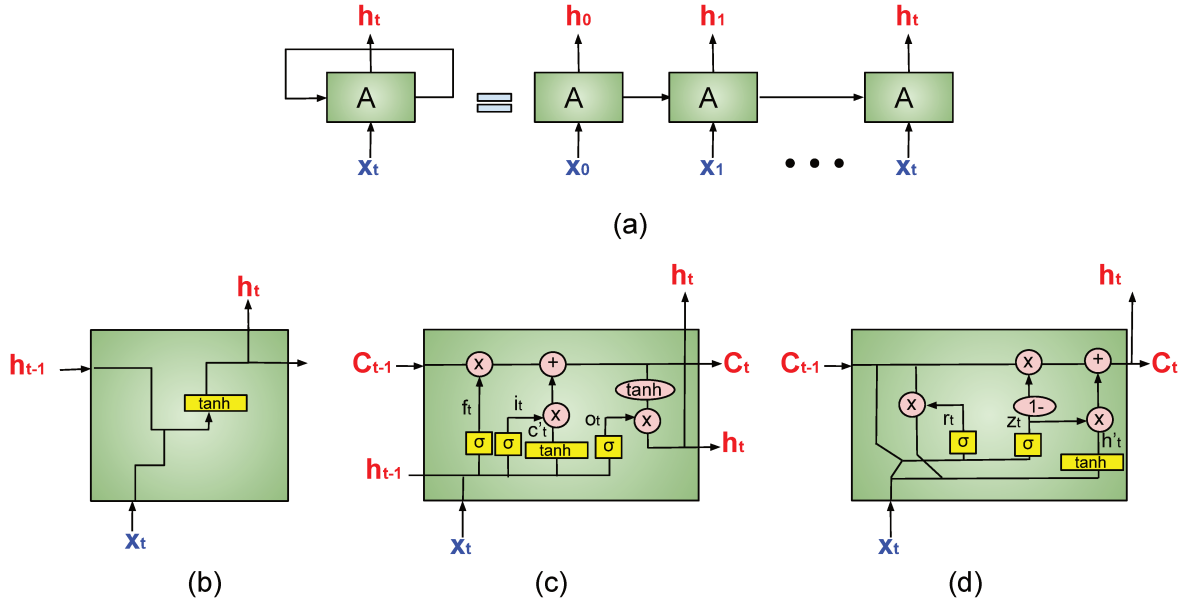


Figure 2.9: Figure (a) shows an unrolled recurrent neural network, where X_i is some input that outputs a value h_t , whereas (b), (c) and (d) show a chunk of the basic RNN, LSTM and GRU architectures.

2.4.3 Recurrent Neural Networks

In the last years, Recurrent Neural Networks (RNN) have proven to be a very powerful technique for extracting temporal features in analysis and classification of videos, which is complex to perform through traditional techniques. In essence, RNNs are neural networks that employ recurrence. This architecture is able to learn tasks which involve short time intervals between inputs, however, this memory usually becomes insufficient when dealing with real-world problems (for instance, video sequences) and, like most neural networks, the vanishing gradient problem is present.

In order to alleviate these problems, Gers et al. [25] proposed a specific recurrent architecture, namely Long Short-Term Memory (LSTM). These networks use a special node, called Constant Error Carousel (CEC), that allows for constant error signal propagation through time [5]. A variation on the LSTM is the Gated Recurrent Unit (GRU), introduced by Cho et al. [15]. Figures 2.9(c) and (d) show a chunk of these architectures, respectively.

The following equations are the mathematical representation of the two previous architectures mentioned, Equation 2.8 for LSTM and Equation 2.9 for RGU.

$$\begin{aligned}
 i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
 C'_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\
 C_t &= f_t * C_{t-1} + i_t * C'_t \\
 o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
 h_t &= o_t * \tanh(C_t)
 \end{aligned} \tag{2.8}$$

$$\begin{aligned}
z_t &= \sigma(W_z \cdot [h_{t-1}, x_t]) \\
r_t &= \sigma(W_r \cdot [h_{t-1}, x_t]) \\
h'_t &= \tanh(W \cdot [r_t * h_t - 1, x_t]) \\
h_t &= (1 - z_t) * h_{t-1} + z_t * h'_t
\end{aligned} \tag{2.9}$$

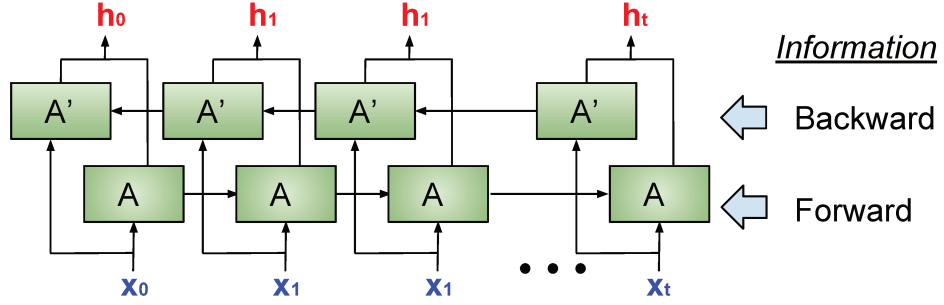


Figure 2.10: Shows an unrolled Bi-directional recurrent neural network architecture, where X_i is some input, that outputs a value h_t .

Bi-Directional Recurrent Neural Networks

Bi-Directional Recurrent Neural Networks were introduced by Graves et al. [28] and basically consist of putting two independent RNN together. Nevertheless, the input sequences are not fed the same way for both, that is, the first RNN receives the data in the original order while the second one in the reverse, thus allowing both backward and forward information about the sequence. Finally, the output is the concatenation of both previous outputs at each time step (Figure 2.10).

2.5 Human Action Recognition

In the Oxford Dictionary, action is defined as *the fact or process of doing something, typically to achieve an aim.* and activity is *a thing that a person or group does or has done.* However, in this work, the term action covers the two concepts mentioned above, this to maintain a standard title of related works in the literature.

This section briefly reviews some concepts and related approaches to the human action recognition problem.

2.5.1 Problem Definition

Videos have been used for many tasks in our daily lives, such as surveillance, health monitoring, and entertainment. In general, a human operator has to constantly examine video sequences to identify events of interest. However, this procedure is very time-consuming and susceptible to failure. Moreover, an increasing amount of data has been

produced and released every day, making the process sometimes impracticable. In many real-world scenarios, video output is stored without any further processing due to the massive amount of data involved. Therefore, automatic procedures are needed to extract useful information from videos.

Although many studies have been conducted in the literature to recognize human activities in video sequences, there is no generic methodology for solving the problem and many questions remain open. Some challenges include the diversity of actions present in the scenes, modeling spatio-temporal relations, understanding of interactions among persons and objects, difficulties related to scene conditions such as occlusions, background clutter, camera motion, lighting conditions, among other factors.

2.5.2 Literature Review

Before the use of Deep Learning techniques for the problem of human action recognition, several approaches based on traditional strategies using hand-crafted features were developed. They basically have two main components: representation and classification of actions. The action representation focuses on converting a video into a feature vector that is subsequently used to perform the classification step. Traditionally, since these two processes have been performed separately, there was a lack of end-to-end architectures to address the problem, which is efficiently performed by most of the deep-learning techniques [31].

Bobick [10] presented an approach based on the representation and recognition of the actor movement during the performance of actions by encoding motion information through a simple image. For this purpose, Motion Energy Image (MEI) and Motion History Image (MHI) were employed, where MEI is a binary image that describes where the movement occurs, whereas MHI shows how the image is moving. Due to the useful contextual information extracted by MEI and MHI representations, many other works based on this information were proposed. Tian et al. [76] extracted gradient from MHI to filter out moving and cluttered background. Blank et al. [9] introduced a mechanism for performing the volumetric extension of the MEI images. Weinland et al. [90] represented MHI images through spatio-temporal volumes.

Local representation in images usually follows a point detection pipeline, that is, local descriptor extraction and local descriptor combination. However, several works were proposed to extend this approach to video sequences. Laptev [42] extended the Harris corner detector to 3D Harris. Traditional 2D Harris corner detector focuses on finding spatial locations in an image with significant changes in two orthogonal directions, whereas 3D Harris approach identifies points with large spatial variations and non-constant motions. Klaser et al. [37] proposed the 3D Histogram of Gradient Orientations (HoG3D) as a motion descriptor, which is spanned to the spatio-temporal domain. Laptev et al. [44] developed the Histogram of Optical Flow (HoF) as a spatio-temporal descriptor over local regions. Dalal et al. [18] introduced the Motion Boundary Histogram (MBH), a more robust extension of the HoF descriptor. Zhao and Pietikainen [98] proposed the Volume Local Binary Patterns (VLBP), an extension of the Local Binary Pattern (LBP) descriptor, where the main idea consists of encoding local volumes through the histogram of binary

patterns.

Associated with action representation, various traditional action classifiers, such as Support Vector Machine (SVM) [43], K-Nearest Neighbors (KNN) [9], were applied to recognize actions using the feature vector previously obtained. A Bag-of-Word (BoW) model encodes the distribution of local motion patterns using a histogram of visual words [49].

Torres and Pedrini [77] explored these 2D images, referred to as visual rhythm [81], to tackle three computer vision problems: abnormal event detection, human action recognition and hand gesture recognition. Visual rhythm is used to estimate object trajectories throughout the video. Slices that capture leg motion, for instance, produce an interesting braided pattern in actions as walking. For the action recognition problem, high-pass filters are applied, followed by the selection of regions of interest (ROI) to keep only the information relating to the trajectory. Since the representation process is entirely hand-crafted, this approach belongs to the first group (traditional methods). In our method, the sequence of hand-crafted processes that extract information from visual rhythm images are replaced by a 2D CNN (Convolutional Neural Network). Thus, the network automatically learns relevant patterns to describe actions.

Most of the recent action recognition approaches employ deep learning since it has shown to be a useful tool to generalize data in complex scenarios, achieving impressive results in different computer vision problems, especially in image classification [14, 30, 39, 68]. According to Herath et al. [31], in general the main drawback of video networks (that is, CNNs composed of 3D filters) is the rigid temporal structure. The architectures usually require a fixed and small number of frames which does not take into account the duration of the action. Moreover, the higher training cost of 3D extensions caused by the number of trainable parameters and the absence of large video datasets compared to image ones have led the researchers to explore image networks for videos.

In order to create a more sophisticated feature extractor, Tran et al. [78] proposed an approach to spatio-temporal feature learning using 3D deep convolutional networks (3D ConvNets) with a simple linear classifier. Such deep feature descriptor presents important properties, such as generality, compactability, simplicity and efficiency. The feature learned by a linear classifier can produce high performance on various video analysis tasks. The classification model consists of extracting features with 3D ConvNets and inputs them to a multiclass linear SVM for training models. However, despite being powerful in the extraction of features, the computational power required to perform this approach is typically very expensive and lacks an end-to-end architecture for training.

Temporal information may be incorporated at different stages of the process. Several works explore 2D CNN to capture only static information (frame-level feature extractor), and incorporate motion in the fusion stage [34, 35, 59]. Other approaches use hand-crafted inputs for early incorporation of the dynamics, achieving higher accuracies. The two-stream network [67] is composed of two parallel CNNs individually trained, working with different image modalities. The first one, the spatial stream, receives a single RGB frame randomly sampled from the video representing appearance information. To capture motion information, the temporal stream has as input 20 stacked optical flow images, 10 for each direction (horizontal and vertical). Although the combination of complementary information achieves promising results, the short temporal extension encoded in the inputs

is once more an issue.

To consider longer temporal evolution, Ng et al. [50] repeated the feature extraction process from the two-stream network [67] for several frames and stacks of optical flow images on a given video. They considered two different feature aggregation methods: pooling layers and LSTM (Long Short-Term Memory) cells. This architecture is capable of processing up to 120 frames per video. A similar approach is proposed by Wang et al. [88], however, for each input (RGB frame or stack of optical flow images) the network outputs a preliminary prediction of the action classes instead of features. The predictions are then fused using a segmental consensus function that does not impose temporal limits.

Song et al. [69] proposed an end-to-end spatio-temporal attention model from skeleton data. Their work built a model using Recurrent Neural Networks (RNNs) with Long Short-Term Memory (LSTM), which learns to selectively focus on discriminative joints of skeleton within each frame of the inputs. The network is designed to automatically select dominant joints through the spatial attention module and assign different degrees of importance to different frames through the temporal attention module.

An alternative strategy consists in exploring additional long-term input modalities. Wang et al. [84] proposed a three-stream network, adding a new stream to the network from [67]. The third stream receives as input dynamic images that encode simultaneously appearance and motion information along 20 consecutive frames. Compared to original RGB and optical flow, it represents a long-term input and its inclusion improves network performance. Here, we explore a longer-term modality, the visual rhythm, that encodes information from the entire video in a single image, so the network has access to the complete slice evolution at once to learn patterns. An important contribution of our work is how to select proper slices that presents useful patterns for the 2D CNN.

Some more sophisticated works pre-trained their networks using the large Kinetics dataset, which contains 400 human action classes and over 400 clips per class. Based on the 2D ConvNets, Carreira et al. [12] introduced a new Two-Stream Inflated 3D ConvNet (I3D), where pooling kernels and filters are expanded into 3D, making it possible to learn spatio-temporal feature extractor from videos while leveraging well-known ImageNet architectures and their parameters. Choutas et al. [16] introduced a novel representation that encodes the movement of some semantic keypoints. They used human joints as keypoints in a scheme known as pose motion (PoTion) representation. They extracted heatmaps for the human joints from each frame and used a shallow convolutional neural network to classify the actions. Popular deep models for action recognition in videos generated independent predictions for short clips, which were then pooled heuristically to assign an action label to the full video segment. Wang et al. [85] proposed discriminative pooling based on the notion that, among the deep features generated on all short clips, there is at least one that characterizes action. The method learned a hyperplane that separates this unknown, yet discriminative, feature from the remaining parts.

Chapter 3

Materials

In this chapter, we present the datasets used in the experiments to validate our results, as well as the hardware and software resources employed in the implementation.

3.1 Action Datasets

Although there are many dataset available for this problem, only some have large amounts of data that allow a proper training of deep networks (Table 3.1). In our experiments, we used the two most challenging datasets, HMDB51 and UCF101, to evaluate the effectiveness of our action recognition approach. However, despite there are other larger and more varied dataset than the previous two, they were not considered due to the intensive computing power required.

| Datasets | Year | Videos | Actions | Modality | Environment |
|--------------------|-------------|---------------|------------|------------|---------------------|
| KTH [64] | 2004 | 599 | 6 | RGB | Controlled |
| Weizmann [9] | 2005 | 90 | 10 | RGB | Controlled |
| INRIA XMAS [90] | 2006 | 390 | 13 | RGB | Controlled |
| IXMAS [95] | 2006 | 1,148 | 11 | RGB | Controlled |
| UCF Sports [61] | 2008 | 150 | 10 | RGB | Uncontrolled |
| Hollywood2 [49] | 2009 | 3,669 | 12 | RGB | Uncontrolled |
| UCF11 [46] | 2009 | 1,100+ | 11 | RGB | Uncontrolled |
| HMDB51 [41] | 2011 | 7,000 | 51 | RGB | Uncontrolled |
| UCF50 [60] | 2012 | 50 | 50 | RGB | Uncontrolled |
| UCF101 [70] | 2012 | 13,320 | 101 | RGB | Uncontrolled |
| CAD-120 [38] | 2013 | 120 | 10 | RGB-D | Controlled |
| Sports-1M [35] | 2014 | 1,133,158 | 487 | RGB | Uncontrolled |
| YouTube-8M [1] | 2016 | 8,000,000 | 4,716 | RGB | Uncontrolled |
| Kinetics [36] | 2017 | 500,000 | 600 | RGB | Uncontrolled |

Table 3.1: Most used datasets in the human action recognition problem in videos.

In the following subsections, we briefly describe each dataset, including details about its number of classes and video clips.

3.1.1 UCF101 Dataset

UCF101 is a dataset of realistic action videos, collected from YouTube, with 13320 videos from 101 action categories, each grouped into 25 groups, where each group can consist of 4-7 videos of an action. The videos from the same group may share some common features, such as similar background, viewpoint, etc.

The samples have a fixed resolution of 320×240 pixels, frame rate of 25 FPS (frames per second) and various lengths. The dataset also includes recommended three splits, where each of them contains approximately 70-30 for training and testing, respectively. The validation protocol consists of evaluating each split individually, then the final result is the average of the three. UCF has proposed a few datasets (UCF11 [46], UCF50 [60] and UCF101 [70]) for human action recognition. UCF101 is an extended version of UCF50 and UCF11 [70]. Some examples are shown in Figure 3.1.



Figure 3.1: Frames of some videos from the UCF101 dataset. Extracted from [70].

3.1.2 HMDB51 Dataset

This dataset was collected from various sources, mostly from movies, and a small proportion from public databases, such as the Prelinger archive, YouTube and Google videos. The dataset contains 6849 clips divided into 51 action categories, each containing a minimum of 101 clips. The actions categories can be grouped in five types: General facial actions, facial actions with object manipulation, general body movements, body movements with object interaction and body movements for human interaction [40].

Since it combines commercial and non-commercial sources, it presents a rich variety of sequences, including blurred videos or with lower quality and actions from different points of views. The authors also provide a recommended three splits of the samples, where each split contains 70 samples for training and 30 for testing per action class. The validation protocol follow the same steps of the previous dataset. Some video clips are shown in Figure 3.2.

3.2 Computational Resources

We used convolutional and recurrent neural network in several experiments conducted in this work. Graphics processing units (GPUs) are suitable for performing operations,



Figure 3.2: Frames of some videos from the HMDB dataset. Extracted from [40].

such as multiplication of matrices, which are carried out in the CNNs and accelerate the process of training and testing of these types of architectures.

Our method is based on the PyTorch implementation of the very deep two-stream network [87] provided by Zhu [101]. All experiments were performed on a machine with an Intel® Core™ i7-3770K 3.50GHz processor, 32GB of memory, an NVIDIA GeForce® GTX 1080 GPU and Ubuntu 16.04.

Our adaptive visual rhythm approach was implemented in Python programming language. The most important libraries used in the code development were Numpy, Scipy, Scikit-Learn and OpenCV, which provided mechanisms for image processing, interest point tracking, matrix manipulation, and generation of confusion matrices.

Chapter 4

Proposed Action Recognition Method

In this chapter, we described the proposed methodology for action recognition in video sequences [17]. Section 4.1 presents an overview of the methods, where each stage of our two approaches are described, both focused on the adaptive visual rhythm (AVR) representation. Section 4.2 reports the AVR in more details and explains how the representation is built. Section 4.3 presents our multi-stream convolutional neural network, highlighting the improvement over the spatial stream proposed by previous works [67, 87] and also describes all the used streams. Finally, Section 4.4 reports our LSTM stream and the weighted average fusion used in both approaches.

For the human action recognition problem, two main sources of information to be considered are spatial and temporal. Therefore, our methodology is focused on the extraction and evaluation of both information. In addition, the spatio-temporal information (visual rhythm representation) is explored and analyzed to demonstrate its importance and effectiveness in the action recognition problem.

We implemented a method for human action recognition in videos based on deep learning using CNN and RNN architectures in conjunction with hand-crafted techniques for the pre-processing of the data (videos). This combination of two approaches have been addressed more recently (traditional computer vision techniques with deep learning), achieving levels of precision competitive to those of the state-of-the-art methods.

4.1 Methodology Overview

In this work, we propose two main approaches, the second based on the first one (see the next subsections for more information). Figure 4.8 (seen later in the text) illustrates the first one, which basically follows two main stages: one of them consists of fine-tuned three CNNs (ResNet152 [30]/Inception V3 [75]) with RGB, optical flow and visual rhythm images respectively, whereas the other is responsible for the fusion of their results (softmax layers) through a weighted average strategy (see Section 4.3 and Section 4.2 for more details).

Figure 4.13 (also seen later in the text) illustrates the second approach, that basically is a extension of the first one. The three CNNs are trained in the same way, however, in addition to that, the RGB and optical flow streams are used as feature extractors. Next,

the feature vectors obtained from the two previous streams are concatenated and used as data to feed an RNN. Finally, the softmax layer of the RNN, optical flow and visual rhythm streams are merged to obtain the final result through the same strategy of the previous approach.

Despite having two slightly different approaches, the core of this work is based on our *adaptive visual rhythm*. The use of this type of data as a source of spatio-temporal information, along with our innovative proposal for CNN training of the visual rhythm stream, demonstrates that this type of information is useful and complementary for the well-known two-stream approach [67, 87].

4.2 Visual Rhythms

This section describes the core technique explored in this work. This is a representation that allows us to encode videos into images, facilitating the access to the spatio-temporal information through patterns represented in these images.

4.2.1 Visual Rhythm Construction

As mentioned in the literature review (Section 2.2), there are two types of strategies for the construction of visual rhythms, both focused on the creation of a row or column from each frame of the video, but with slight differences. The first is based on the path that must be tracked, such as horizontal, vertical, circular, zig-zag and random path (See Figure 4.1), whereas the second one is more oriented to the selection of a small number of pixels, such as diagonal pixels, and to the information compression, such as the choice of a single row or column, or the average of the pixels of each of these ones (See Figure 4.2).

The base technique used in our proposal for the construction of the visual rhythm is the compression of information, that is, for each frame i of the video sequence v , we get a row or column $r_i = \{p_{i,0}, p_{i,1}, p_{i,2}, \dots, p_{i,w}\} / c_i = \{q_{0,i}, q_{1,i}, q_{2,i}, \dots, q_{h,i}\}$, where w, h are the dimensions of the frame and $p_{i,k}/q_{k,i}$ is the mean of all pixel values in column/row k of frame i . This results in the visual rhythm image $VR_r = \{r_0, r_1, r_2, \dots, r_t\} / VR_c = \{c_0, c_1, c_2, \dots, c_t\}$, which, from this point, it will be referred to as horizontal-mean and vertical-mean, respectively, where t corresponds to the number of frames (See Figure 4.2). It is worth mentioning that the resulting row/column i is the row/column i of the visual rhythm image.

4.2.2 Resizing of Visual Rhythm Images

Resnet [30] and Inception V3 [75] are networks that have defined the dimensions of their input images that are used for their training. Due to this fact, it is necessary to generate visual rhythm images that satisfy this requirement, in such a way that the information saved by each one is maintained without losing details that may influence its pre-processing and extraction of features.

As shown in Figure 4.1, one of the dimensions of the visual rhythm resulting from a video depends on the number of frames it has, therefore, short videos in time may be a

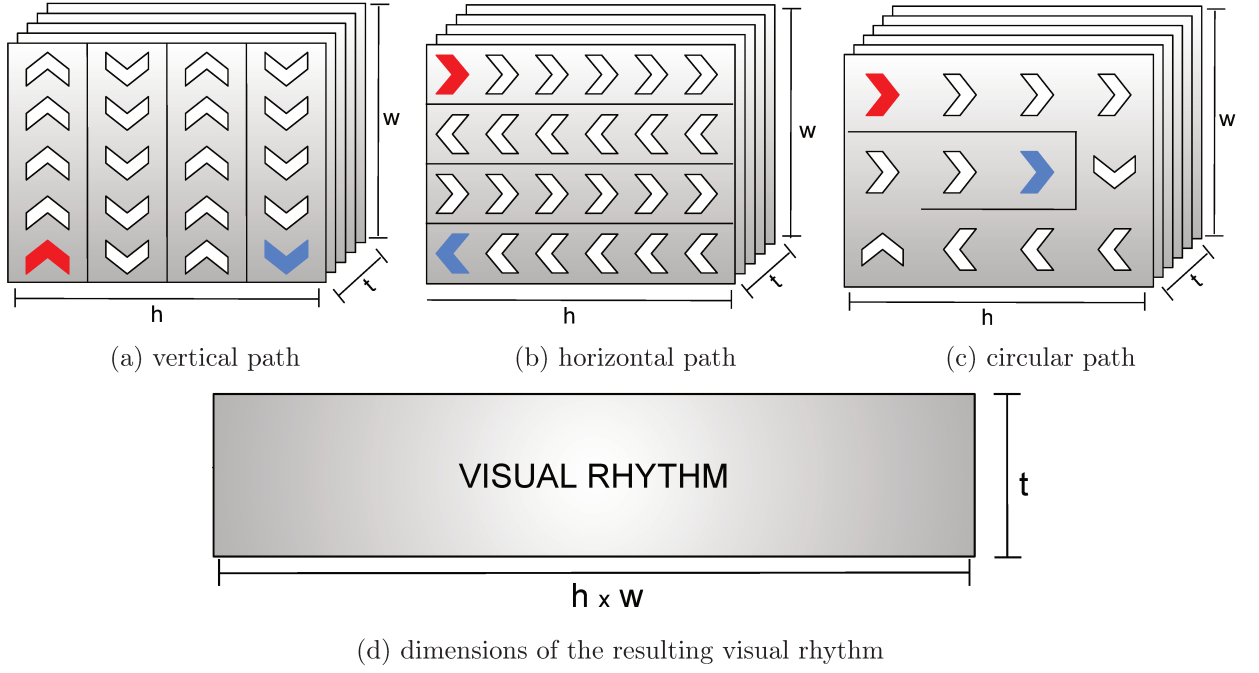


Figure 4.1: Examples of visual rhythm construction using the tracking a certain path, such as (a) vertical, (b) horizontal and (c) circular.

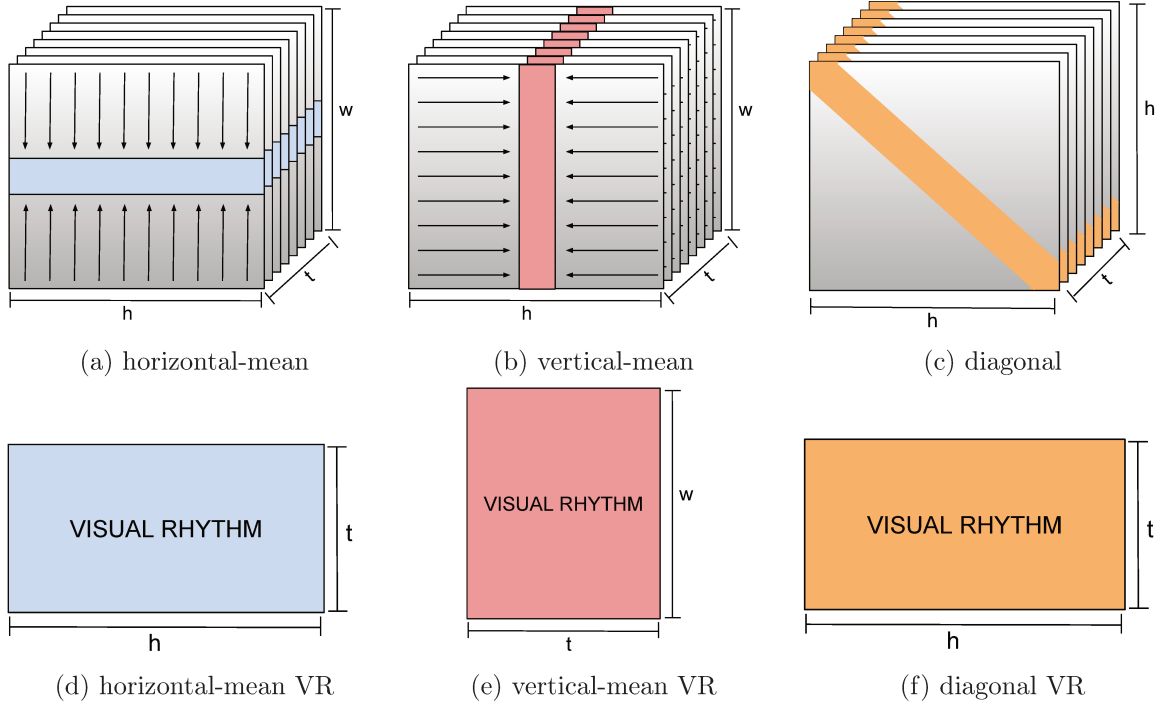


Figure 4.2: Examples of visual rhythm construction using the diagonal pixels and compression of the information through the mean operation.

problem, since their dimensions do not satisfy the required size due to the small number of frames. Let $n' \times t'$ be the required dimensions. In this work, we opted to replicate $(\lfloor \frac{t'}{t} \rfloor + 1)$ times the first $(t' \bmod t)$ frames, and $(\lfloor \frac{t'}{t} \rfloor)$ the remaining ones. In long videos, we keep only the first t' frames. This technique is performed before the visual rhythm

calculation.

4.2.3 Analysis of Vertical-Mean and Horizontal-Mean Visual Rhythms

Initially, multiple techniques for the creation of visual rhythm images were explored, training Resnet [30] and Inception V3 [75] networks with these data. However, the results showed higher precision levels using the vertical-mean and horizontal-mean visual rhythms (see Section 5), where one was slightly higher than the other.

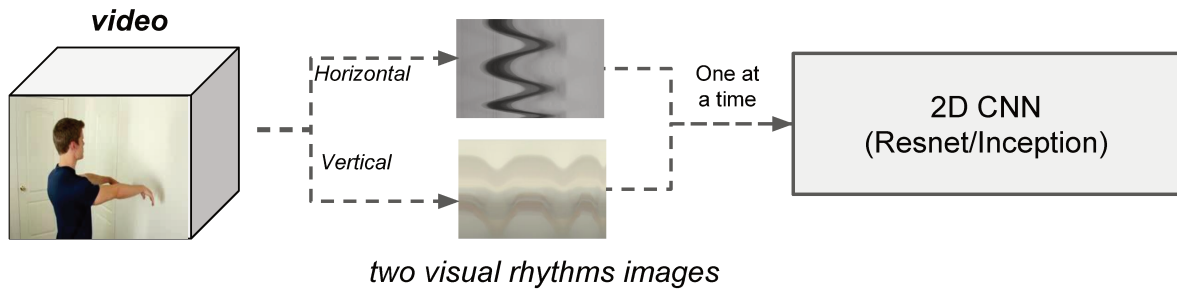


Figure 4.3: Horizontal-mean and vertical-mean visual rhythm images as a unique dataset.

Since both techniques presented the best results among all the others, we decided to join their visual rhythm images generated as a single dataset for the training of the aforementioned networks, that is, networks were trained with two different visual rhythm images by video (horizontal-mean and vertical-mean) (Figure 4.3). This assumes that an improvement would be obtained because there would be more than one type of information and, therefore, more data to be used in the training stage. However, the results did not improve, on the contrary, they worsened considerably, which led us to analyze the individual behavior of these two techniques in order to enhance their precision rates.

Our first attempt to discover and understand their individual behavior was based on simulating their creation using two possible videos composed of white background and a black square figure that moves from left to right and down to up, respectively (Figure 4.4). This was an important resource, because it showed us relevant information that could be considered and a possible explanation of why it worsened the results by joining both data for training.

By analyzing in Figure 4.4 two videos where the actor (black square) and background (white) are the same or similar, but the direction of the movement is different, it is possible to observe some interesting behaviors. For example, we can notice that, in the horizontal-mean visual rhythm of both videos, a vertical movement predominates, that is, there are small concatenated rectangles, where each one represents a line or column (depending on the type of visual rhythm used) obtained from a video frame, where clearly the displacement is more vertical than horizontal. The opposite occurs with the vertical-mean visual rhythm. Then, when training a network with both data, this learns that, regardless of the original direction of the actor in the video, it will have a vertical and

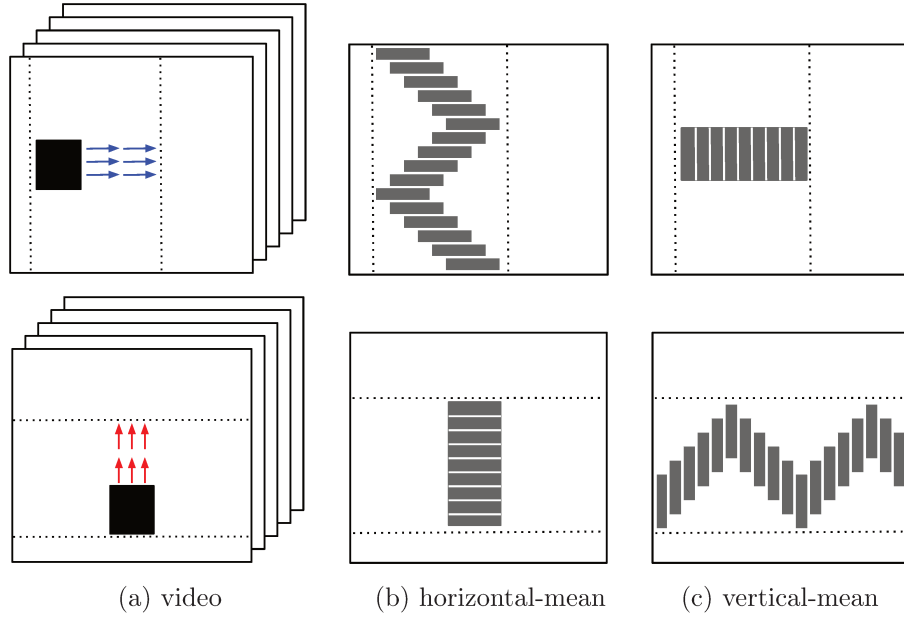


Figure 4.4: Examples of modified visual rhythms for a white background video and a black square figure that moves according to the direction of the arrows drawn next to it.

horizontal movement that is represented in the horizontal-mean and vertical-mean visual rhythm images, respectively, which confuses it, so that the accuracy obtained is smaller than individually.

An important question is to understand what would happen if, for the first and second videos, we used as training data only the vertical-mean or horizontal-mean visual rhythm image, respectively. We will revisit this issue in Subsection 4.2.4.

Figure 4.6 shows examples of horizontal-mean and vertical-mean visual rhythms for *TrampolineJumping* and *WallPushups* classes from UCF101 dataset. It is worth mentioning that the predominant direction of the movement affects the resulting rhythm.

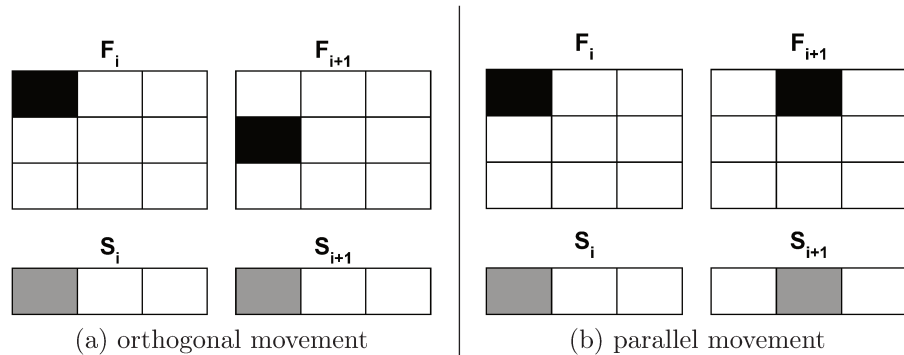


Figure 4.5: A moving object considering two consecutive frames and horizontal-mean slices. Parallel movement is better captured in the slice.

From Figure 4.6, we mentioned the relation between the predominant movement and the resulting rhythm. Consider a point $p_j \in P$, the set $\{F'_1(p_j), F'_2(p_j), \dots, F'_t(p_j)\}$ represents the variation in the average value regarding p_j across the time and can be seen in the columns/rows of the horizontal-mean/vertical-mean rhythm. If the mean value remains constant in p_j , the corresponding column/row will form a line with homogeneous

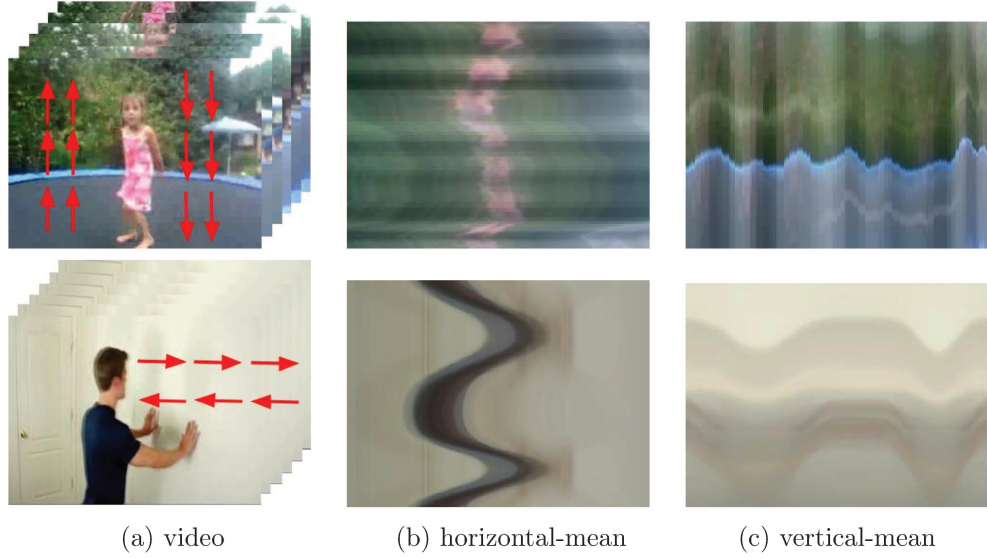


Figure 4.6: Examples of modified visual rhythms for TrampolineJumping and Wall-Pushups classes from UCF101 dataset. Red arrows indicate the predominant direction of the action. Extracted from [17].

intensity. Assuming, without loss of generality, that we are working with horizontal-mean slices. If a given object moves vertically (that is, orthogonally to the slice direction) between two frames, it is very likely that the mean color of the corresponding column remains the same (Figure 4.5). However, a horizontal movement affects the average color of all columns spanned by the object. Therefore, movements parallel to the slice direction tend to produce more distinctive patterns.

4.2.4 Adaptive Visual Rhythm

The previous subsection described why using both types of visual rhythms as a single training dataset is not recommended. Therefore, in this subsection, we propose a technique that allows us to make a decision to generate the most appropriate visual rhythm image for a video (horizontal-mean or vertical-mean visual rhythm image), based on the direction of movement that is predominant.

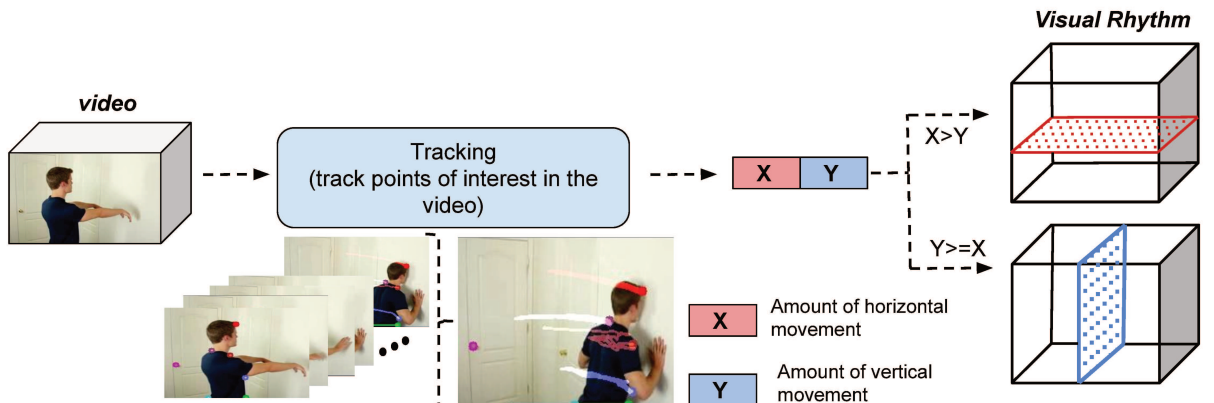


Figure 4.7: Construction process of the adaptive visual rhythm. Extracted from [17].

As explained in subsection 4.2.3, Figure 4.4 clearly shows that the visual rhythm image that best adapts to a video depends on the direction of its most predominant movement, that is, if the horizontal movement is greater than the vertical, the vertical-mean visual rhythm will be the best choice, otherwise, the horizontal-mean visual rhythm (Figure 4.7). Based on this observation, we propose a decision algorithm (Algorithm 1) named Adaptive Visual Rhythm (AVR) to define only one visual rhythm direction for a given video using the tracking of its interest points.

Algorithm 1 *Decision*(V)

Input Video $V = \{F_1, F_2, \dots, F_t\}$.

Output Visual rhythm direction {1: vertical-mean; 2: horizontal-mean}.

```

1:  $H \leftarrow 0$                                 ▷ Initialize the accumulated horizontal displacement.
2:  $V \leftarrow 0$                                 ▷ Initialize the accumulated vertical displacement.
3:  $P_a \leftarrow \text{goodFeaturesToTrack}(F_1)$         ▷ Find corners in  $F_1$ .
4: for each  $F_i \in V - \{F_1\}$  do:
5:    $P_b, St \leftarrow \text{PyrLK}(F_{i-1}, F_i, P_a)$     ▷ Pyramidal Lucas-Kanade point tracking.
6:    $P_a, P_b \leftarrow \text{SelectGoodPoints}(P_a, P_b, St)$   ▷ Select good points
7:    $H \leftarrow H + \sum_{j=1}^n |P_b[j].x - P_a[j].x|$     ▷  $n$  = size of  $P_a$ .
8:    $V \leftarrow V + \sum_{j=1}^n |P_b[j].y - P_a[j].y|$ 
9:    $P_a \leftarrow P_b$ 
10: if  $H \leq V$  then
11:   return 1                                ▷ Vertical movement is predominant.
12: else
13:   return 2                                ▷ Horizontal movement is predominant.

```

The AVR algorithm consists in estimating the total movement in each direction using Lucas-Kanade vectors, such that the highest value defines the rhythm direction. First, the function *goodFeaturesToTrack()* is used to extract Shi-Tomasi [65] interest points in the first frame. It returns a set P_a containing the selected points. At each iteration, *PyrLK()* tracks the reference points from P_a in the frame F_i , returning the corresponding points P_b along with flags indicating if they were found in F_i . We use the pyramidal version [11] of the Lucas-Kanade tracker [48]. The flags are used by the *SelectGoodPoints()* function to filter out some points in P_a and P_b , keeping only the points that were found by the tracker.

The absolute horizontal and vertical displacement given by P_a and P_b are accumulated in two scalars H and V , respectively. The points from P_b become the reference for the next search. Finally, H and V are compared to choose the most suitable visual rhythm. We use the vertical-mean response if $H \leq V$ and horizontal-mean otherwise. The *goodFeaturesToTrack()* and *PyrLK()* routines correspond to *goodFeaturesToTrack()* and *calcOpticalFlowPyrLK()* from OpenCV. The method is depicted in Figure 4.7.

4.3 Multi-Stream Convolutional Neural Network

Our network consists of three streams (Figure 4.8): (i) an improved spatial stream, (ii) the temporal stream, and (iii) a new spatio-temporal stream. The inputs are, respectively, two RGB frames per video (one at time), using a random choice in the first and second halves of the video frames, a stack of optical flow images and a single visual rhythm image computed from the video. Each stream is individually trained as proposed by Simonyan and Zisserman [67] using the parameters from ImageNet training as initialization, applying a fine-tuning and all the training data is augmented using multiscale and corner cropping [87] and random horizontal flipping.

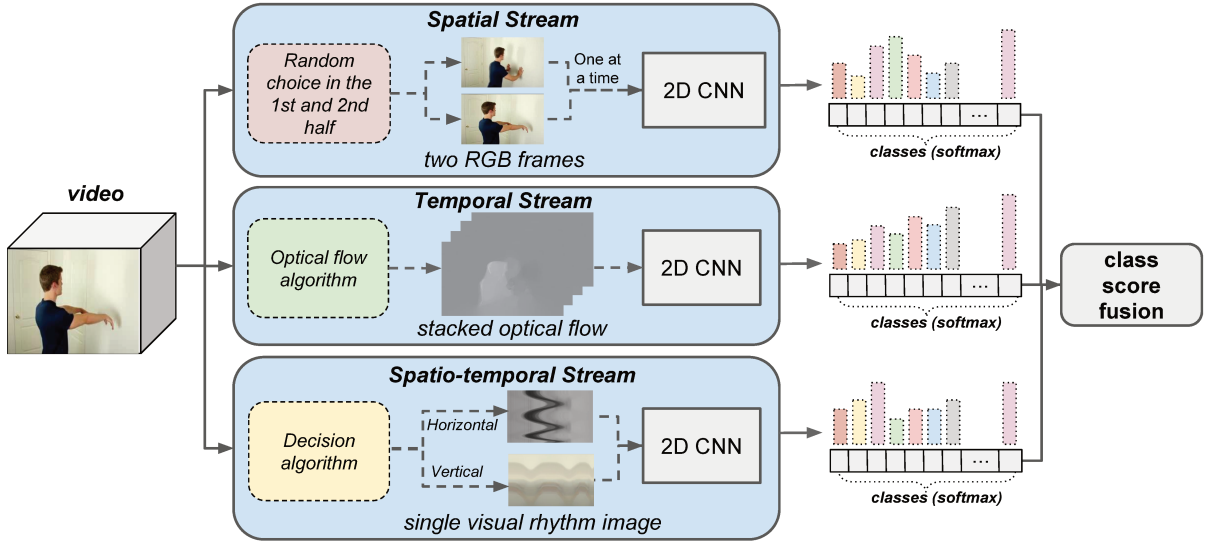


Figure 4.8: Overview of our three-stream proposal for action recognition. Extracted from [17].

Given a video, the output of each stream is a vector containing the softmax score for every class. A weight is assigned to each softmax layer using a grid search in the training/validation set. Then, the class decision is obtained by applying a weighted average strategy in the scores. Further details about the streams are given as follows.

4.3.1 Improved Spatial Stream

Despite the good results achieved by the original spatial stream exploring a single frame, the appearance of the scene may change significantly during the time, either by scene conditions as lighting and occlusions or by the variety of poses, objects and background in the video (Figure 4.9). Therefore, a single appearance may not be sufficient to describe the action, since the elements that characterize it may not be apparent in the frame. As such, we collect two frames per video, using a random choice, to train the network: one in the first half of the video and another in the second half.

Let V be a video with N frames. Indices i and j are random floating point numbers returned in the range $[0, 1)$ by means of the $random()$ function in Equations 4.1 and 4.2, respectively. These indices are used to select the corresponding frames in each half of the video.

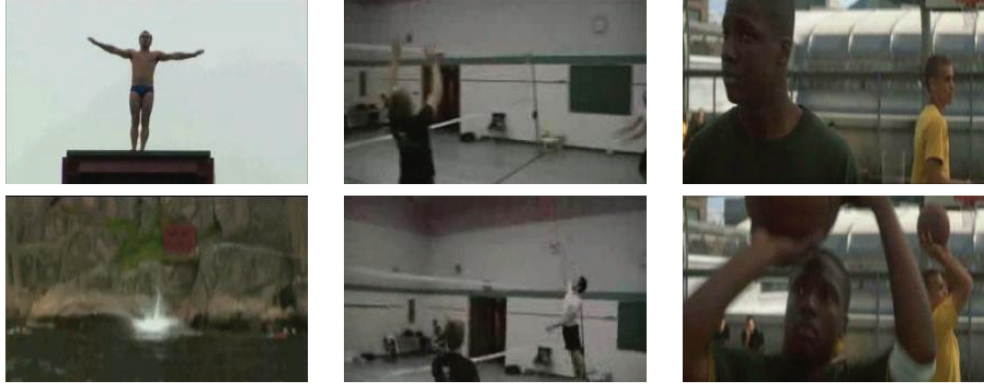


Figure 4.9: Fragments from the UCF101 that show the significant variation in appearance in the two halves of the videos. Each column contains a frame from the first and the second half of the same video, respectively. The background changes in the first video, different actors are present in each frame in the second one and the balls are not in the first frame from the third video. Extracted from [17].

Finally, our spatial stream receives each of both frames at a time (Figure 4.10). As mentioned previously, this approach is used to cover the possible variations produced during the course of the video.

$$i = \lfloor \text{random}() * \lfloor \frac{N}{2} \rfloor + 1 \rfloor \quad (4.1)$$

$$j = \lfloor \frac{N}{2} \rfloor + \lfloor \text{random}() * \lfloor \frac{N}{2} \rfloor + 1 \rfloor \quad (4.2)$$

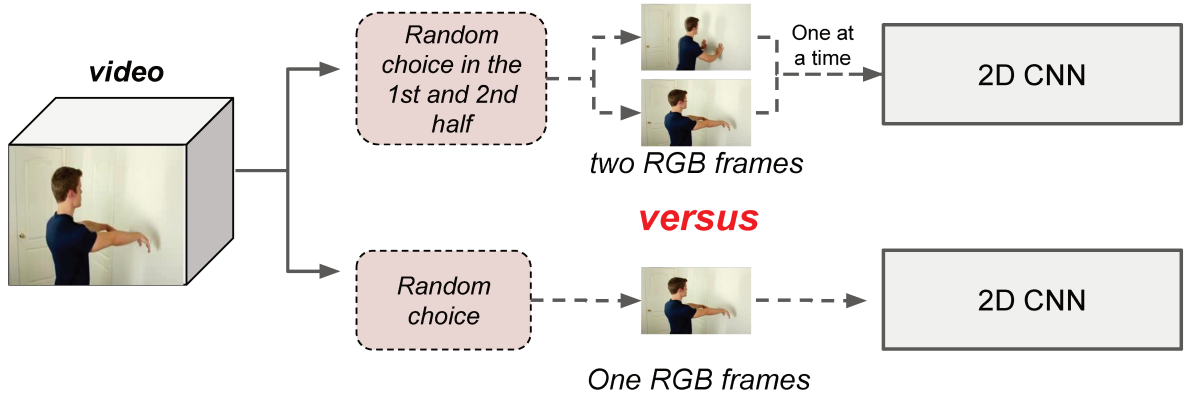


Figure 4.10: Overview of our improved spatial stream versus the spatial stream of the literature.

Testing protocol remains the same in our spatial stream: we use 25 frames evenly sampled from each testing video, and 10 new samples are produced from them, derived from data augmentation. Finally, all the computed outputs are combined through the average of the scores to obtain the stream result.

4.3.2 Temporal Stream

Since the temporal stream achieves great individual results, as reported in the works by Simonyan and Zisserman [67] and Wang et al. [87], it remains the same in our architecture. It consists of a CNN that receives 10 pairs of consecutive optical flow images in the form of a 20-channel image (stack) for training. An overview of the temporal stream is illustrated in Figure 4.11.

Similar to the spatial stream, 25 stacks of optical flow images were used for testing. They were sampled from each video and used to produce 10 new samples per stack by corner cropping (4 corners and 1 central crop) and horizontal flipping techniques. Each sample is individually tested, and all the computed outputs are combined through the average of the scores to obtain the stream result.

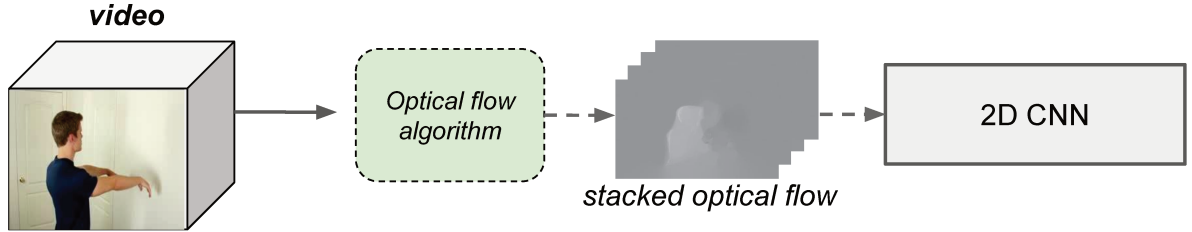


Figure 4.11: Overview of the temporal stream of the literature.

4.3.3 Spatio-Temporal Stream

Our spatio-temporal stream is very similar to the spatial stream, that is, it consists of a CNN that receives a single grayscale image. We consider two main approaches to computing our input: horizontal-mean and vertical-mean slices, following the modification given by Equations 2.1 and 2.2. Nevertheless, the final input is obtained through the decision algorithm proposed in Subsection 4.2.4. An overview of our spatio-temporal stream is illustrated in Figure 4.12.

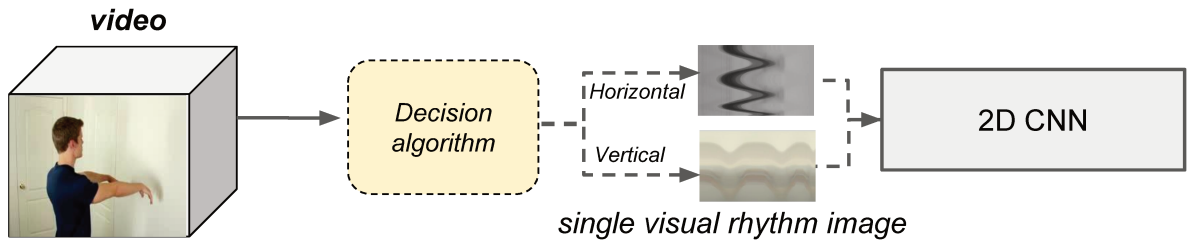


Figure 4.12: Overview of our spatial-temporal stream.

As mentioned previously, this spatial-temporal stream was explored with multiple types of visual rhythm, showing better results our innovative proposal of adaptive visual rhythm. These results are presented and analyzed in Chapter 5.

4.4 Multi-Stream Convolutional and Recurrent Neural Network

As mentioned in the beginning of the chapter, this approach is an extension of the first one, where the main contribution or core of the work remains the adaptive visual rhythm. In addition, RNN is integrated into the first approach in order to take advantage of its ability to preserve sequential information in its hidden states.

An overview of our three-stream network is shown in Figure 4.13. Similar to our previous approach, it contains three deep CNNs working with different modalities: RGB (spatial), optical flow (temporal) and AVR (spatio-temporal) and each CNN is pre-trained in ImageNet dataset and independently fine-tuned with its corresponding modality. However, the trained spatial and temporal CNNs are frozen and used as feature extractors (without the fully connected layer) to feed an LSTM network. This combination of spatial, temporal and LSTM networks represents the first stream of our architecture and is referred to as LSTM stream. Further details about this strategy are given in the following subsections.

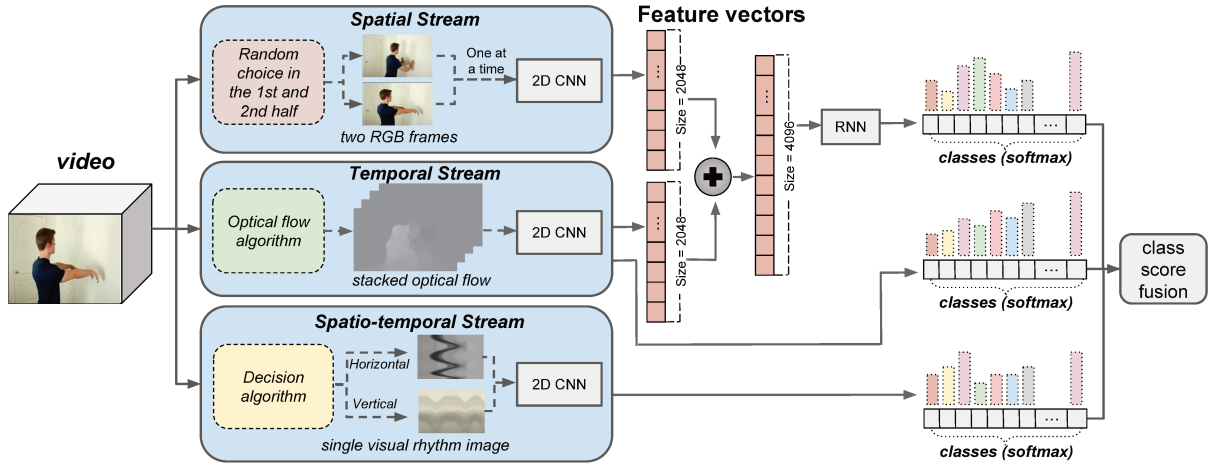


Figure 4.13: Overview of our CNN-RNN multi-stream proposal for action recognition.

4.4.1 LSTM Stream (RGB + Optical Flow)

The LSTM stream is composed of three different networks: two parallel CNNs for spatial and temporal information, and one LSTM. Each network is trained separately. The main premise of the spatial CNN is that the appearance of the scene may help to distinguish the classes. A green grass field, for instance, may be a clue for actions related to soccer games; a horse may help to recognize the horse riding action. Therefore, in the training step, two RGB frames are extracted per video: one in the first half of the video and another in the second half, both randomly chosen. The CNN receives one of those frames at a time. However, by presenting two samples taken at different positions of the video, we are able to capture variations in appearance such as different background that may be characteristics of certain actions.

After the training steps, these networks are frozen and used to generate feature vectors. For the spatial CNN 25 frames per video are used, which are selected every A frames of N in total ($\{F_1, F_A, F_{2 \times A}, \dots, F_{25 \times A}\}$), where $A = \frac{N}{25}$, and F_i represents the i th frame. The same approach is used to select the first 25 frames of each stack for the temporal stream, in other words, from each frame F_i , this one with the next 9 frames are stacked. We decided to take this number of frames per video in order to have a uniform distribution and thus not overlook frames that may have relevant information.

In both CNNs, the fully connected layer is not considered and thus the size of the outputs is 2048. Each pair of vectors generated for a RGB and corresponding stack are concatenated and used to train the LSTM.

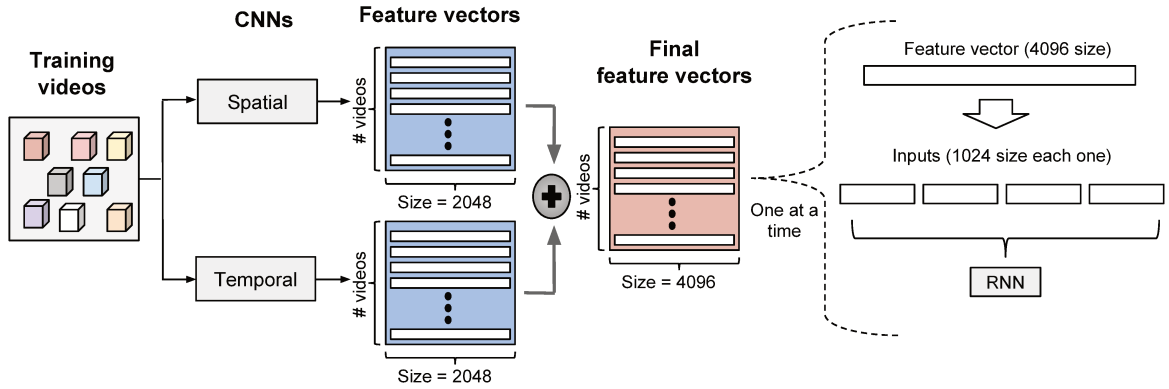


Figure 4.14: Overview of our innovative LSTM stream.

For network training, multiple configurations of parameters were explored, such as: sequence length, input size, hidden size, number of layers, dropout and number of epochs. The best configuration was: 4, 1024, 124, 1, without dropout and 200 epochs, respectively, that is, each feature vector of size 4096 was divided into 4 chunks (sequence length) of sizes 1024 each (input size), which are sequentially passed to the network. Then, $N \times 25$ is the total number of feature vectors that are independently passed as input data, where N is the number of videos for the training stage (Figure 4.14).

The test stage follows the same sequence of steps as the training and are also considered 25 samples per video. However, the final result is the average of their softmax vectors, thus obtaining only one of them per video.

4.4.2 Weighted Average Fusion

In the testing stage, the three weights for softmax fusion were defined through a grid search strategy. For each weight, we tested every value from 0.5 to 10 with a 0.5 step. The best combination for the first and second approach is 3 for temporal, 2 for spatial/LSTM and 1 for spatio-temporal stream (Figure 4.15).

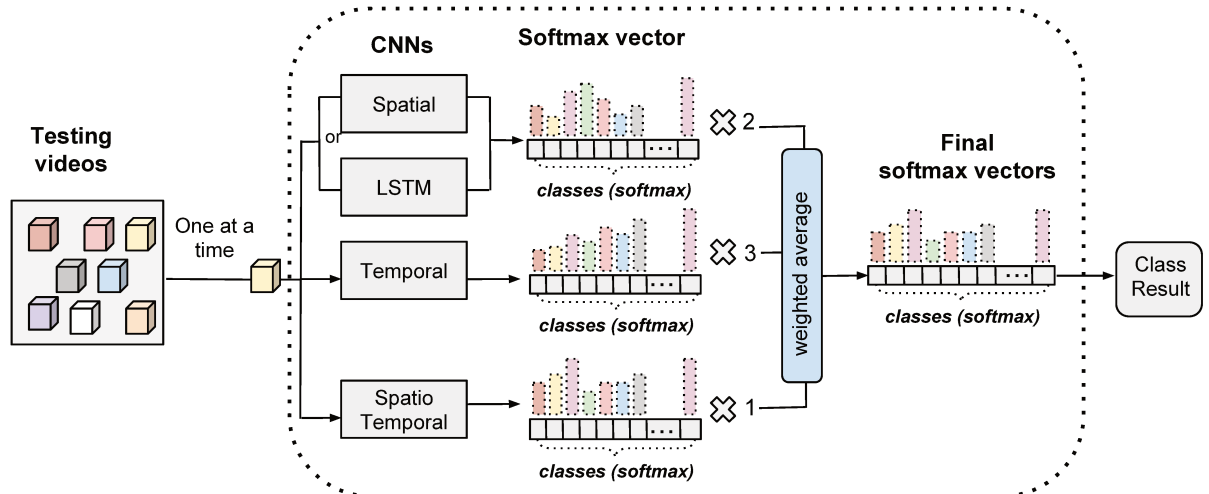


Figure 4.15: Overview of our weighted average fusion technique.

Chapter 5

Experiments

Results obtained with our experiments are described in this chapter. Two datasets are used to evaluate the effectiveness and precision of our two proposed architectures. Both were chosen due to their complexity and amount of data, as described in the previous chapter (Table 3.1 shows a summary of the most commonly used datasets most used in the human action recognition problem in video sequences). This chapter is organized into three sections. The first describes experiments of several visual rhythm representations and compares their results with our proposal (AVR). The second one presents the experiments and results obtained with our two proposed architectures. Finally, we compare our results to those of the state of the art.

In this work, as mentioned in the previous chapter, we adopt the ResNet152 [30]/Inception V3 [75] as CNN architectures for the three streams. We train the spatial and temporal stream using the same strategy provided by Wang et al. [87] and we follow the temporal stream strategy for the spatio-temporal one.

The first three sections are divided as follows: tables that show individual results (for each split and final results) with both networks mentioned before, comparison of individual results and, finally, a bar graph that shows the accuracy rates obtained for each action class present in the videos.

5.1 Visual Rhythms

In the first experiment, we compare five different approaches to the spatio-temporal stream separated from the other streams. The results are shown in Tables 5.1, 5.2 and 5.3. The first three approaches consider, respectively, one horizontal-mean, one vertical-mean and diagonal image as input; in the fourth one, the input is a stack of both images, that is, a 2-channel input; the fifth approach consists of our adaptive method, so the input image is taken according to the direction of the video calculated in Algorithm 1, described in the previous chapter. Figures 5.1 and 5.2 show visual rhythm and optical flow images extracted from the UCF101 dataset.

Tables 5.1 and 5.2 show the individual results in more details, where the results are presented for each split. It is possible to observe that split 2 is slightly higher than the others in most cases and our AVR approach has the best results in all splits.

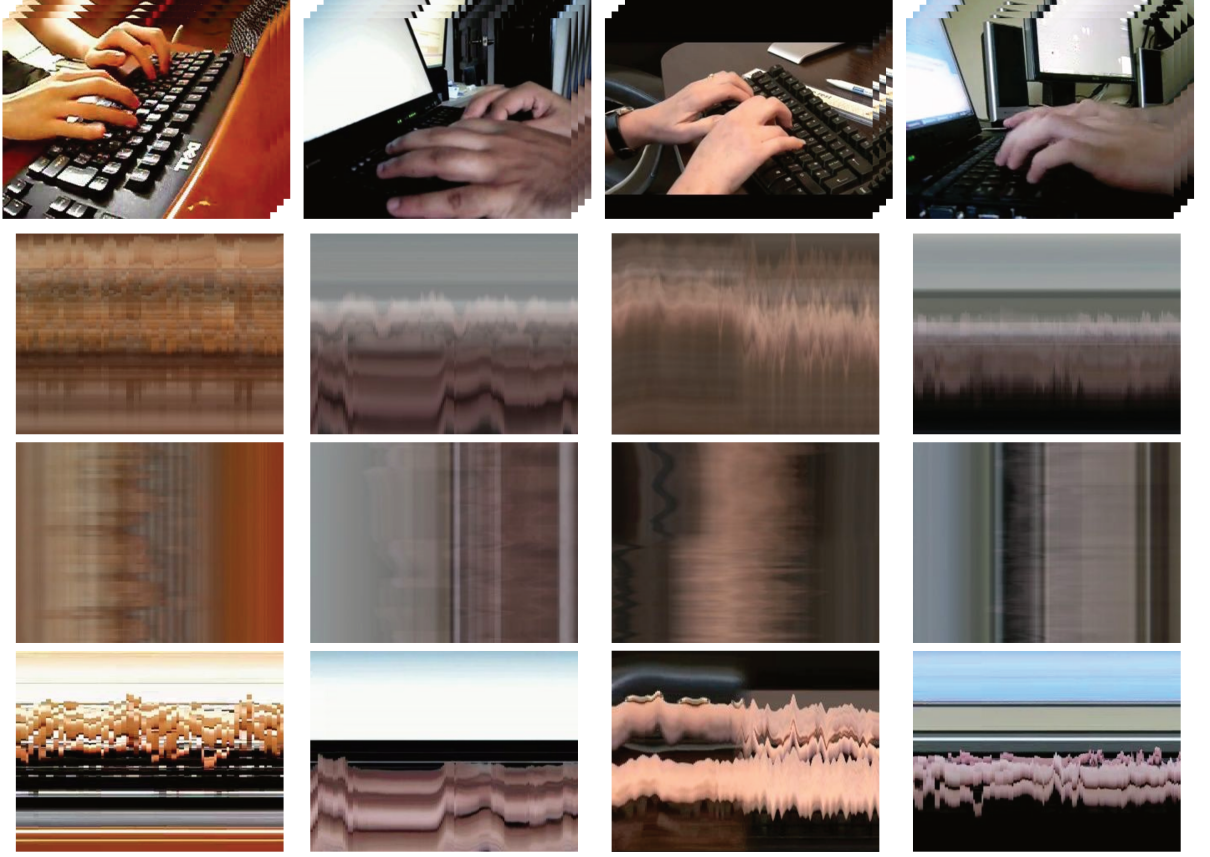


Figure 5.1: First row, from top to bottom, shows some videos of the UCF101 dataset that belong to the **Typing** class. The second, third and fourth row exhibit visual rhythm images obtained from the horizontal-mean, vertical-mean and diagonal strategies.

Table 5.1: Individual results for ResNet152.

| Approach (Visual Rhythm) | UCF101 | | | | HMDB51 | | | |
|-----------------------------|---------|---------|---------|--------------|---------|---------|---------|--------------|
| | Split 1 | Split 2 | Split 3 | Average | Split 1 | Split 2 | Split 3 | Average |
| Horizontal-mean | 62.12 | 62.51 | 61.04 | 61.89 | 35.95 | 37.32 | 33.14 | 35.47 |
| Vertical-mean | 53.40 | 54.87 | 53.35 | 53.87 | 29.92 | 30.41 | 30.03 | 30.12 |
| Diagonal | 59.08 | 59.78 | 56.42 | 58.43 | 33.38 | 34.12 | 30.76 | 32.75 |
| Stacked previous approaches | 50.65 | 50.83 | 48.22 | 49.90 | 28.76 | 30.04 | 29.82 | 29.54 |
| Adaptive Visual Rhythm | 64.13 | 64.38 | 63.23 | 63.91 | 38.56 | 39.80 | 39.48 | 39.28 |

Table 5.3 reports the final results for each approach, where clearly our AVR approach presents superior results compared to the others. It is worth mentioning that the combination of both directions at the same time (third approach) achieves the lowest accuracy rates, even compared to the individual ones. This means that, although each video is better represented by a specific direction, the presence of the second one has an adverse effect on the performance. This observation is reinforced by the adaptive results, since it improves the performance using only one chosen direction per video. In addition, Table 5.3 shows that, for this stream, Inception V3 works slightly better than ResNet152.



Figure 5.2: Videos of the UCF101 dataset with their respective optical flow images below each of them.

Table 5.2: Individual results for Inception V3.

| Approach (Visual Rhythm) | UCF101 | | | | HMDB51 | | | |
|-----------------------------|---------|---------|---------|--------------|---------|---------|---------|--------------|
| | Split 1 | Split 2 | Split 3 | Average | Split 1 | Split 2 | Split 3 | Average |
| Horizontal-mean | 61.93 | 62.02 | 63.15 | 62.37 | 36.15 | 34.92 | 35.65 | 35.57 |
| Vertical-mean | 55.18 | 57.17 | 53.12 | 55.16 | 29.38 | 31.24 | 30.18 | 30.27 |
| Diagonal | 59.31 | 56.28 | 58.74 | 58.11 | 33.18 | 32.31 | 33.92 | 33.14 |
| Stacked previous approaches | 50.60 | 48.15 | 47.22 | 48.65 | 29.12 | 29.88 | 30.21 | 29.74 |
| Adaptive Visual Rhythm | 65.24 | 63.85 | 65.12 | 64.74 | 39.35 | 39.08 | 40.46 | 39.63 |

Figures 5.3 and 5.4 show bar graphs that help us better understand the results obtained for each class. From Figure 5.3, corresponding to the results for the HMDB51 dataset, it is observed that no class is predicted with accuracy of 100%, however, only 3 of them are below 10%. Another fact to notice is that the results are not well distributed, that is, some classes have very high or very low accuracy rates. Similarly, Figure 5.4 shows that no class is predicted with accuracy of 100% for the UCF101 dataset, however, only 1 is below 20% and most of them are above 40%. The results are not well distributed either.

Figure 5.1 shows some examples of visual rhythm images for the *Typing* class of the UCF101 dataset. In this set of images, we can notice an interesting behavior: different

Table 5.3: Results and comparison of different approaches used to create the visual rhythms.

| Approach (Visual Rhythm) | ResNet152 | | Inception V3 | |
|-----------------------------|-----------|--------|--------------|--------------|
| | UCF101 | HMDB51 | UCF101 | HMDB51 |
| Horizontal - mean | 61.89 | 35.47 | 62.37 | 35.57 |
| Vertical - mean | 53.87 | 30.12 | 55.16 | 30.27 |
| Diagonal | 58.43 | 32.75 | 58.11 | 33.14 |
| Stacked previous approaches | 49.90 | 29.54 | 48.65 | 29.74 |
| Adaptive Visual Rhythm | 63.91 | 39.28 | 64.74 | 39.63 |

approaches show similar patterns for videos of the same class, which is a great advantage of training our convolutional neural network since, thanks to it, our network can associate these patterns in common for their respective classes in a more effective way.

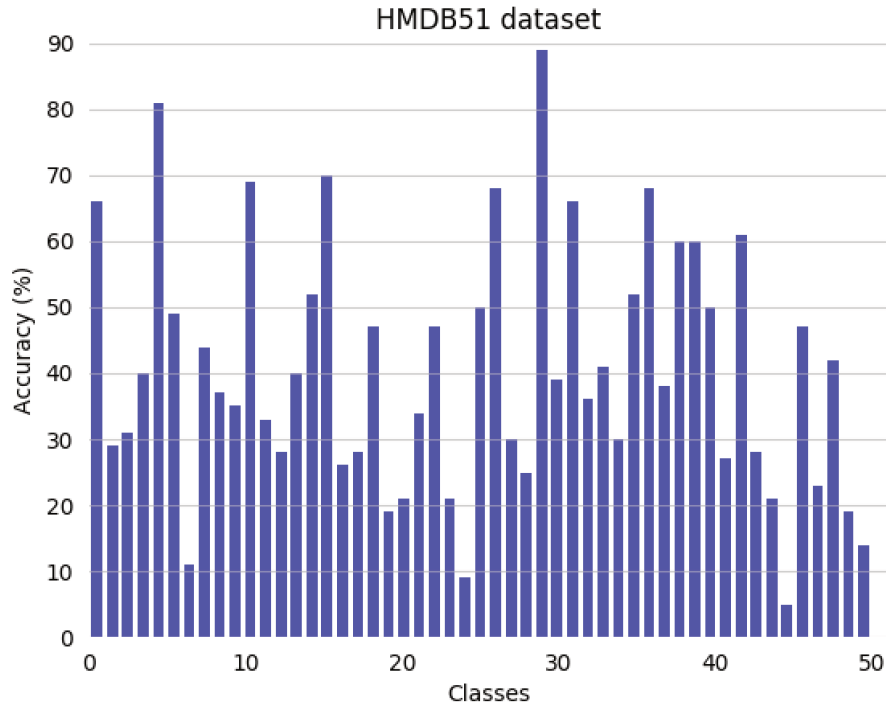


Figure 5.3: Bar graph for the accuracy obtained for each class of the HMDB51 dataset using our adaptive visual rhythm.

5.2 Multi-Stream Architectures

This section describes the experiments and respective results obtained with our two proposed architectures.

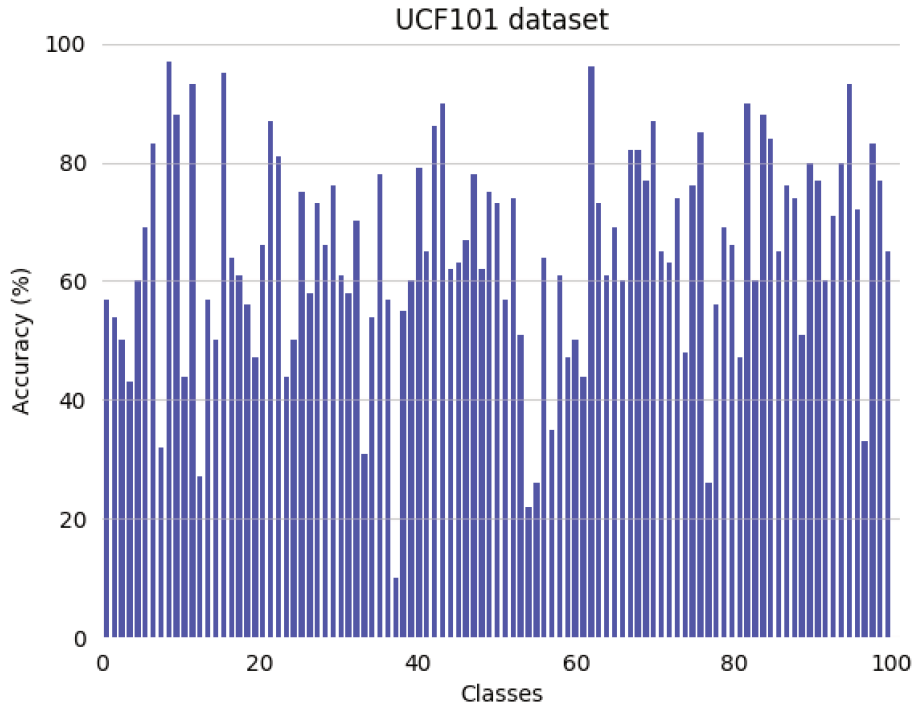


Figure 5.4: Bar graph for the accuracy obtained for each class of the UCF101 dataset using our adaptive visual rhythm.

5.2.1 Approach 1

The results of the second experiment are shown in Tables 5.4, 5.5, and 5.6, which is fundamental to understand the importance of each stream individually. RGB* corresponds to our improved spatial stream and RGB to the original one. For the visual rhythms, we report the best approach (AVR) from Table 5.3.

Concerning the two spatial approaches, the proposed stream (RGB*) outperforms the original (RGB), since it collects more appearances from each video. Similar to the other multi-stream networks [67, 87], the temporal stream achieves the best results among the four strategies. It is followed by RGB*, RGB and AVR, in this order. This justifies the set of weights found with grid search strategy (Subsection 4.4.2). We can also see in this table that Inception V3 presents superior results than ResNet152 in the spatial and spatio-temporal streams, especially for the HMDB51 dataset, however, the opposite occurs in the temporal stream for UCF101 dataset.

Table 5.9 reports the accuracy rates of every combination for the three streams: improved spatial, temporal and spatio-temporal. Note that RGB* + AVR and optical flow + AVR outperform individual RGB* and optical flow. Thus, AVR provides complementary information for the network by encoding long-term dynamics. The worst results were achieved by the combination of two temporal features (optical flow + AVR), suggesting that appearance is very relevant for the recognition task. The combination of the three streams outperforms the others, therefore, all the three features contribute to the recognition process.

Table 5.4: Individual results for ResNet152.

| Single-Stream | UCF101 | | | | HMDB51 | | | |
|---------------|---------|---------|---------|--------------|---------|---------|---------|--------------|
| | Split 1 | Split 2 | Split 3 | Average | Split 1 | Split 2 | Split 3 | Average |
| RGB* images | 85.65 | 85.94 | 86.72 | 86.10 | 46.08 | 47.06 | 44.71 | 45.95 |
| RGB image | 85.57 | 86.15 | 85.93 | 85.88 | 44.27 | 43.37 | 43.99 | 43.88 |
| Optical flow | 85.28 | 88.38 | 87.91 | 87.19 | 57.45 | 58.10 | 60.00 | 58.52 |
| AVR | 64.13 | 64.38 | 63.23 | 63.91 | 38.56 | 39.80 | 39.48 | 39.28 |

Table 5.5: Individual results for Inception V3.

| Single-Stream | UCF101 | | | | HMDB51 | | | |
|---------------|---------|---------|---------|--------------|---------|---------|---------|--------------|
| | Split 1 | Split 2 | Split 3 | Average | Split 1 | Split 2 | Split 3 | Average |
| RGB* images | 86.73 | 86.50 | 86.61 | 86.61 | 54.58 | 51.37 | 49.35 | 51.77 |
| RGB image | 85.83 | 86.35 | 86.08 | 86.09 | 52.71 | 51.12 | 48.45 | 50.76 |
| Optical flow | 86.04 | 87.44 | 87.36 | 86.95 | 59.67 | 60.52 | 59.54 | 59.91 |
| AVR | 65.24 | 63.85 | 65.12 | 64.74 | 39.35 | 39.08 | 40.46 | 39.63 |

Table 5.6: Results and comparison of the individual results (streams).

| Single-Stream | ResNet152 | | Inception V3 | |
|---------------|--------------|--------|--------------|--------------|
| | UCF101 | HMDB51 | UCF101 | HMDB51 |
| RGB* images | 86.10 | 45.95 | 86.61 | 51.77 |
| RGB image | 85.88 | 43.88 | 86.09 | 50.76 |
| Optical flow | 87.19 | 58.52 | 86.95 | 59.91 |
| AVR | 63.91 | 39.28 | 64.74 | 39.63 |

Table 5.7: Results for RGB*, optical flow and adaptive visual rhythm stream fusion for ResNet152.

| Multi-Stream | UCF101 | | | | HMDB51 | | | |
|---------------------------|--------|--------|--------|--------------|--------|--------|--------|--------------|
| | split1 | split2 | split3 | Average | split1 | split2 | split3 | Average |
| RGB* + AVR | 90.64 | 90.26 | 90.54 | 90.48 | 59.74 | 57.31 | 58.30 | 58.45 |
| RGB* + optical flow | 93.29 | 93.60 | 93.31 | 93.40 | 64.88 | 63.08 | 65.12 | 64.36 |
| optical flow + AVR | 87.92 | 87.40 | 87.90 | 87.74 | 64.06 | 64.41 | 64.13 | 64.20 |
| RGB* + AVR + optical flow | 93.79 | 94.81 | 94.30 | 94.30 | 67.58 | 68.43 | 68.95 | 68.32 |

Figure 5.5 and 5.6 show the bar graphs for the HMDB51 and UCF101 datasets, respectively. In the first we can note that all the classes are above 20% and the most of them above 60%. In the second one, a important detail to observe is that almost half reach 100% of accuracy and all are above 70%. Both bar graphs show that the result are not well distributed.

Table 5.8: Results for RGB*, optical flow and adaptive visual rhythm stream fusion for Inception V3.

| Multi-Stream | UCF101 | | | | HMDB51 | | | |
|---------------------------|--------|--------|--------|--------------|--------|--------|--------|--------------|
| | split1 | split2 | split3 | Average | split1 | split2 | split3 | Average |
| RGB* + AVR | 90.91 | 90.79 | 90.52 | 90.74 | 63.86 | 59.87 | 60.20 | 61.31 |
| RGB* + optical flow | 93.16 | 92.60 | 93.06 | 92.94 | 68.10 | 65.31 | 65.88 | 66.43 |
| Optical flow + AVR | 89.12 | 89.62 | 88.95 | 89.23 | 65.62 | 65.49 | 65.24 | 65.45 |
| RGB* + AVR + optical flow | 93.87 | 93.55 | 93.80 | 93.74 | 70.96 | 70.00 | 68.97 | 69.98 |

Table 5.9: Results for RGB*, optical flow and adaptive visual rhythm stream fusion.

| Multi-Stream | ResNet152 | | Inception V3 | |
|---------------------------------|--------------|--------|--------------|--------------|
| | UCF101 | HMDB51 | UCF101 | HMDB51 |
| RGB* images + AVR | 90.48 | 58.45 | 90.74 | 61.31 |
| RGB* images + optical flow | 93.40 | 64.36 | 92.94 | 66.43 |
| Optical flow + AVR | 89.74 | 64.20 | 89.23 | 65.45 |
| RGB* image + AVR + optical flow | 94.30 | 68.32 | 93.74 | 69.98 |

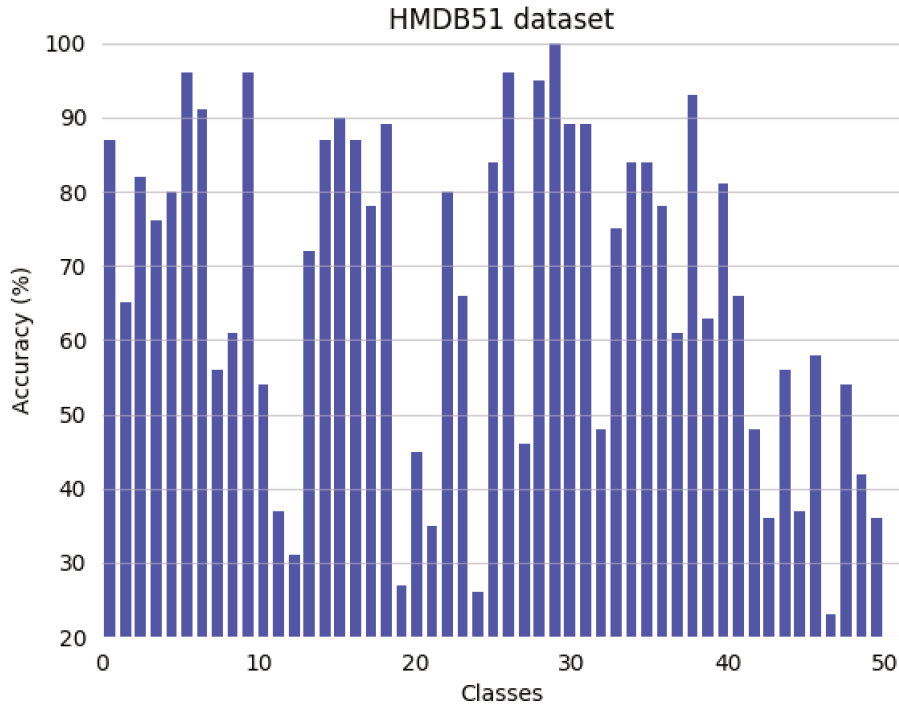


Figure 5.5: Bar graph for the accuracy obtained for each class of the HMDB51 dataset using our three-stream approach 1.

5.2.2 Approach 2

Tables 5.10, 5.11 and 5.12 show the results for our third experiment conducted on the UCF101 and HMDB51 datasets using ResNet152 and Inception V3. The results obtained

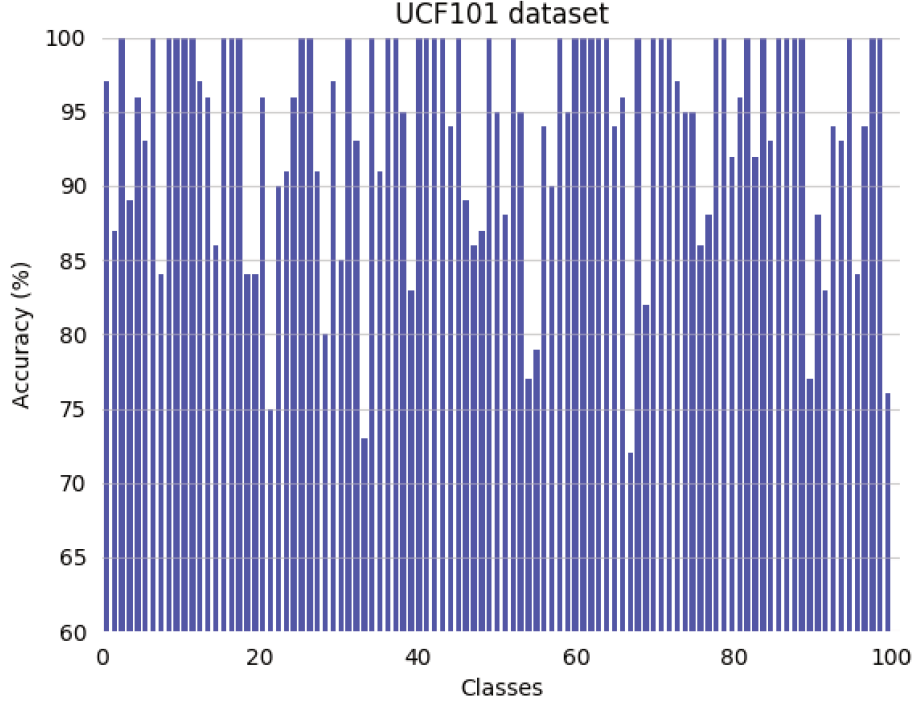


Figure 5.6: Bar graph for the accuracy obtained for each class of the UCF101 dataset using our three-stream approach 1.

individually for the spatio-temporal (AVR) and temporal stream are not exhibited since they are the same of the previous experiments (Subsection 5.2.1). The first rows of each table correspond to the LSTM stream. The following four rows show the results of all possible combinations using the previous three stream.

Table 5.10: Results for stream combination using the ResNet152.

| Modality | UCF101 | | | | HMDB51 | | | |
|---------------------------|--------|--------|--------|--------------|--------|--------|--------|--------------|
| | split1 | split2 | split3 | Average | split1 | split2 | split3 | Average |
| LSTM | 88.18 | 91.54 | 91.80 | 90.50 | 63.27 | 63.79 | 62.55 | 63.20 |
| LSTM + optical flow | 91.09 | 92.47 | 93.24 | 92.27 | 65.36 | 65.36 | 65.95 | 65.56 |
| LSTM + AVR | 91.67 | 92.42 | 93.26 | 93.45 | 67.39 | 66.60 | 68.63 | 67.54 |
| Optical flow + AVR | 87.79 | 91.48 | 90.07 | 89.78 | 62.88 | 64.71 | 65.23 | 64.27 |
| LSTM + optical flow + AVR | 92.76 | 94.19 | 93.99 | 93.65 | 69.61 | 70.13 | 69.93 | 69.89 |

In both experiments, LSTM + optical flow and LSTM + AVR presented similar results. However, since with both networks the worst combination was the optical flow + AVR, we can infer that the context information provided by the RGB frames are relevant to recognize the action instead of using only temporal modalities.

Although our LSTM stream obtains significantly better results than the other two, its weight in the fusion process is smaller than the temporal stream, but larger than the spatio-temporal (Figure 4.15). However, the combination of the three streams allows us to achieve in the UCF101 dataset a considerable improvement of 3% in the best individual

Table 5.11: Results for stream combination using the Inception V3.

| Modality | UCF101 | | | | HMDB51 | | | |
|---------------------------|--------|--------|--------|--------------|--------|--------|--------|--------------|
| | split1 | split2 | split3 | Average | split1 | split2 | split3 | Average |
| LSTM | 90.53 | 91.76 | 92.01 | 91.43 | 63.82 | 64.33 | 64.20 | 64.12 |
| LSTM + optical flow | 92.20 | 93.11 | 93.18 | 92.86 | 65.82 | 64.91 | 66.15 | 65.63 |
| LSTM + AVR | 91.91 | 92.78 | 93.43 | 92.71 | 67.52 | 67.42 | 68.93 | 67.96 |
| Optical flow + AVR | 89.12 | 90.64 | 90.31 | 90.02 | 63.11 | 64.93 | 65.18 | 64.41 |
| LSTM + optical flow + AVR | 93.75 | 94.98 | 94.78 | 94.50 | 69.82 | 70.45 | 70.01 | 70.09 |

result (LSTM) and almost 2% in the best pair (LSTM + AVR and LSTM + optical flow) and 6% and 3%, respectively, in the HMDB51 dataset.

Table 5.12: Results for RGB*, optical flow and adaptive visual rhythm stream fusion.

| Modality | ResNet152 | | Inception V3 | |
|---------------------------|-----------|--------|--------------|--------------|
| | UCF101 | HMDB51 | UCF101 | HMDB51 |
| LSTM + optical flow | 92.27 | 65.56 | 92.86 | 65.63 |
| LSTM + AVR | 93.45 | 67.54 | 92.71 | 67.96 |
| Optical flow + AVR | 89.78 | 64.27 | 90.02 | 64.41 |
| LSTM + optical flow + AVR | 93.65 | 69.89 | 94.50 | 70.09 |

An important detail to notice here is that the LSTM + AVR combination obtains superior results than any other pair in most of the experiments (three out of the four results). This may be due to the fact that our LSTM stream already contains temporal information from optical flow images.

Unlike the first approach (Subsection 5.2.1), the bar graph for the HMDB51 dataset 5.7 shows that more than one class reach the accuracy of 100%, however, similarly to it, all the classes are above 20%. On the other hand, the bar graph for the UCF101 dataset (Figure 5.8) shows the same behavior as the previous approach, reaching almost half of the classes accuracy of 100% and all above 70%.

5.3 State-of-the-Art Comparison

After reporting and analyzing the results obtained in both datasets, we compare our accuracy rates to state-of-the-art approaches in Table 5.13. An important detail must be mentioned before comparing our results: the methods use different pre-training strategies. Some of them are not based on deep networks, so they do not have a pre-training step. Concerning the remaining methods, one group is pre-trained on the ImageNet dataset, whereas the others use the ImageNet and Kinetics. The approaches pre-trained with both have an advantage over those trained only on ImageNet, since the Kinetics is one of the largest and most varied dataset for the action recognition problem, generating more effective networks. However, it is difficult to use due to the intensive computing power required.

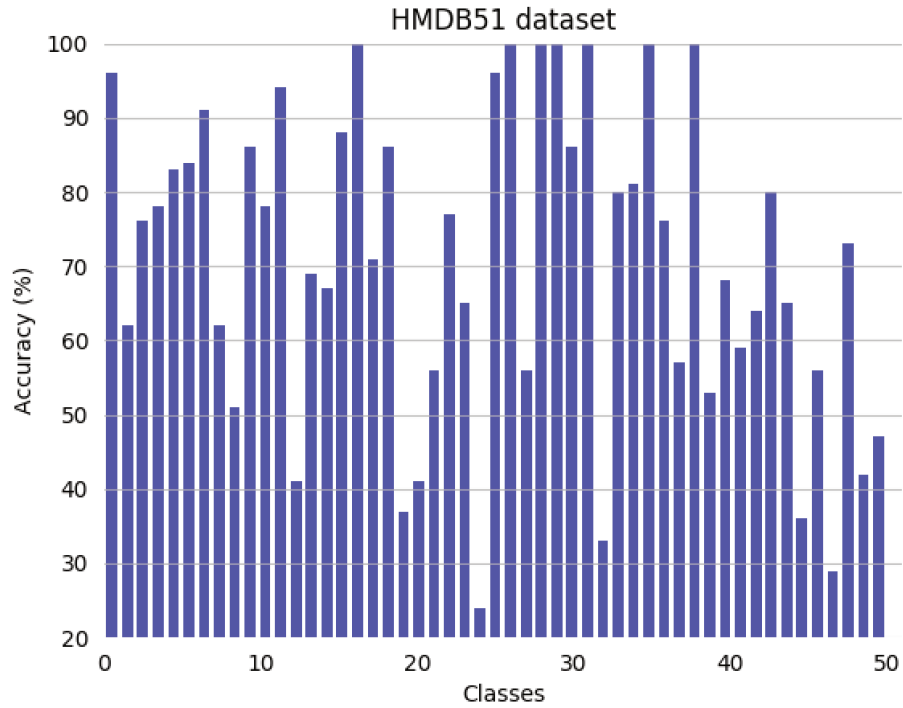


Figure 5.7: Bar graph for the accuracy obtained for each class of the HMDB51 dataset using our three-stream approach 2.

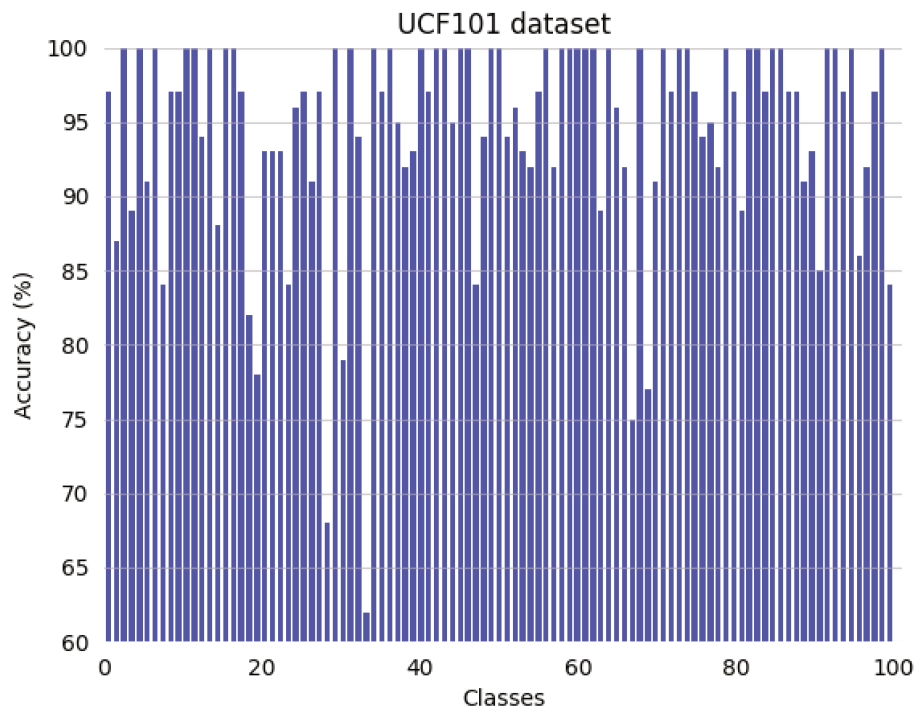


Figure 5.8: Bar graph for the accuracy obtained for each class of the UCF101 dataset using our three-stream approach 2.

Table 5.13 shows that our second approach is slightly better than the first one, achieving 94.5% and 70.1% in the UCF101 and HMDB51 datasets, respectively, both trained over the Inception V3 network and pre-trained with the ImageNet. However, the most recent approaches pre-trained with ImageNet + Kinetics reach approximately 97-98% on UCF101 and 78-82% on HMDB51, outperforming our best results in up to 3.7% and 12% in each dataset, respectively. Nonetheless, the best results of the approaches only trained with ImageNet are 94.6% on UCF101 and 70.4% on HMDB51, placing us as the second best result in this group. Therefore, our method achieves competitive results against other methods available in the literature.

Table 5.13: Comparison of accuracy rates (%) for UCF101 and HMDB51 datasets. Cells on bold represents the overall highest accuracies, whereas underlined cells consist of the best results using only ImageNet to pre-train the network.

| Method | Pre-training Dataset | UCF101 | HMDB51 |
|-----------------------------|----------------------|-------------|-------------|
| iDT + HSV [55] | — | 87.9 | 61.1 |
| Two-Stream [67] | ImageNet | 88.0 | 59.4 |
| Two-Stream + LSTM [50] | ImageNet | 88.6 | — |
| Two-Stream TSN [88] | ImageNet | 94.0 | 68.5 |
| Three-Stream TSN [88] | ImageNet | 94.2 | 69.4 |
| Three-Stream [84] | ImageNet | 94.1 | <u>70.4</u> |
| TDD + iDT [86] | ImageNet | 91.5 | 65.9 |
| LTC + iDT [82] | — | 92.7 | 67.2 |
| KVMDF [100] | ImageNet | 93.1 | 63.3 |
| STP [89] | ImageNet | <u>94.6</u> | 68.9 |
| L ² STM [72] | ImageNet | 93.6 | 66.2 |
| Two-Stream I3D [12] | ImageNet+Kinetics | 98.0 | 80.9 |
| I3D+PoTion [16] | ImageNet+Kinetics | 98.2 | 80.9 |
| DTPP [99] | ImageNet+Kinetics | 98.0 | 82.1 |
| SVMP + I3D [85] | ImageNet+Kinetics | — | 81.3 |
| R(2+1)D-TwoStream [79] | Kinetics | 97.3 | 78.7 |
| Our method 1 (ResNet152) | ImageNet | 94.3 | 68.3 |
| Our method 1 (Inception V3) | ImageNet | 93.7 | 69.9 |
| Our method 2 (ResNet152) | ImageNet | 93.6 | 69.9 |
| Our method 2 (Inception V3) | ImageNet | 94.5 | 70.1 |

Chapter 6

Conclusions and Future Work

In this work, we addressed an investigation based on multi-stream convolutional neural network approaches applied to the action recognition problem. We proposed two multi-stream architectures. The first is composed of an improved spatial stream, a temporal stream and a spatio-temporal stream, whereas the second one is composed of the second and third streams previously mentioned and an LSTM stream, which is the result of the fusion of the two first streams of our previous approach.

Our improved spatial stream uses twice as many samples as proposed in the original to capture variations in the appearance information, achieving superior results. The input of the spatio-temporal stream is defined through a new decision algorithm based on point tracking that estimates the predominant direction in each video. We referred to this method as Adaptive Visual Rhythm (AVR). Our experiments showed that the AVR approach outperformed fixed-direction approaches and provided complementary information for the network, so that such technique was the core of the two proposed approaches. We also demonstrated that our methods achieved fairly competitive results compared to state-of-the-art approaches on two challenging datasets.

The visual rhythm method provided spatio-temporal information that is very useful to create representative patterns that are shown through an image. This type of information can be used as a data source to train deep learning architectures. This is an additional stream in the well known two-stream CNN, providing complementary information that helped us improve previous results and reaching competitive results compared to approaches available in the literature. Moreover, the confusion matrices (shown as bar graphs) provided us a better comprehension of the individual results for each action class, offering new opportunities for addressing the weaknesses of our architecture in future approaches.

The visual rhythm proved to be a powerful representation to reduce a video into an image, creating a compact and rich source of spatio-temporal information. However, there are several strategies for the visual rhythm image construction, each one used to train a VR stream of our multi-stream architecture. The AVR approach was the best way to train the spatio-temporal stream, achieving results clearly superior to the other methods.

The visual rhythm images presented certain patterns that are very similar to videos of the same class, which allowed us to create models capable of achieving a high level of accuracy and also improving the overall results when combined with other models trained

with other types of information (RGB and optical flow). Even though the individual results did not surpass others, each stream contributed enormously to the final result, providing insightful ideas about the relevance of spatio-temporal features for video analysis problems.

Based on these conclusions, we are able to answer the research questions described in Chapter 1:

- Is the visual rhythm representation a useful data source to train a deep learning architecture?

Answer: Although it is the core of our work, individual results after training a deep learning architecture with this type of data did not show a very good performance. This is due to the restriction of the amount of data available, since we obtain only a VR image for each video sequence. A possible solution would be to create more than one VR image using small segments of a video. Longer videos would also aid the VR construction process. Another issue is the complexity of the videos due to their low resolution and unstable background.

- Is the spatio-temporal information extracted from the visual rhythm method useful for the action recognition problem?

Answer: As well as some other methods mentioned in the literature review (Chapter 2), our approach showed that the spatio-temporal information is complementary and an important source of data for video analysis tasks. Spatial or temporal information individually is not sufficient to explore the relations between them, making necessary the use of representations that contain both information in a single type of data, such as the VR images.

- Only one RGB frame per video is sufficient to train a spatial stream?

Answer: Although some previous works of the literature employ a single frame in spatial stream, our work showed that additional frames (two in our case) can be a better choice since the background and actors may vary over time.

- Is the visual rhythm stream more/less discriminative than the optical flow and RGB streams?

Answer: According to our experiments carried out using the weighted average fusion, the visual rhythm stream is the least important, but essential, contributing a few percentage points to the final results. On the other hand, the temporal stream is the most important.

As directions for future work, we intend to use the ImageNet and Kinetics datasets to pre-train each CNN, as well as take advantage of attention modules for LSTM training. Additionally, from Figure 5.8, we noticed that only five classes in the UCF101 dataset achieved results below 80%. Therefore, in a following work, we plan to analyze the content of these classes to explore potential weaknesses of our architecture and thus improve it.

Bibliography

- [1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. YouTube-8M: A Large-Scale Video Classification Benchmark. *arXiv preprint arXiv:1609.08675*, pages 1–11, 2016.
- [2] J. K. Aggarwal and M. S. Ryoo. Human Activity Analysis: A Review. *ACM Computing Surveys*, 43(3):16, 2011.
- [3] J. Almeida, J. A. dos Santos, B. Alberton, L. P. C. Morellato, and R. S. Torres. Phenological Visual Rhythms: Compact Representations for Fine-Grained Plant Species Identification. *Pattern Recognition Letters*, 81:90–100, 2016.
- [4] J. Almeida, N. J. Leite, and R. S. Torres. Rapid Cut Detection on Compressed Video. In *Iberoamerican Congress on Pattern Recognition*, pages 71–78. Springer, 2011.
- [5] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential Deep Learning for Human Action Recognition. In *International Workshop on Human Behavior Understanding*, pages 29–39. Springer, 2011.
- [6] F. Baumann, J. Lao, A. Ehlers, and B. Rosenhahn. Motion Binary Patterns for Action Recognition. In *International Conference on Pattern Recognition Applications and Methods*, pages 385–392, 2014.
- [7] S. S. Beauchemin and J. L. Barron. The Computation of Optical Flow. *ACM Computing Surveys*, 27(3):433–466, 1995.
- [8] H. Bilen, B. Fernando, E. Gavves, and A. Vedaldi. Action Recognition with Dynamic Image Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2799–2813, 2018.
- [9] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as Space-Time Shapes. In *IEEE International Conference on Computer Vision*, pages 1395–1402, Beijing, China, 2005. IEEE.
- [10] A. F. Bobick and J. W. Davis. The Recognition of Human Movement Using Temporal Templates. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 1(3):257–267, 2001.
- [11] J.-Y. Bouguet. Pyramidal Implementation of the Affine Lucas Kanade Feature Tracker Description of the Algorithm. *Intel Corporation*, 5(1-10):4, 2001.

- [12] J. Carreira and A. Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4733. IEEE, 2017.
- [13] M. Chan, D. Estève, C. Escriba, and E. Campo. A Review of Smart Homes: Present State and Future Challenges. *Computer Methods and Programs in Biomedicine*, 91(1):55–81, 2008.
- [14] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the Devil in the Details: Delving Deep into Convolutional Nets. *arXiv preprint arXiv:1405.3531*, pages 1–11, 2014.
- [15] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, 2014.
- [16] V. Choutas, P. Weinzaepfel, J. Revaud, and C. Schmid. PoTion: Pose MoTion Representation for Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7024–7033, 2018.
- [17] D. Concha, H. Maia, H. Pedrini, H. Tacon, A. Brito, H. Chaves, and M. Vieira. Multi-Stream Convolutional Neural Networks for Action Recognition in Video Sequences Based on Adaptive Visual Rhythms. In *17th IEEE International Conference on Machine Learning and Applications*, pages 473–480, Orlando-FL, USA, Dec. 2018.
- [18] N. Dalal, B. Triggs, and C. Schmid. Human Detection Using Oriented Histograms of Flow and Appearance. In *European Conference on Computer Vision*, pages 428–441. Springer, 2006.
- [19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.
- [20] S. Dodge and L. Karam. A Study and Comparison of Human and Deep Learning Recognition Performance under Visual Distortions. In *26th International Conference on Computer Communication and Networks*, pages 1–7. IEEE, 2017.
- [21] P. Druzhkov and V. Kustikova. A Survey of Deep Learning Methods and Software Tools for Image Classification and Object Detection. *Pattern Recognition and Image Analysis*, 26(1):9–15, 2016.
- [22] D. Fleet and Y. Weiss. Optical Flow Estimation. In *Handbook of Mathematical Models in Computer Vision*, pages 237–257. Springer, 2006.
- [23] D. Fortun, P. Bouthemy, and C. Kervrann. Optical Flow Modeling and Computation: A Survey. *Computer Vision and Image Understanding*, 134:1–21, 2015.

- [24] H. Gammulle, S. Denman, S. Sridharan, and C. Fookes. Two Stream LSTM: A Deep Fusion Framework for Human Action Recognition. In *IEEE Winter Conference on Applications of Computer Vision*, pages 177–186. IEEE, 2017.
- [25] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber. Learning Precise Timing with LSTM Recurrent Networks. *Journal of Machine Learning Research*, 3(Aug.):115–143, 2002.
- [26] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep Learning*, volume 1. MIT Press Cambridge, 2016.
- [27] I. Gori, J. K. Aggarwal, L. Matthies, and M. S. Ryoo. Multitype Activity Recognition in Robot-Centric Scenarios. *IEEE Robotics and Automation Letters*, 1(1):593–600, Jan. 2016.
- [28] A. Graves, S. Fernández, and J. Schmidhuber. Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition. In *International Conference on Artificial Neural Networks*, pages 799–804. Springer, 2005.
- [29] K. He, X. Zhang, S. Ren, and J. Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.
- [30] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [31] S. Herath, M. Harandi, and F. Porikli. Going Deeper into Action Recognition: A Survey. *Image and Vision Computing*, 60:4–21, 2017.
- [32] G. W. Humphreys and V. Bruce. *Visual Cognition*. Lawrence Erlbaum Associates London, 1985.
- [33] S. Ji, W. Xu, M. Yang, and K. Yu. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013.
- [34] R. Kahani, A. Talebpour, and A. Mahmoudi-Aznaveh. A Correlation Based Feature Representation for First-Person Activity Recognition. *arXiv preprint arXiv:1711.05523*, pages 1–15, 2017.
- [35] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-Scale Video Classification with Convolutional Neural Networks. In *IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.
- [36] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, and P. Natsev. The Kinetics Human Action Video Dataset. *arXiv preprint arXiv:1705.06950*, pages 1–22, 2017.

- [37] A. Klaser, M. Marszałek, and C. Schmid. A Spatio-Temporal Descriptor Based on 3D-Gradients. In *19th British Machine Vision Conference*, pages 275–1. British Machine Vision Association, 2008.
- [38] H. S. Koppula, R. Gupta, and A. Saxena. Learning Human Activities and Object Affordances from RGB-D Videos. *The International Journal of Robotics Research*, 32(8):951–970, 2013.
- [39] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.
- [40] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A Large Video Database for Human Motion Recognition. In *International Conference on Computer Vision*, pages 2556–2563, 2011.
- [41] H. Kuehne, H. Jhuang, R. Stiefelhagen, and T. Serre. HMDB51: A Large Video Database for Human Motion Recognition. In *High Performance Computing in Science and Engineering*, pages 571–582. Springer, 2013.
- [42] I. Laptev. On Space-Time Interest Points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.
- [43] I. Laptev, B. Caputo, et al. Recognizing Human Actions: A Local SVM Approach. In *International Conference on Pattern Recognition*, pages 32–36. IEEE, 2004.
- [44] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning Realistic Human Actions from Movies. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.
- [45] D. Li, T. Yao, L. Duan, T. Mei, and Y. Rui. Unified Spatio-Temporal Attention Networks for Action Recognition in Videos. *IEEE Transactions on Multimedia*, pages 416–428, 2018.
- [46] J. Liu, J. Luo, and M. Shah. Recognizing Realistic Actions from Videos "In the Wild". In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1996–2003. IEEE, 2009.
- [47] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi. A Survey of Deep Neural Network Architectures and their Applications. *Neurocomputing*, 234:11–26, 2017.
- [48] B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. *DARPA Image Understanding Workshop*, pages 121–130, 1981.
- [49] M. Marszałek, I. Laptev, and C. Schmid. Actions in Context. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2929–2936. IEEE, 2009.

- [50] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond Short Snippets: Deep Networks for Video Classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4694–4702, 2015.
- [51] C.-W. Ngo, T.-C. Pong, and R. T. Chin. Camera Break Detection by Partitioning of 2D Spatio-Temporal Images in MPEG Domain. In *IEEE International Conference on Multimedia Computing and Systems*, volume 1, pages 750–755. IEEE, 1999.
- [52] C.-W. Ngo, T.-C. Pong, and R. T. Chin. Detection of Gradual Transitions through Temporal Slice Analysis. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 36–41. IEEE, 1999.
- [53] C.-W. Ngo, T.-C. Pong, and H.-J. Zhang. Motion Analysis and Segmentation through Spatio-Temporal Slices Processing. *IEEE Transactions on Image Processing*, 12(3):341–355, 2003.
- [54] S. J. Pan and Q. Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.
- [55] X. Peng, L. Wang, X. Wang, and Y. Qiao. Bag of Visual Words and Fusion Methods for Action Recognition: Comprehensive Study and Good Practice. *Computer Vision and Image Understanding*, 150:109–125, 2016.
- [56] E. A. Perez, V. F. Mota, L. M. Maciel, D. Sad, and M. B. Vieira. Combining Gradient Histograms using Orientation Tensors for Human Action Recognition. In *21st International Conference on Pattern Recognition*, pages 3460–3463. IEEE, 2012.
- [57] H.-H. Phan, N.-S. Vu, V.-L. Nguyen, and M. Quoy. Motion of Oriented Magnitudes Patterns for Human Action Recognition. In *International Symposium on Visual Computing*, pages 168–177. Springer, 2016.
- [58] H. Rahmani, A. Mian, and M. Shah. Learning a Deep Model for Human Action Recognition from Novel Viewpoints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):667–681, 2018.
- [59] M. Ravanbakhsh, H. Mousavi, M. Rastegari, V. Murino, and L. S. Davis. Action Recognition with Image based CNN Features. *arXiv preprint arXiv:1512.03980*, pages 1–10, 2015.
- [60] K. K. Reddy and M. Shah. Recognizing 50 Human Action Categories of Web Videos. *Machine Vision and Applications*, 24(5):971–981, 2012.
- [61] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach a Spatio-Temporal Maximum Average Correlation Height Filter for Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

- [62] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.
- [63] M. S. Ryoo and L. Matthies. First-Person Activity Recognition: Feature, Temporal Structure, and Prediction. *International Journal of Computer Vision*, 119(3):307–328, Sept. 2016.
- [64] C. Schuldt, I. Laptev, and B. Caputo. Recognizing Human Actions: A Local SVM Approach. In *17th International Conference on Pattern Recognition*, volume 3, pages 32–36. IEEE, 2004.
- [65] J. Shi and C. Tomasi. Good Features to Track. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600. IEEE, 1994.
- [66] A. Silva Pinto, H. Pedrini, W. Schwartz, and A. Rocha. Video-based Face Spoofing Detection through Visual Rhythm Analysis. In *25th SIBGRAPI Conference on Graphics, Patterns and Images*, pages 221–228. IEEE, 2012.
- [67] K. Simonyan and A. Zisserman. Two-Stream Convolutional Networks for Action Recognition in Videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.
- [68] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, pages 1–14, 2014.
- [69] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu. An End-To-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data. In *Thirty-First AAAI Conference on Artificial Intelligence*, pages 4263–4270, 2017.
- [70] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A Dataset of 101 Human Actions Classes from Videos in the Wild. *arXiv preprint arXiv:1212.0402*, pages 1–7, 2012.
- [71] M. R. Souza. Digital Video Stabilization: Algorithms and Evaluation. Master’s thesis, Institute of Computing, University of Campinas, Campinas, Brazil, 2018.
- [72] L. Sun, K. Jia, K. Chen, D. Y. Yeung, B. E. Shi, and S. Savarese. Lattice Long Short-Term Memory for Human Action Recognition. *IEEE International Conference on Computer Vision*, pages 2147–2156, 2017.
- [73] S. Sun, Z. Kuang, L. Sheng, W. Ouyang, and W. Zhang. Optical Flow Guided Feature: A Fast and Robust Motion Representation for Video Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1390–1399, 2018.
- [74] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper With Convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, June 2015.

- [75] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the Inception Architecture for Computer Vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.
- [76] Y. Tian, L. Cao, Z. Liu, and Z. Zhang. Hierarchical Filtered Motion for Action Recognition in Crowded Videos. *IEEE Transactions on Systems, Man, and Cybernetics*, 42(3):313–323, 2012.
- [77] B. S. Torres and H. Pedrini. Detection of Complex Video Events through Visual Rhythm. *The Visual Computer*, pages 1–21, 2016.
- [78] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning Spatiotemporal Features with 3D Convolutional Networks. In *IEEE international conference on computer vision*, pages 4489–4497, 2015.
- [79] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.
- [80] Z. Tu, W. Xie, J. Dauwels, B. Li, and J. Yuan. Semantic Cues Enhanced Multimodality Multi-Stream CNN for Action Recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2018.
- [81] F. B. Valio, H. Pedrini, and N. J. Leite. Fast Rotation-Invariant Video Caption Detection based on Visual Rhythm. In *Iberoamerican Congress on Pattern Recognition*, pages 157–164. Springer, 2011.
- [82] G. Varol, I. Laptev, and C. Schmid. Long-Term Temporal Convolutions for Action Recognition. *arXiv preprint arXiv:1604.04494*, pages 1510–1517, 2016.
- [83] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action Recognition by Dense Trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3169–3176. IEEE, 2011.
- [84] H. Wang, Y. Yang, E. Yang, and C. Deng. Exploring Hybrid Spatio-Temporal Convolutional Networks for Human Action Recognition. *Multimedia Tools and Applications*, 76(13):15065–15081, 2017.
- [85] J. Wang, A. Cherian, F. Porikli, and S. Gould. Video Representation Learning Using Discriminative Pooling. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1149–1158, 2018.
- [86] L. Wang, Y. Qiao, and X. Tang. Action Recognition with Trajectory-Pooled Deep-Convolutional Descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4305–4314, 2015.
- [87] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao. Towards Good Practices for very Deep Two-Stream Convnets. *arXiv preprint arXiv:1507.02159*, pages 1–5, 2015.

- [88] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In *European Conference on Computer Vision*, pages 20–36. Springer, 2016.
- [89] Y. Wang, M. Long, J. Wang, and P. S. Yu. Spatiotemporal Pyramid Network for Video Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2097–2106. IEEE, 2017.
- [90] D. Weinland, R. Ronfard, and E. Boyer. Free Viewpoint Action Recognition using Motion History Volumes. *Computer Vision and Image Understanding*, 104(2-3):249–257, 2006.
- [91] H. Yang, C. Yuan, B. Li, Y. Du, J. Xing, W. Hu, and S. J. Maybank. Asymmetric 3D Convolutional Neural Networks for Action Recognition. *Pattern Recognition*, 85:1–12, 2019.
- [92] L. Yeffet and L. Wolf. Local Trinary Patterns for Human Action Recognition. In *IEEE 12th International Conference on Computer Vision*, pages 492–497. IEEE, 2009.
- [93] B.-L. Yeo and B. Liu. On the Extraction of DC Sequence from MPEG Compressed Video. In *IEEE International Conference on Image Processing*, volume 2, pages 260–263. IEEE, 1995.
- [94] YouTube, 2019. <https://www.youtube.com/>.
- [95] J. Yuan, Z. Liu, and Y. Wu. Discriminative Subvolume Search for Efficient Action Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2442–2449. IEEE, 2009.
- [96] H. Zen and H. Sak. Unidirectional Long Short-Term Memory Recurrent Neural Network with Recurrent Output Layer for Low-Latency Speech Synthesis. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4470–4474. IEEE, 2015.
- [97] G. Zhang and H. Chanson. Application of Local Optical Flow Methods to High-Velocity Free-Surface Flows: Validation and Application to Stepped Chutes. *Experimental Thermal and Fluid Science*, 90:186–199, 2018.
- [98] G. Zhao and M. Pietikainen. Dynamic Texture Recognition using Local Binary Patterns with an Application to Facial Expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):915–928, 2007.
- [99] J. Zhu, Z. Zhu, and W. Zou. End-to-End Video-level Representation Learning for Action Recognition. In *24th International Conference on Pattern Recognition*, pages 645–650. IEEE, 2018.

- [100] W. Zhu, J. Hu, G. Sun, X. Cao, and Y. Qiao. A Key Volume Mining Deep Framework for Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1991–1999. IEEE, 2016.
- [101] Y. Zhu. PyTorch Implementation of Popular Two-stream Frameworks for Video Action Recognition, 2018. <https://github.com/bryanyzhu/two-stream-pytorch>.