Deep learning for action recognition in videos

Ahmed Mazari

THÈSE DE DOCTORAT DE SORBONNE UNIVERSITÉ

Spécialité **Informatique**

École Doctorale Informatique, Télécommunications et Électronique (Paris)

Deep learning for action recognition in videos

Apprentissage profond pour la reconnaissance d'actions en videos

Présentée par

Ahmed MAZARI

Pour obtenir le grade de

DOCTEUR de SORBONNE UNIVERSITÉ

Présentée et soutenue publiquement le **Mardi 22 Septembre 2020 à 10H00**

Devant le jury composé de :

M. Frédéric Dufaux	**Rapporteur**
Directeur de Recherche au CNRS, CentraleSupelec, Université Paris-Saclay	
M. Hichem Snoussi	**Rapporteur**
Professeur à l'Université de Technologie de Troyes	
Mme. Catherine Achard	**Examinatrice**
Maître de Conférences (HDR) à Sorbonne Université	
M. Michel Crucianu	**Examinateur**
Professeur au CNAM, Paris	
Mme. Nicole Vincent	**Examinatrice**
Professeur à l'Université de Paris	
M. Hichem Sahbi	**Directeur de thèse**
Chercheur au CNRS (HDR), Sorbonne Université	

ABSTRACT

Nowadays, video contents are ubiquitous through the popular use of internet and smartphones, as well as social media. Many daily life applications such as video surveillance and video captioning, as well as scene understanding require sophisticated technologies to process video data. It becomes of crucial importance to develop automatic means to analyze and to interpret the large amount of available video data. In this thesis, we are interested in video action recognition, i.e. the problem of assigning action categories to sequences of videos. This can be seen as a key ingredient to build the next generation of vision systems. It is tackled with Artificial Intelligence (AI) frameworks, mainly with Machine Learning (ML) and Deep Convolutional Neural Networks (ConvNets).
Current ConvNets are increasingly deeper, data-hungrier and this makes their success tributary of the abundance of labeled training data. ConvNets also rely on (max or average) pooling which reduces dimensionality of output layers (and hence attenuates their sensitivity to the availability of labeled data); however, this process may dilute the information of upstream convolutional layers and thereby affect the discrimination power of the trained video representations, especially when the learned action categories are fine-grained.
In the first part of this thesis, we introduce a hierarchical aggregation design based on tree-structured temporal pyramids, for final pooling, that controls granularity of the learned representations w.r.t the actual granularity of action categories. Moreover, ConvNets are basically designed to handle vectorial data (such as still images) but their extension to non-vectorial and semi-structured data (namely graphs with variable sizes, topology, etc.) remains a major challenge. As a second part of this thesis, we introduce a Graph Convolutional Network (GCN) model based on a spectral decomposition of graph-Laplacians. It consists in learning graph Laplacians as convex combinations of other elementary Laplacians each one dedicated to a particular topology of the input graphs. Then, we introduce a pooling operator, on graphs, which achieves permutation invariance. All models are thoroughly evaluated on standard datasets and the results are competitive w.r.t the literature [1].

Keywords : Deep Video Representations, Multiple Aggregation Learning, Hierarchical Pooling, Graphs Construction, Graph Pooling and Convolution, Geometric Deep Learning

1. This work is supported by the EDITE (Ecole Doctorale Informatique, Télécommunications et Electronique de Paris) three years scholarship.

De nos jours, les contenus vidéos sont omniprésents grâce à Internet et les smartphones, ainsi que les médias sociaux. De nombreuses applications de la vie quotidienne, telles que la vidéo surveillance et la description de contenus vidéos, ainsi que la compréhension de scènes visuelles, nécessitent des technologies sophistiquées pour traiter les données vidéos. Il devient nécessaire de développer des moyens automatiques pour analyser et interpréter la grande quantité de données vidéo disponibles. Dans cette thèse, nous nous intéressons à la reconnaissance d'actions dans les vidéos, c.a.d au problème de l'attribution de catégories d'actions aux séquences vidéos. Cela peut être considéré comme un ingrédient clé pour construire la prochaine génération de systèmes visuels. Nous l'abordons avec des méthodes d'intelligence artificielle, sous le paradigme de l'apprentissage automatique et de l'apprentissage profond, notamment les réseaux de neurones convolutifs.
Les réseaux de neurones convolutifs actuels sont de plus en plus profonds, plus gourmands en données et leur succès est donc tributaire de l'abondance de données d'entraînement étiquetées. Les réseaux de neurones convolutifs s'appuient également sur le pooling qui réduit la dimensionnalité des couches de sortie (et donc atténue leur sensibilité à la disponibilité de données étiquetées) ; cependant, ce processus peut diluer l'information des couches convolutives et ainsi affecter le pouvoir discriminant des représentations vidéos obtenues, notamment lorsque les catégories d'actions apprises sont de granularités fines.

Dans la première partie de cette thèse, nous introduisons une méthode d'agrégation hiérarchique basée sur une pyramide temporelle arborescente, pour le *pooling* final, qui contrôle la granularité des représentations apprises par rapport à la granularité réelle des catégories d'actions. De plus, les réseaux de neurones convolutifs sont essentiellement conçus pour traiter des données vectorielles (telles que les images fixes) mais leur extension aux données non vectorielles et semi-structurées (à savoir des graphes de taille variable, ayant une forte variation topologique, etc.) reste un défi majeur. Dans la deuxième partie de cette thèse, nous introduisons un réseau de neurones convolutif sur les graphes basé sur une décomposition spectrale de graphes Laplaciens. Il consiste à apprendre les Laplaciens de graphes sous forme de combinaisons convexes d'autres Laplaciens élémentaires, chacun est dédié à une topologie particulière de graphes en entrée. Ensuite, nous introduisons un opérateur de *pooling*, sur des graphes, qui est invariant par permutation des noeuds. Tous les modèles sont expérimentalement évalués sur des jeux de données standards et les résultats obtenus sont compétitifs avec ceux de l'état de l'art.

Mots clés : Apprentissage de Représentations Vidéos, Apprentissage d'Aggregations Multiples, Pooling Hiérarchique, Construction de graphes, Pooling et Convolution sur les Graphes, l'Apprentissage Profond Geométrique

INSPIRING MINDS

- *" Le principal ingrédient de la première révolution quantique, la dualité onde-particule, a conduit à des inventions telles que le transistor et le laser qui sont à la base de la société de l'information. "* **Alain Aspect**

- *" Néanmoins, il reste concevable que les relations de mesure de l'espace dans l'infiniment petit ne soient pas conformes aux hypothèses de notre géométrie (géométrie Euclidienne), et, en fait, nous devrions supposer qu'elles ne le sont pas si, ce faisant, nous devrions un jour être en mesure d'expliquer les phénomènes d'une manière plus simple. "* **Bernhard Riemann**

- *" Je dirai que, alors que si, sur le plan de l'accès à la réalité empirique, la science est seule reine, en revanche elle ne jouit d'aucun privilège lorsqu'il s'agit du "fond des choses". Que là, l'émotion, artistique par exemple, se trouve (au moins!) à égalité avec elle, l'une comme l'autre ne nous fournissant que des lueurs sur un domaine qu'elles ne nous laissent qu'entrevoir. "*
  " Dans les zones tout à fait supérieures de la pensée, je fais une place à certains discrets intuitifs et intuitives au moins à tels ou tels moments privilégiés qu'ils ont connus. Un nombre infime d'entre eux est parvenu à s'exprimer par le moyen de la grande littérature. Les autres gardent le silence: mais je sais qu'ils sont là, présents. " **Bernard d'Espagnat**

- *" Comme on le sait, le dernier continent inconnu à l'homme est l'homme, et le centre de ce continent, le cerveau, nous est non seulement inconnu, mais encore incompréhensible. "*
  " Comprendre, ce n'est pas tout comprendre, c'est aussi reconnaître qu'il y a de l'incompréhensible. "
  " La connaissance est une navigation dans un océan d'incertitudes à travers des archipels de certitudes. " **Edgar Morin**

- *" Des pensées sans matière sont vides, des intuitions sans concepts sont aveugles. "*
  " L'entendement ne peut rien percevoir, ni les sens rien penser. La connaissance ne peut résulter que de leur union. " **Emmanuel Kant**

- *" Il suffit de regarder une chose avec attention pour qu'elle devienne intéressante. "*
  Eugenio d'Ors Y Rovira

- *" La connaissance du réel est une lumière qui projette toujours quelque part des ombres. "*
  " Le réel n'est jamais ce qu'on pourrait croire, mais il est toujours ce qu'on aurait dû penser. " **Gaston Bachelard**

- *" L'homme raisonnable s'adapte lui-même au monde ; l'homme déraisonnable continue à essayer d'adapter le monde à lui-même. Donc, tout progrès dépend de l'homme déraisonnable. "* **George Bernard Shaw**

- *" La philosophie procède par variation, l'art par variété et la science par variable. La philosophie se dessine sur le plan d'immanence, l'art sur le plan de composition et la science sur le plan de référence. La philosophie procède par concept, l'art par sensation et la science par connaissance. "* **Gilles Deleuze**

- *" Une mesure exacte vaut l'avis d'un millier d'experts. "* **Grace Hopper**

- *" L'intelligence dans ce qu'elle a d'inné est la connaissance d'une forme, l'instinct implique celle d'une matière. "* **Henri Bergson**

- *" La vérité sera toujours à trouver dans la simplicité, et non dans la complexité et la confusion des objets. "* **Isaac Newton**

- *" Les principes de la théorie sont dérivés, comme ceux de la mécanique rationnelle, d'un très petit nombre de faits primaires, dont les causes ne sont pas considérées par les géomètres, mais qu'ils admettent comme les résultats d'observations communes confirmées par toute expérience. "* **Joseph Fourier**

- *" Le fondement de l'intuition parmi nous vient du désir de se transporter hors de soi. "*
  " Penser, c'est ce que nous savons déjà n'avoir pas encore commencé à faire. " **Jacques Derrida**

- *" Le critère de la scientificité d'une théorie réside dans la possibilité de l'invalider, de la réfuter ou encore de la tester. "* **Karl Popper**

- *" Soit les mathématiques sont trop grandes pour l'esprit humain, soit l'esprit humain est plus qu'une machine. "* **Kurt Gödel**

- *" Sans la curiosité de l'esprit, que serions-nous ? Telle est bien la beauté et la noblesse de la science : désir sans fin de repousser les frontières du savoir, de traquer les secrets de la matière et de la vie sans idée préconçue des conséquences éventuelles. "* **Marie Curie**

- *" Au sens positif du temps on peut dire : seul le présent est, l'avant et l'après ne sont pas mais le présent concret est le résultat du passé et il est plein de l'avenir. Le présent véritable est, par conséquent, l'éternité. "* **Martin Heidegger**

- *" Un scientifique est heureux, non pas en se satisfaisant de ses accomplissements mais en étant constamment dans la recherche de nouvelles connaissances. "* **Max Planck**

- *" Chaque grande et profonde difficulté porte en elle-même sa propre solution. Elle nous oblige à changer notre façon de penser pour la trouver. "* **Niels Bohr**

- *" Pour atteindre la vérité, il faut une fois dans sa vie, se défaire de toutes opinions que l'on a reçues, et reconstruire à nouveau et dès le fondement le système de ses connaissances. "***René Descartes**

- *" Il est admissible qu'un exotériste ignore l'ésotérisme, bien qu'assurément cette ignorance n'en justifie pas la négation; mais, par contre, il ne l'est pas que quiconque a des prétentions à l'ésotérisme veuille ignorer l'exotérisme, ne fût-ce que pratiquement, car le «plus» doit forcément comprendre le «moins». "* **René Guénon**

- *" Qu'importe à quel point ta théorie soit belle, que tu sois intelligent. Si elle est en désaccord avec l'expérience, c'est qu'elle est fausse. "* **Richard Feynman**

- *" La science progresse mieux lorsque les observations nous obligent à modifier nos idées préconçuess. "* **Vera Rubin**

- *" Paris est la grande salle de lecture d'une bibliothéque que travarse la Seine. "* **Walter Benjamin**

- *" L'observateur influence l'observation. L'observateur ne peut être séparé de ce qu'il observe. Sans observateur, pas de réalité à observer. "*
  " Quand je rencontrerai Dieu, je Lui poserai deux questions : Pourquoi la relativité ? Et pourquoi la turbulence ? Je pense réellement qu'il aura une réponse pour la première " **Werner Heisenberg**

- *" La réussite est la capacité de passer d'un échec à un autre sans perte d'enthousiasme. "* **Winston Churchill**

REMERCIEMENTS

J'aimerai remercier toutes les personnes qui ont contribué de prés ou de loin à l'aboutissement de cette thése.

Tout d'abord, je tiens à remercier mon directeur de thèse **Hichem Sahbi** pour m'avoir donné l'opportunité de faire cette thèse et pour son suivi scientifique de qualité pendant ces trois années.

Je remercie **Frédéric Dufaux** et **Hichem Snoussi** pour le temps qu'ils ont consacré à rapporter ce manuscrit ainsi qu'à **Catherine Achard**, **Michel Crucianu**, **Michel Crucianu**, **Nicole Vincent** pour leur participation à mon jury de soutenance.

Un grand merci aux copains et copines du labo, pour l'excellent environnement de travail, le soutien moral, l'entraide, l'esprit de partage, l'ambiance avec qui j'ai passé de très bons moments : **Rémi Cadene, Perrine Cribier-Delande, Clara Gainon de Forsan de Gabriac, Antoine Saporta, Micael Carvalho, Etienne Simon, Hedi Ben Younes, Thomas Robert, Arthur Pajot, Tom Veniat, Vincent Grari, Arthur Douillard, Jérémie Donna, Mickael Chen, Yifu Chen, Ibrahim Ayed, Corentin Dancette, Daniel Brooks, Taylor Mordan, Agnès Mustar, Adrien Pouyet, Jean-Yves Franceschi, Vincent Grari, Arnaud Dapogny, Clément Rebuffel, Emmanuel De Bezenac, Marie Dechelle, Robin Dupont, Haoming Zhan, Mathieu Crilout, Eloi Zablocki**.
Je remercie aussi, les permanents de l'équipe MLIA ainsi que les personnes du LIP6 pour le soutien moral, et pour l'aide, en particulier **Marie-Jeanne Lesot**, **Matthieu Cord**, **Edouard Oyallon** (aussi pour les discussions scientifiques), **Christophe Bouder**, **Nadine Taniou**, ainsi que **Marin Ferecatu** du CNAM Paris.

J'aimerai également remercier mes professeurs de *machine learning* à Paris Sud et à Paris Descartes : **Isabelle Guyon** (pour ses encouragements et pour les discussions sur l'avenir de *machine learning*), **Yohann Tendero** (pour la qualité de ses cours en traitement de signal et images qui m'ont beaucoup inspiré), **Michele Sebag** (de m'avoir accueilli en tant que stagiaire de M1 au sein de son équipe de recherche à l'INRIA TAO), **Mohamed Nadif, Lazhar Labiod, Themis Palpanas** (avec qui j'ai commencé à travailler sur les réseaux de neurones) et **Laurent Wendling** (pour son cours portant sur la reconnaissance des formes en lien avec les réseaux de neurones qui a joué un rôle prépondérant dans la passion que j'ai développé pour la vision par ordinateur). Je tiens aussi à remercier mes deux

encadrants de stage de M1 à l'INRIA TAO : **Cyril Furtlehner, Aurélien Decelle** pour le temps consacré à mon apprentissage et de m'avoir fait découvrir les réseaux de neurones d'un point de vue *physique statistique*. Puis je remercie mes deux encadrants de stage de M2 à *Dhatim* : **Faris Avdagic, Pierre de Chastellier** de m'avoir fait confiance pour développer mon premier réseau de neurones en production (reconnaissance optique de caractères).
Vient ensuite le tour de mes enseignants à l'Université de Béjaia (Algérie) qui m'ont vachement inspiré et encouragé pour faire de la recherche : **Toufik Amayas Mostefaoui** (pour les longues discussions sur la philosophie et la physique), **Hachem Slimani** (pour sa rigueur scientifique et son cours de la théorie des graphes) et **Moumen Hamouma** (qui m'a initié à la recherche en systèmes distribués et avec qui j'ai appris l'autonomie scientifique).

Un grand merci à ma famille (mes parents, ma sœur, cousines, cousins, tantes, oncles, mes grands parents), mes ami.e.s pour leur soutiens et d'avoir supporté ma longue absence pendant ces trois années de l'autre coté de la méditerranée.

TABLE DES MATIÈRES

TABLE DES FIGURES

LISTE DES TABLEAUX

ACRONYMS

AI	Artificial Intelligence
ANNs	Artificial Neural Networks
AR	Action Recognition
BDRNN	Bi-Directional Recurrent Neural Network
BoVW	Bag of Visual Words
ConvNets	Convolutional Neural Networks
CL	Convolutional Layers
CV	Computer Vision
DL	Deep Learning
DMKL	Deep Multiple Kernel Learning
ESURF	Enhanced Speeded-Up Robust Features
FL	Fuzzy Logic
FPS	Frames per Second
FV	Fisher Vector
GCN	Graph Convolutional Network
GDL	Geometric Deep Learning
GP	Global Pooling
GPUs	Graphics Processing Units
HOF	Histogram of Optical Flow
HOG	Histogram of Oriented Gradients
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
LSTMs	Long Short Term Memory
MBH	Motion Boundary Histogram
MKL	Multiple Kernel Learning
ML	Machine Learning
MLP	Multi-Layer Perceptron
NLP	Natural Language Processing
OCR	Optical Character Recognition
ReLU	Rectified Linear Unit

RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
SIFT	Scale-Invariant Feature Transform
SPM	Spatial Pyramid Matching
SR	Speech Recognition
STIP	Space Time Interest Point
STISM	Spatial-Temporal Implicit Shape Model
SVMs	Support Vector Machines
t-SNE	t-distributed Stochastic Neighbor Embedding

CHAPITRE 1

INTRODUCTION

Contents

1.1 Context

Since the dawn of the industrial revolution, traditional economical activities have been transformed following the upgraded technologies. The latter have experienced a galloping rush thanks to the emergence of brand-new engineering fields in order to oversee their development and hence to respond to the need of contemporary society that keeps evolving [1].
Among these engineering fields, Artificial Intelligence (AI) has drawn a lot of attention and has involved both academic and industrial partners to support societal challenges with concrete solutions [2].
AI can be defined as a field of study which ultimately aims at designing intelligent machines capable of mimicking human intelligence and beyond [3, 4]. It has several advantages ; on the one hand, it plays a key role in solving complex industrial tasks automatically with the least possible error rate, and on the other hand, it helps making progress in human intelligence understanding and its philosophical implications.
We distinguish three types of AI : *Weak*, *Strong* and *Super* AI [5, 6]. *Weak* AI emphasises on achieving specific tasks accurately, it is also known as specialized AI. *Strong* AI is able to achieve anything that humans can do. *Super* AI is the ultimate

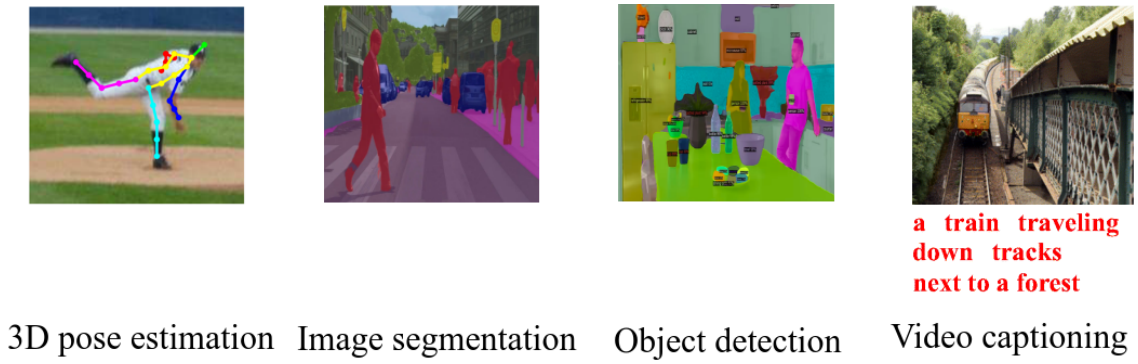


FIGURE 1.1 – Examples of different tasks in Computer Vision (CV).

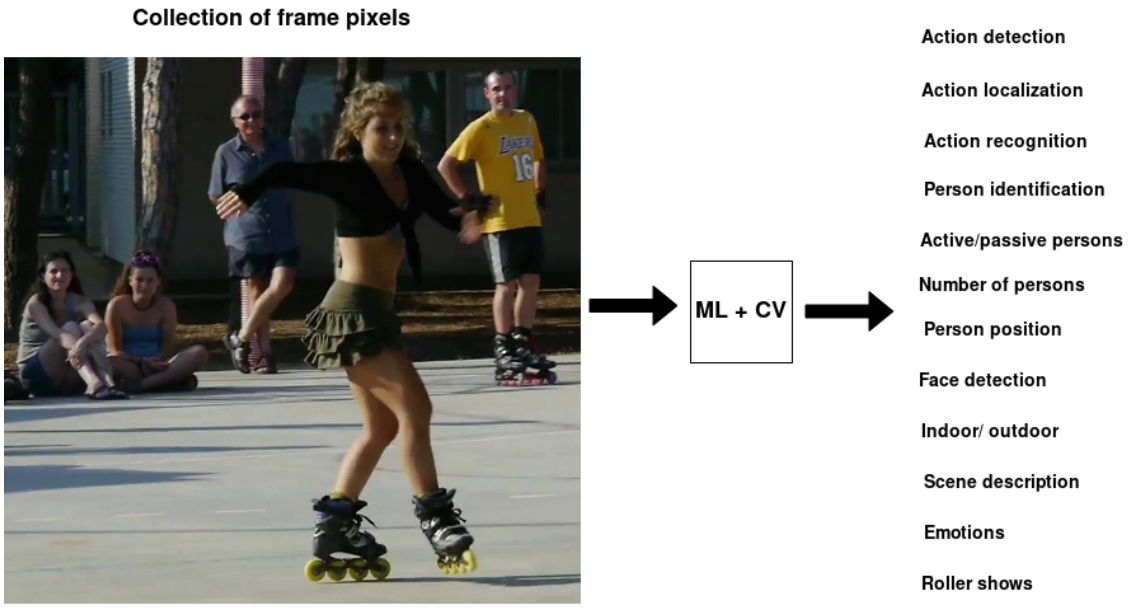


FIGURE 1.2 – Example of video analysis task.

purpose which aims at surpassing the human ability, doing what human is not able/imaginable to do. The current understanding of AI is limited and is still at the era of *weak* AI where there is a lot of progress and effort to do. Despite the fact that AI has been oscillating between waves of optimism and several AI winters [7-9] for decades, it comes back as a key technology in miscellaneous domains.

Computer Vision (CV) is one of the rapidly growing and emerging sub-fields of AI, whose goal is to understand, extract meaningful and semantic information from visual scenes such as images and videos. Applications of CV include : 3D pose estimation, image segmentation, object detection, video captioning etc. (see examples in Figure 1.1).

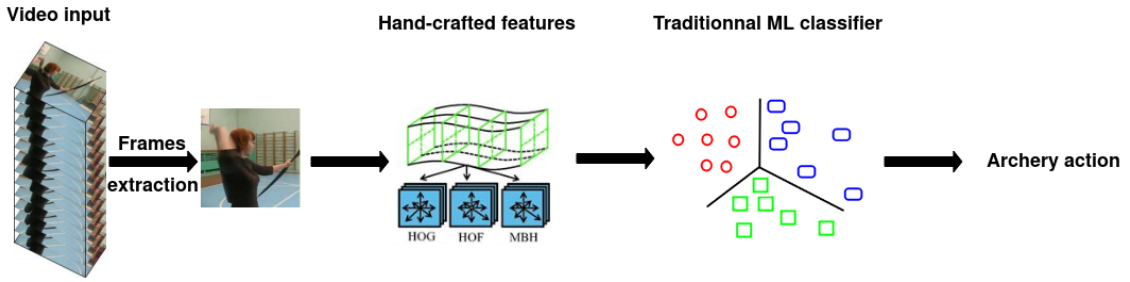


FIGURE 1.3 – Example of Handcrafted representation (based on Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF) and Motion Boundary Histogram (MBH)) and shallow classification method.

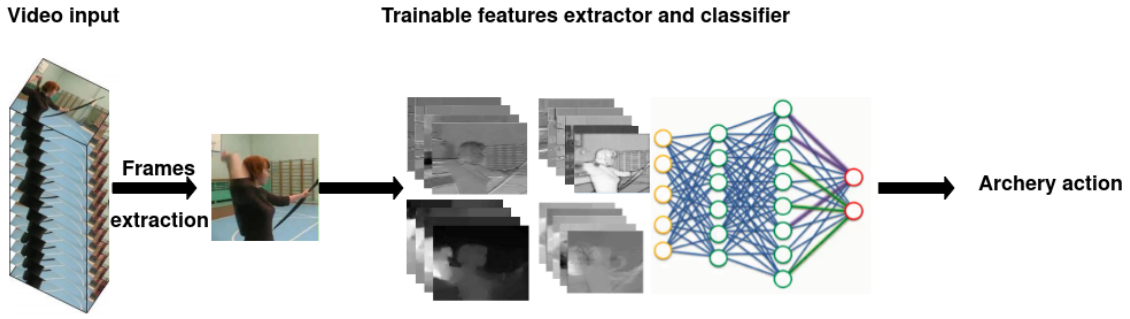


FIGURE 1.4 – Example of learned representation and deep classification method.

In this thesis, we are interested in video analysis, particularly in the task of Action Recognition (AR). One of the motivations of this specific task resides in the substantial increase of video contents while their manual annotation becomes out of reach. For that reason, there is a need of reliable automatic video analysis solutions able to annotate these large collections of data. One of the difficulties is that video data are pixel-based carrying no explicit information. From a collection of pixels, CV methods seek to transform them into meaningful information, abstract them into high level features and concepts such as assigning action category to a sequence of video frames, detecting and localizing persons, extracting their pose, describing scenes etc. (see Figure 1.2).

One of the successful solutions that learn high level abstraction concepts from raw video is Machine Learning (ML). Given a sample of labeled videos, referred as training data, ML aims at designing decision criteria that assign action categories to unseen data as illustrated in Figure 1.3. ML has achieved a rapid growth and has drawn lots of attention thanks to the recent advances and success of Deep Learning (DL) which is a subfield of ML based on biologically-inspired Artificial Neural Networks (ANNs).

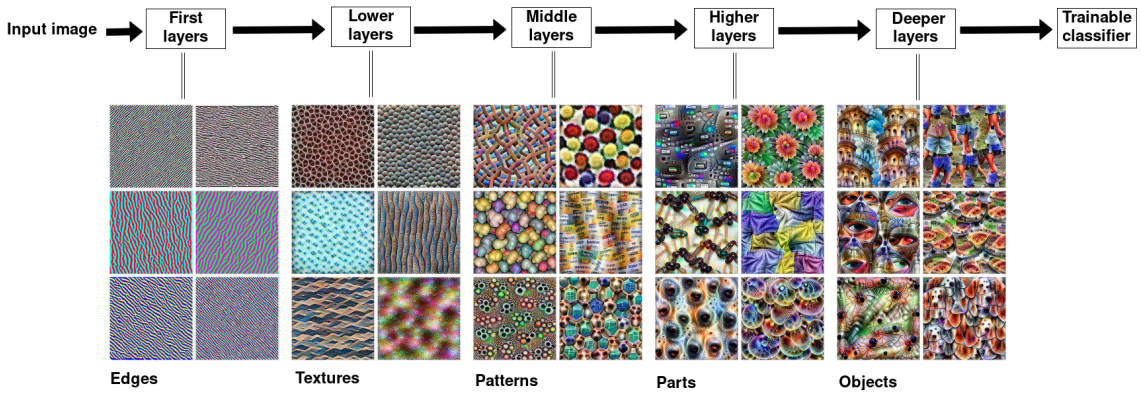


FIGURE 1.5 – Trainable deep feature extractor and classifier. The images below the scheme are taken from [10].

ANNs are trainable feature extractors that provide highly discriminant representations for classification from raw data in a hierarchical process as illustrated in Figure 1.4. Early layers are dedicated to low-level visual characteristics including edges and gradient orientation, intermediate layers provide mid-level characteristics such as shapes and structures while deeper layers are dedicated to high-level semantic concepts such as faces and objects (see Figure 1.5).

The embryonic stage of DL dates back to 1980s [11-13]. However, it has experienced two decades of hibernation following the success of Support Vector Machines (SVMs), kernel methods during the 1990s and the lack of large amount of learning data required for DL, as well as powerful computer resources to process them in a reasonable time.

The resurgence of DL happened in 2012, in a CV competition of image classification called ImageNet Large Scale Visual Recognition Challenge (ILSVRC). This 2012 edition was marked by the lightning win of AlexNet method [14] ; DL model based on Convolutional Neural Networks (ConvNets) (see Figure 1.6).
This major success of DL is mainly due to : (i) the availability of a large labeled dataset of about 1.2 million images, which was at least two orders of magnitude larger than any other existing datasest and (ii) the advances in parallel programming : graphics processing units and recently tensor processing units. Since then, DL has been extended and applied to neighboring tasks in CV including action recognition in videos [15].

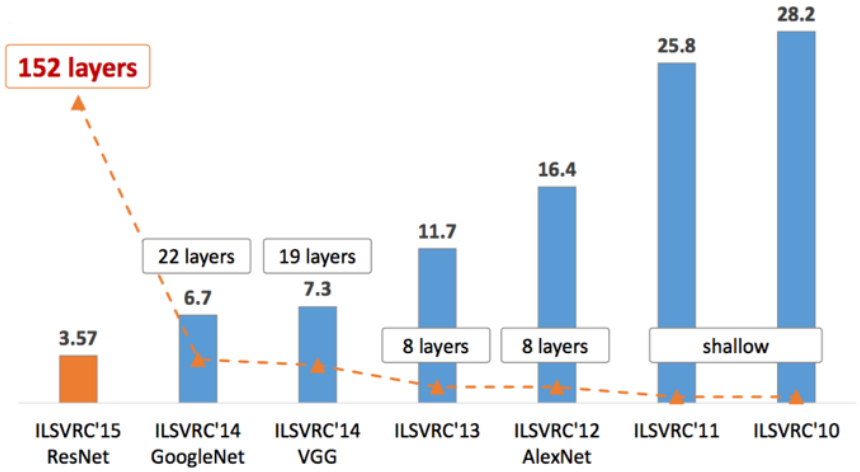


FIGURE 1.6 – Evolution of performances of ImageNet Large Scale Visual Recognition Challenge (ILSVRC) along with depth of networks. Picture credit for [16].

1.2 Statistical Supervised Learning at a Glance

Broadly speaking, ML is a field of study that gives computers the ability to learn without being explicitly programmed [**Arthur Samuel, 1959**]. ML models aim at solving specific tasks by improving their generalization ability from training data. This generalization ability is usually estimated with some performance measure.

We consider action recognition as a supervised learning task, i.e., an interpolation problem in high dimensional space. The goal is to build an unknown function f accessible only through annotated videos dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ of N examples, with $x_i \in \mathcal{X}$ being the input video sequence and $y_i \in \mathcal{Y}$ its corresponding correct action category, known as label.

We denote $\mathcal{F}$ as the space of functions that maps $\mathcal{X}$ to $\mathcal{Y}$. Thereby, the learning problem consists at finding the function $\hat{f} \in \mathcal{F}$ that best fits the dataset $\mathcal{D}$. In order to measure the prediction of $f(x_i)$ w.r.t the ground truth y_i, a loss function l is introduced. Hence, we define the regularized loss $\mathcal{L}$ of f over $\mathcal{D}$ as

$$\mathcal{L}(f, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} l(f(x_i), y_i) + R(f, \mathcal{D}). \tag{1.1}$$

The second term $R(f, \mathcal{D})$ in Equation 1.1, is called a regularization term which corresponds to a prior of f and makes it possible to tradeoff between data under-

fitting and overfitting so that f better generalizes to unseen samples. In the above equation $l,\ R$ are data dependent, their choice vary from one task to another following the problem to solve. Finally, the optimal model $\hat{f}$ is found as

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \mathcal{L}(f, \mathcal{D}). \tag{1.2}$$

In practice, we use a muti-class cross-entropy loss function l to measure the prediction accuracy of our models :

$$l(f(x_i), y_i) = -\sum_{c=1}^{C} y_i^{(c)} log(f_c(x_i)) + (1 - y_i^{(c)}) log(1 - f_c(x_i)). \tag{1.3}$$

with C being the total number of action categories (classes), $y_i^{(c)}$ a binary indicator (0 or 1) whose value depends on whether action category c is the correct classification for the example x_i, and $f_c(x_i)$ the output of the i-th example corresponding to its predicted probability.
The performances of our models are measured with accuracy metric on test data. This metric is defined as the ratio of the number of correct predictions to the total number of examples.

1.3 Action Recognition

1.3.1 Task Definition

Video **action recognition** consists in assigning action categories to sequences of observed videos. It requires capturing context encoded in the whole video rather than at single frame. Action within sequence of frames may be performed throughout the whole video or at specific part depending on the video trimming process. For that reason, some videos encode only the actions of interest while others encode extra context.

Action recognition is similar to image classification and can be seen as an extension of static image classification to multiple frames classification. The latter is quite challenging since the score of classification requires prediction at video level rather than at frame (image) level. Moreover, the strategies used to design architectures that are capable to capture spatio-temporal information is not-trivial and expensive. The possible strategies include : (i) end-to-end training or feature extraction and classification in two separated processes, (ii) spatio-temporal net-

work or (iii) two-streams network, separately for spatial and temporal information.

Existing action recognition techniques are usually based on ML [17-22]; their general principle consists first in describing video frames using handcrafted or learned representations [23-25] as illustrated in Figure 1.3 and in Figure 1.4 prior to assigning these representations to action categories using variety of ML algorithms including SVMs and DL [26-28].

If DL strategies for image classification grow rapidly, progress in architecture design and learning representations for action recognition is slow. It can be explained by the expensive computational cost of architecture search. For instance, 2D ConvNets for classifying 101 classes [1] have about 5 million parameters while 3D ConvNets (for the temporal dimension) include about 33 million parameters [29].

1.3.2 Application Domains

In our daily life, many applications need real-time action recognition module to solve tasks such as : scene understanding [30] and video captioning [31], as well as video surveillance [32] as shown in Figure 1.7. Their automation helps in drastically reducing the human labor in analyzing such abundant amount of visual contents.

Video surveillance. For security reasons, several public areas are placed under surveillance to identify suspicious and infrequent actions. Video surveillance system is composed of a number of cameras powered by action recognition algorithms to automatically supervise environments in real-time and to increase the accuracy of capturing non-common actions such as criminal actions.

Video retrieval. Internet is characterized by an abundant amount of data. Visual data are of particular interest where people keep uploading and sharing videos on different applications and websites. However, retrieving videos according to their contents is challenging since most search engines operate on text queries to manage video data. Text queries can be expressed as keywords, tags, person names, titles. They can be difficult to express, inaccurate and may not fit the targeted visual content. For that reason, alternative methods based on video retrieval framework [33] have been designed to seek accurately for appropriate

1. UCF-101 video dataset has a comparable number of frames with ImageNet

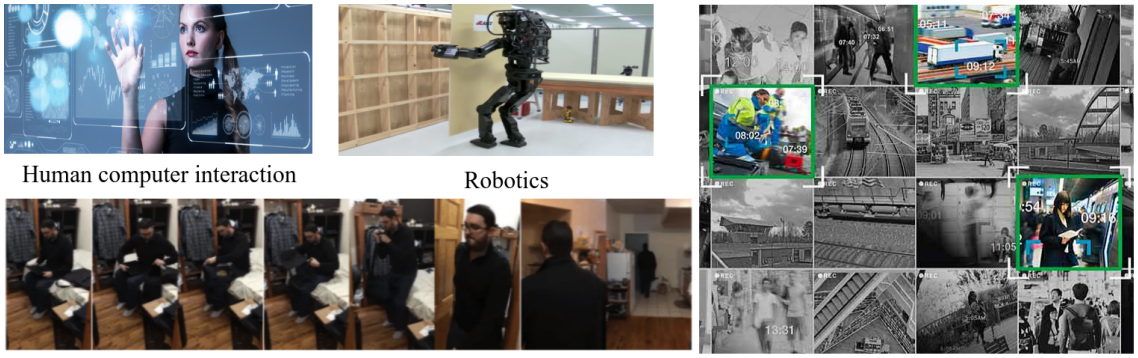


FIGURE 1.7 – Applications of action recognition.

videos.

Robotics and human-computer interaction. A wide range of applications in domotic and industrial environments get benefits and facilities from human-robot interaction. For instance, person with disabilities or with reduced mobility may interact with a robot to perform certain tasks such as opening doors and welcoming guests, supervising children in parks, etc.

Autonomous vehicle driving. A vehicle is equipped with a set of camera sensors endowed with action recognition and prediction algorithms. For safety reasons, these sensors can help to avoid unexpected collisions with pedestrians by localizing persons and predicting their actions following real-time motion trajectory analysis.

1.3.3 Challenges

Video action recognition is among the most challenging problems in CV. It is related to intrinsic and extrinsic acquisition conditions of videos such as their annotation process, their trimming, unconstrained environment and to the capture conditions, their large intra-class and small inter-class variability, velocity, truncation, pose variation as well as variable subject scales which rise (i) the difficulty to learn mapping models that assign action categories to sequences of frames while being resilient to these acquisition conditions, and (ii) the hardness in hand-labeling and trimming large collections of training videos prior to designing these mapping models. The major challenges are presented below and illustrated in Figure 1.8.

- **Occlusion** : can be related to crowded scenes, camera motion, subject hiding discriminating parts of actions. Occlusion is an impediment for cameras as well as for human eye which can't see through the wall and hence can sense only visible objects [34]. A solution based on setting multiple cameras with different viewpoints has been conceived to mitigate the effect of occlusion [35, 36].

- **Illumination** : this highly depends on the type of sensors and cameras [37], as well as on lighting conditions. Low lighting condition leads to object confusion and occlusion [35]. Two images describing the same visual content under different illumination conditions have different pixels distributions. As a result, ML models may assign them to incorrect labels.

- **Viewpoint change and camera motion** : In the case of human actions, frontal and profile human poses result into different appearance information [38, 39] which may lead to confusion even if they belong to the same action category. Moreover their tracking provides different trajectories.

- **Motion blur** : may be caused by the speed of moving objects, persons and/or camera, as well as the difficulty for the latter to track motion accurately beyond a given speed rate [40-42]. Moreover, visual scenes are captured by camera with finite shutter speed [43] which results into the violation of brightness constancy and hence results into inaccurate optical flow estimation of pixels displacement.

- **Frame rate** : subtle details can be ignored by low frame rates [44] while high frame rates capture better the sudden change in trajectories and is well adapted for fine-grained actions. However, annotation is more time demanding with high frame rates [45].

- **Trimming** : it is common that videos are not trimmed in the same way which leads to different semantic information [46]. Some videos are endowed with larger context including the one of the action while others include only the action of interest [47]. As a consequence, ML models provide richer representations to videos with larger contexts and may confuse between videos belonging to different action categories due to the lack of context.

- **Background clutter** : can be related to arbitrary activities of people behind actions such as their free motion (behind scenes) [48]. This may lead to a strong fluctuating background in the clutter and to difficulties in distinguishing and tracking correctly the body parts of different people [49]. Moreover, instable lighting conditions render the task of clutter elimination more challenging.

- **Multiple persons** : this results into difficulties to distinguish persons participating actively in actions from passive ones and to re-identify incoming and outgoing persons [49]. These difficulties are compounded by the fact that several actions may occur simultaneously in the scene. Moreover, multiple persons can also be the source of occlusion [50] and of cluttered background due to persons overlapping, as well as of wrong tracking due to pose and speed variation.

- **Video duration** : sequences of frames with different durations result into temporal information imbalance. Real world applications involve videos with different duration, ranging from a few seconds to a few minutes [51]. Videos with long duration encode larger context and provide richer information than shorter ones [52]. However, long videos may include spurious details that lead to confusion or provide extra helpful details for better discrimination. Whereas short videos provide the necessary details for action classification, they may be incomplete to build discriminant representations especially for fine-grained actions.

In this thesis, we are interested in building robust learning approaches to tackle some of these challenges by designing appropriate ML and DL models, as well as discriminant representations for classification.

1.4 Motivation

The main power of DL models resides in their ability to learn representations, well suited to the targeted tasks, from raw data. However, the design of appropriate network architectures for specific tasks remains open. As a consequence, the emphasis has shifted from handcrafted representations design to network architectures design.

Success of DL comes back to *ResNet* model [16] thanks to its skip connections which attenuate the vanishing/exploding gradient problem [53]. Moreover, the scalability of DL to large datasets such as *ImageNet* [54] for image classification has aroused the interest of neighboring communities such as video analysis [15, 55, 56].
In the latter, the task consists at generalizing DL to video databases, which are characterized by an extra temporal dimension and at achieving videos classification. Three main problems occur : (i) video datasets are at least two orders of magnitude smaller compared to images datasets [55] while video tasks such as action recognition and prediction are more challenging due to their uncontrolled

FIGURE 1.8 – Common challenges in action recognition.

acquisition conditions, (ii) the success of DL models is tributary to the availability of large enough labeled training data [54], (iii) the difficulty to model moving subjects (persons and objects) across the sequences of frames [57, 58] and to encode explicitly their temporal structures [59].
Moreover, DL models operate directly on vectorial (raw) data such as images, described by a collection of pixels on a regular grid [60-63]. However, wide spectrum of applications such as video analysis as depicted in Figure 1.9 require handling non-vectorial data, mainly graphs (semi-structured data) such as skeleton in action recognition where complex geometric relationships between moving parts should be considered.
In order to handle non-vectorial data, existing methods [64] first vectorize graphs by yielding an embedding of graphs prior to learning their representations and classifying them with DL models.
One of the main drawbacks of graph vectorization is the loss of structural information. For instance, in human action recognition, moving objects can be seen as a constellation of interacting body parts which come in the form of 2D/3D skeletons described by a set of joints relying on the human body connectivity [65-67]. For that reason, we argue that non-vectorial models are well suited to encode the spatio-temporal relationship of body parts in videos.

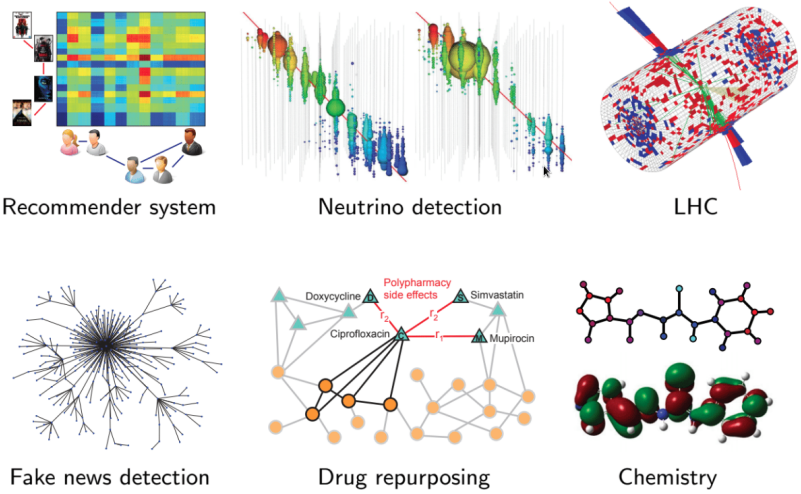


FIGURE 1.9 – Applications of non-vectorial Deep Learning (DL) on graphs. Picture credit for [84]

This thesis studies questions related to DL design and to video representations for the specific task of action recognition :

- Building Global Pooling (GP) function for video classification
- Achieving DL on non-vectorial data : convolution and pooling on graphs

Global pooling for videos. GP function aggregates the convolutional feature maps to build a global representation prior to fully connected layer in hierarchical way, by alternating with convolutional layers and non-linearities [14] such as Rectified Linear Unit (ReLU) or applied only at the end of the back-end of Convolutional Layers (CL). It plays the role of dimensionality reduction operator where the operation can be max or averaging [68] over windows or regions of pixels. As a result, it reduces the computational complexity and the number of parameters of networks [68-70]. It helps also to keep the most informative information and to learn localized representations capable to generalize to different targeted tasks [70-74].

In action recognition, pooling plays a key role in enhancing the resilience of DL models to the lack of training videos [75-83]. It helps discriminating coarse-grained action categories while leading to a loss in discrimination power of fine-grained action categories. As a first part of work in this thesis, we are interested in designing a GP operator well suited to the task of action recognition that controls its granularity level. This topic will be covered in *Chapter 3*.

DL on semi-structured (Graph) data. Recently, several fields such as quantum physics [85], biology [86] and chemistry [87] need to deal with non vectorial data. The remarkable success of DL in large scope of applications including CV [14], Speech Recognition (SR) [88] and Natural Language Processing (NLP) [89] brought on a keen interest in generalizing DL to non-vectorial data : *graphs* and *manifolds*, denoted Geometric Deep Learning (GDL).

The generalization of DL to GDL is intrinsically related to a design principle of Graph Convolutional Network (GCN) based on analogous properties of ConvNets, namely : locality, translation invariance and equivariance, compositionality [63, 90-93], as well as a linear computational complexity in learning. Locality stands for local invariance, an important property to capture intra-class variation where similar representations are associated to similar regions independently of their spatial locations in images. Translation invariance is also known as stationarity which is a key property for convolutional layers. It shows that ConvNets produce the same response regardless how their input images are shifted while translation equivariance means that the response of receptive fields varies equally with distortion [91]. Compositionality is a property inspired by the visual cortex of the brain [94]. It consists at aggregating simple structures in hierarchical way to build high level abstract structures as shown in Figure 1.5.
To do so, during the two last decades, signal processing community has been working actively to extend Fourier transform on graphs [60]. The techniques rely on harmonic analysis combined with graph theory to build a well grounded Fourier basis on graphs. They are based either on the eigen-decompostion of graph Laplacian [95, 96] or on graph wavelets [97, 98].

The success of early GCN is observed on graphs with known and fixed topology such as 2D/3D regular grids characterized by a fixed number of nodes and edges, as well as constant degree. For instance, [99] propose a GCN based on spectral convolutional operator for Optical Character Recognition (OCR) on a widely used benchmark, namely *MNIST*. They show that their GCN is able to find out Fourier basis on graphs (of regular grids) and hence provide competitive results compared to ConvNets.

The main drawback of this GCN resides in its unsuitability to general graphs with arbitrary topological characteristics such as variable number of nodes/edges and heterogeneous degrees [100]. Moreover, in both ConvNets and GCN on regular grids, pooling operator is well defined, it is invariant to node permutation and node reordering by construction. This invariance is not satisfied for general graphs, hence it requires a careful design of pooling operator, known also as graph

coarsening [101, 102]. In addition to locality, translation invariance/equivariance, compositionality, invariant pooling operator to node permutation needs to be considered for learning on general graphs.

Since the first works on GDL [99], few emerging solutions attempt to extend these models to action recognition, including [103] which models connectivity of moving joints in videos using graphs where nodes correspond to joints described by their spatial coordinates and their likelihood and edges characterize their spatio-temporal interaction. One of the disadvantages of this extension resides in the low power expressivity of joints representations which are deprived from rich motion and appearance information.

The reasons of slow development of GDL for the task of action recognition are several : lack of principled convolutional and pooling operator on general graphs, difficulty to build graphs from visual scenes following the challenges discussed in Section 1.3.3, as well as the appropriate network architecture (*ResNet* in the case of ConvNets).

As a second part of work in this thesis, we are interested in designing GDL models based on graphs for the specific task of action recognition in videos. The contributions include a design principle of convolutional and pooling operator on graphs, as well as graphs construction from 2D/3D skeletons and videos frames while being invariant to arbitrary reordering of objects in the scenes especially for highly complex ones with multiple interacting objects and persons. We devote *Chapter 4* to study these issues.

1.5 Contribution and Outline

In the following chapters of this thesis, we study and propose practical solutions to tackle some of the challenges presented in Section 1.3.3 and to solve efficiently the task of action recognition in videos by investigating the appropriate methods discussed in Section 1.4 based on the limitations of current state-of-the-art and hence research axes which are not sufficiently explored. Our contributions are summarized in Figure 1.10.

In order to tackle the aforementioned issues, this thesis is structured as follows :

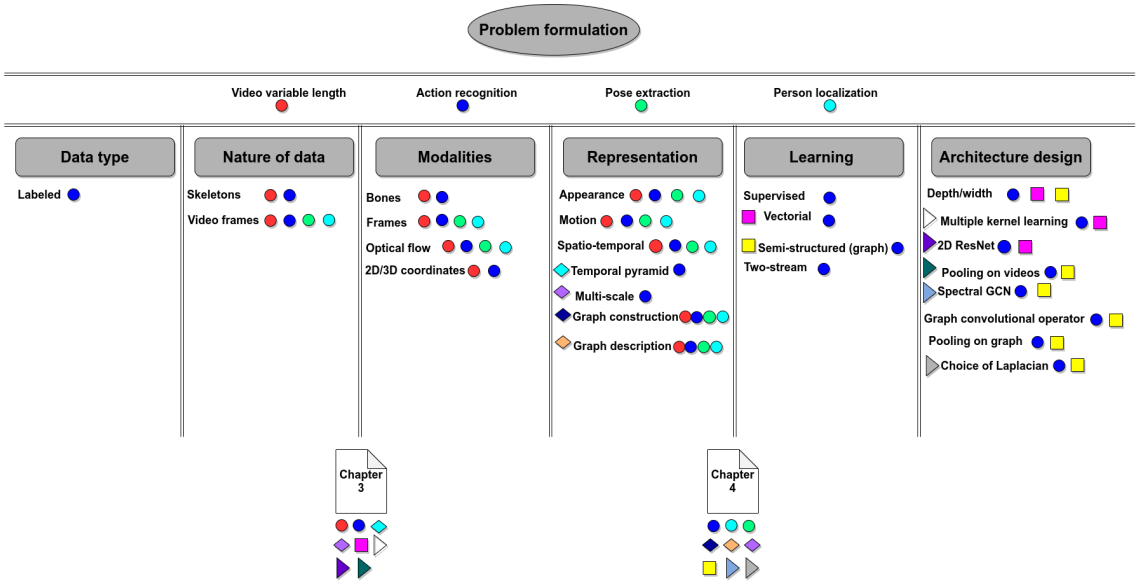


FIGURE 1.10 – Overview of our contributions and keywords.

- Chapter 2 : we give an overview of existing state-of-the-art methods in video action recognition, including handcrafted, learned representations as well as shallow and deep classifiers to provide a background for the present work.

- Chapter 3 : we propose different hierarchical pooling methods based on temporal pyramids and on Multiple Kernel Learning (MKL) to : (i) solve the problem of video variable length while preserving its temporal structure, (ii) control large context variation in videos. These methods allow also to control the granularity of video representations w.r.t the ground truth of action categories. The proposed solutions are based on solving a constrained minimization problem whose solution corresponds to the distribution of weights associated to the nodes of the temporal pyramid. Moreover, we propose a hierarchical pooling layer, learned in an end-to-end and in a differentiable manner along with *ResNet*, and a surrogate back-propagation algorithm to train large video datasets in a reasonable time. In contrast to existing methods which preprocess videos to extract a fixed sample of frames prior to feeding them to DL models and lead to information loss especially for fine-grained actions, our method leverages all the frames during the training process by alternating between different frames through the iterations.

- Chapter 4 : we propose a method to build graph inputs to train Geometric Deep Learning (GDL) models based on 2D/3D skeletons and on video frames for both appearance and motion modalities by exploiting the recent advances in human pose estimation and extraction, and a pooling operator

on graphs which is invariant to node permutations. Pooling is achieved in two steps : context-dependent node expansion followed by a global average pooling. We also propose a spectral graph convolutional network based on convex combination of several Laplacians to learn a highly non-linear graph Laplacian, each one is dedicated to a particular topology of the input graphs. Finally, we propose a temporal pyramid GCN that captures different levels of granularity inspired by the methods of *Chapter 3* and by inception network [104] to design effective convolutional operators on graphs.

- Chapter 5 : we summarize our contributions, then we open a discussion about some research directions for a future work.

1.6 Related Publications

This thesis is based on the material published in the following papers and summarized in Figure 1.11 :

- Ahmed Mazari, Hichem Sahbi, " Deep Temporal Pyramid Design for Action Recognition, " In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2019
- Ahmed Mazari, Hichem Sahbi, " MLGCN : Multi-Laplacian Graph Convolutional Networks for Human Action Recognition, " In the 30th British Machine Vision Conference (BMVC). 2019
- Ahmed Mazari and Hichem Sahbi. Coarse-To-Fine Aggregation For Cross-Granularity Action Recognition. In the 27th IEEE International Conference on Image Processing (ICIP). 2020

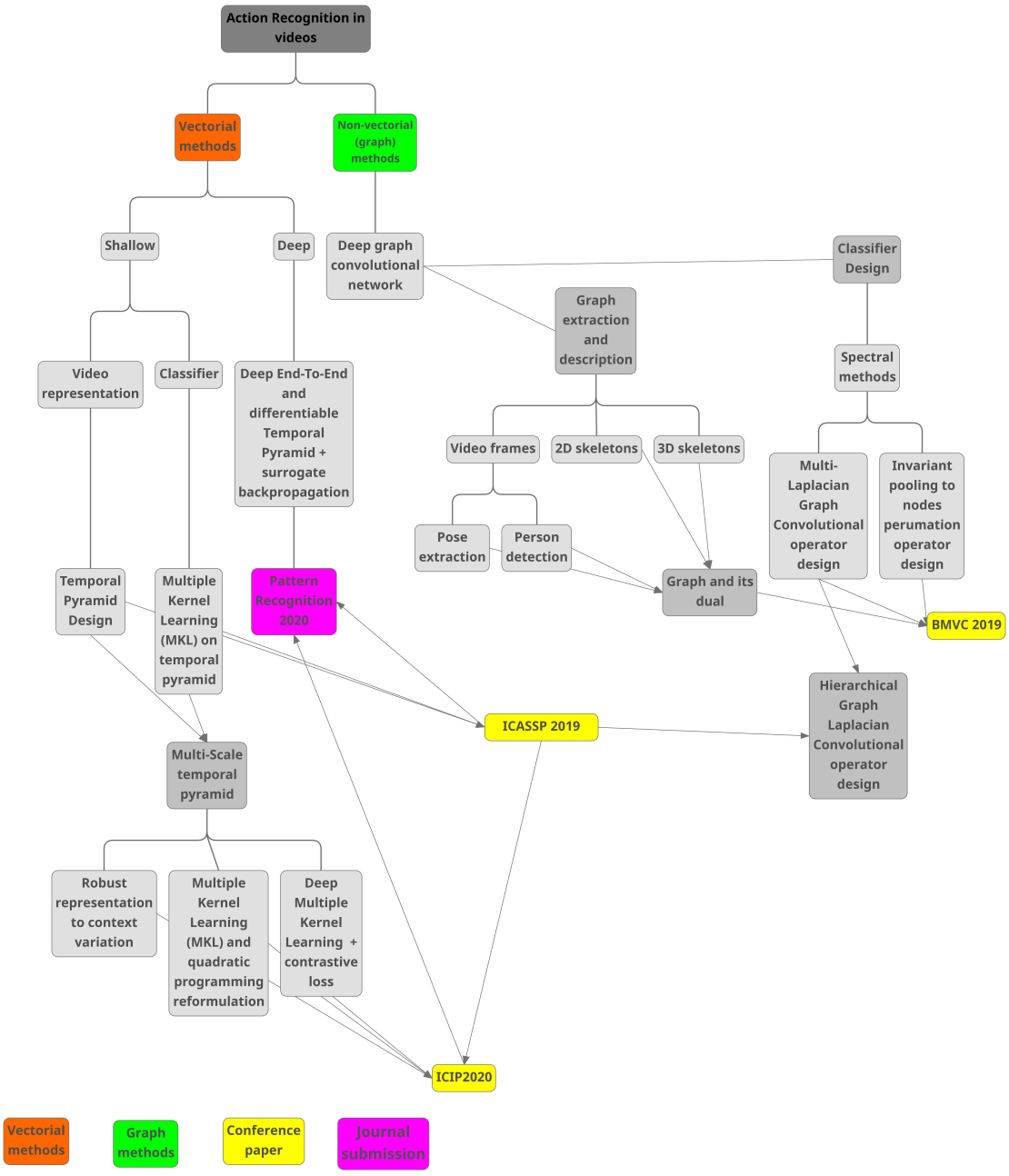


FIGURE 1.11 – A map of our contributions and their relationship.

CHAPITRE 2

ACTION RECOGNITION STATE-OF-THE-ART

Contents

2.1 Historical Notes on Human Actions Understanding

Studying and analyzing human actions dates back to the 15^{th} century in the field of Arts where artistic representations were motivated by human representations. The first works come back to *Leoardo Da Vinci*, since his early age, he had been studying human proportions and measurements as well as human anatomy

for the purpose of improving his art.
This motivation emanates from the importance of understanding human and animal insides to depict them correctly. He wrote in his *On Painting* : "it is indispensable for a painter, to become totally familiar with the anatomy of nerves, bones, muscles, and sinews, such that he understands for their various motions and stresses, which sinews or which muscle causes a particular motion.".
He added :"The space between the mouth and the base of the nose is one-seventh of the face. The space from the mouth to the bottom of the chin is one-fourth of the face and equal to the width of the mouth. The space from the chin to the base of the nose is one-third of the face and equal to the length of the nose and to the forehead.".

Two centuries later, bio-mechanics have emerged to study the different structures, functions and motions of biological systems from mechanics standpoint. One of the works include physiological study of movements by applying analytical and geometrical models initiated by *Galileo Galilei*. Following his rigorous studies, he concluded that bones serve as levers and muscles function, corroborated by mathematical principles.

Later on, 19^{th} century is characterized by the emergence of cinematography where *Eadweard Muybridge* invented a machine for displaying a recorded series of images. He is considered as a pioneer of motion images and techniques for studying human motion.

The birth of computers was a key moment in the history of human motion representation and understanding. Five decades ago, *Gunnar Johansson* initiated the first study of motion perception using a sequence of images to analyze an elementary programmed human motion. As a consequence, his studies inspired many works in computer vision later on for human perception understanding.
A historical overview of human action understanding is summarized in Figure 2.1.

Computer vision based motion techniques aim at providing a deep understanding and representation of motion automatically from a sequence of images. The process is achievable through intelligent machines able to think and reason. This machine was conceived by *Alan Turing* who is considered as the father founder [1] of AI. It is initially designed to solve the fundamental problem of *decidability* in

1. The term *Artificial Intelligence* has not been chosen at random since it happened that *John McCarthy* also would come up with this concept in 1956, two years after *Alan Turing's* death. Moreover, few centuries before, the term of *Artificial Intelligence* has already been thought in 1315 by *Ramon Llull* who cogitated about the idea of qualitative and quantitative reasoning could be artificially implemented in machines.

arithmetic which had a profound influence over several emerging fields of AI. These fields include fuzzy logic, genetic programming and kernel methods, as well as neural networks.

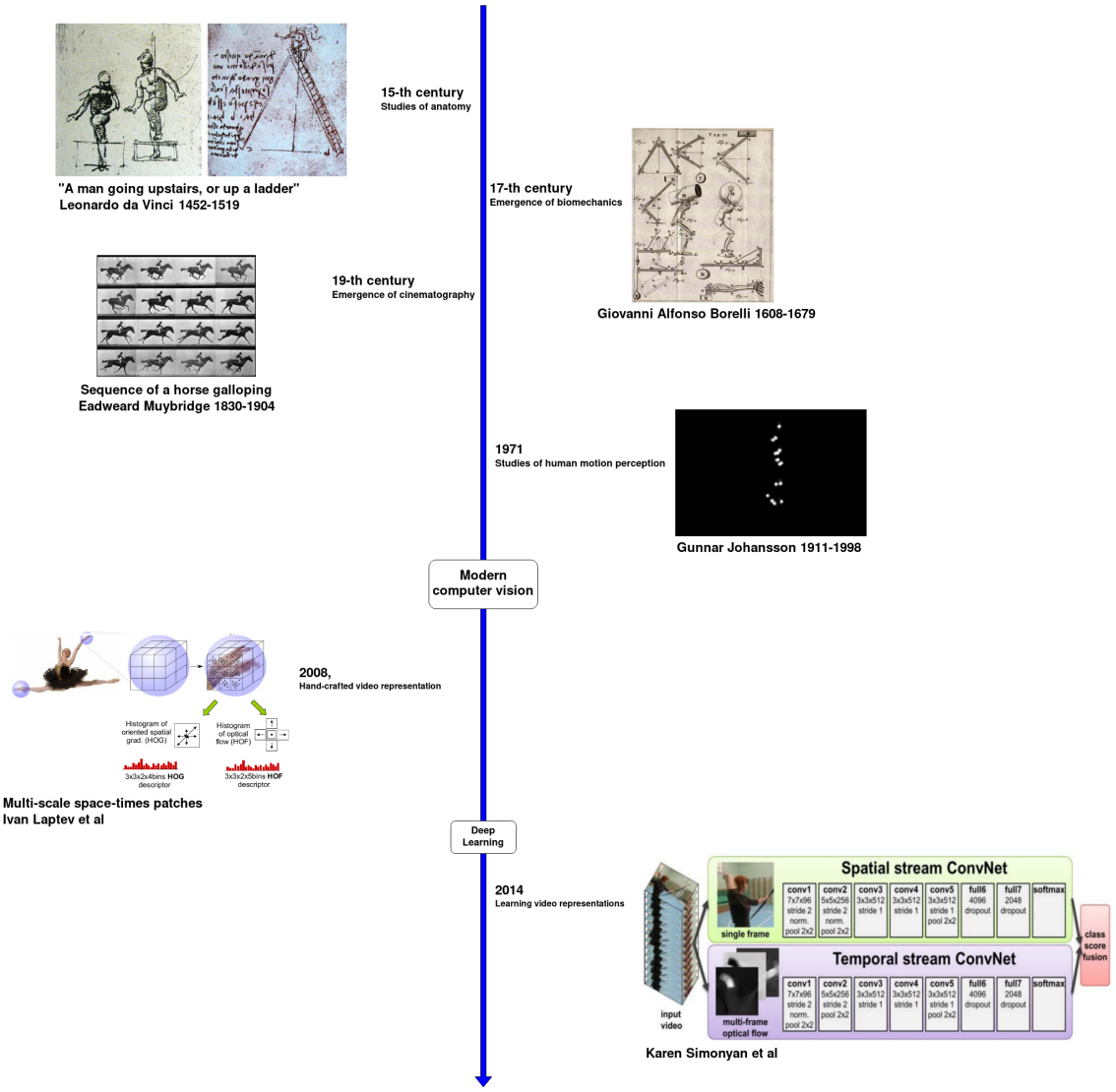


FIGURE 2.1 – Historical overview of human action understanding. This scheme focuses on the important phases that have marked the history of evolution of scientific thoughts for human motion understanding, from early centuries until the emergence of computer vision and the deep learning revolution. This figure is inspired by the talk of [105].

2.2 Overview on Modern Computer Vision Models for Action Recognition

Despite the fact that deep neural networks have shown astonishing results and outperform standard ML models on a wide spectrum of applications, until recently it was still ambiguous for video analysis, whether general ML or DL models are better for action representation and classification or their combination.
This ambiguity is due to the following reasons : (i) video classification is achieved by applying directly the DL models targeted to image classification by adapting their inputs format to video data [2] [15] as illustrated in Figure 2.15, initially designed for image data. Besides, (ii) video data are spatio-temporal, encoding appearance and motion information while images are static and describe only the appearance of objects. As a consequence, the convolutional and pooling operators of DL models don't encode naturally the temporal aspects of data.
Moreover, the motion representation of video data is not clear. One may argue that the sequence of frames describing video action is sufficient to achieve action classification. However, building motion modality from RGB frames could be helpful to encode rich temporal information while being complementary to spatial information to provide discriminant representation prior to their classification. Motion modality can be obtained by processing the sequences of frames via the well known *optical flow* algorithm [106-109].

In the following sections, we present a review of existing approaches that tackle the problem of human action recognition from visual data. Various methods, including *hand-crafted* detectors/descriptors, *learned representations* and *hybrid representations* as illustrated in Figure 2.2, as well as the classifiers that go along with, namely shallow (standard machine learning) and the recent deep learning classifiers which can be divided into two families, *vectorial* and *non-vectorial (graph-based)* as shown in Figure 2.3.

Different approaches are discussed in the subsequent sections; some of them are closely related to our contribution while others not. However, the latter provide an important background about current challenges and solutions designed up to now in the literature, as well as insights for ongoing research axes for a better understanding of the field. Our work is by essence focused on deep learning methods. The general scheme of action recognition representations approaches

2. Appearance inputs based on rgb frames and motion inputs based on the optical flow components of rgb frames. The former models the spatial information while the latter the temporal one.

that we describe in subsequent sections are summarized in Figure 2.4

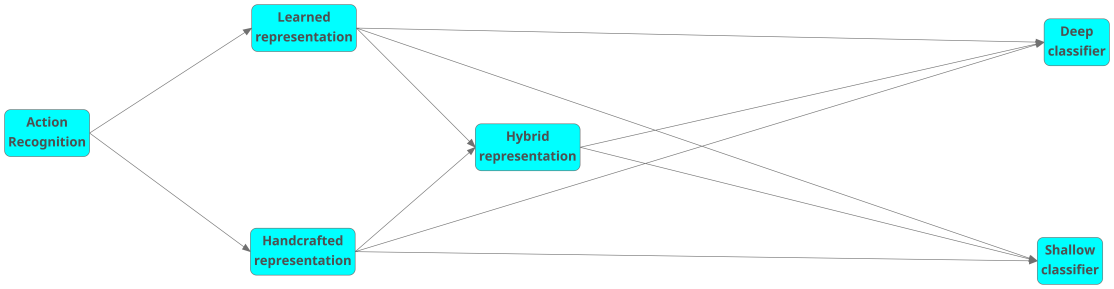


FIGURE 2.2 – Scheme of different methods for action recognition.

2.3 Handcrafted Video Representations

Hand-crafted representation design for action recognition has been an active research and well studied during at least two decades before the resurgence of neural networks. Handcrafted representations have achieved good performances and hence were the standard approaches for different classification problems. These approaches consist at extracting local statistics, referred to as descriptors, for both appearance and motion information from raw video frames, followed by their combination prior to classifying them with ML algorithms such as SVMs. It includes three main techniques, namely *space-time*, *fuzzy logic* and *human pose contours*.

2.3.1 Space-time Approach

An action contains several visual characteristics which describe its appearance and motion information. The former can be color, edge histogram while the later relies on motion history, conveying information about the temporal structure of video action. This approach is composed of four ingredients respectively : feature detector based on Space Time Interest Point (STIP) [110], feature descriptor, feature aggregator and classifier.

STIP detector can be **dense** such as *V-Fast* [111], *Hessian detector* [112] or **sparse** such as *cuboid detector* [113], *Harris 3D* [114] and Spatial-Temporal Implicit Shape Model (STISM) [115]. It aims at detecting interest points for each video segment w.r.t its spatio-temporal structure followed by their representation relying on **local**

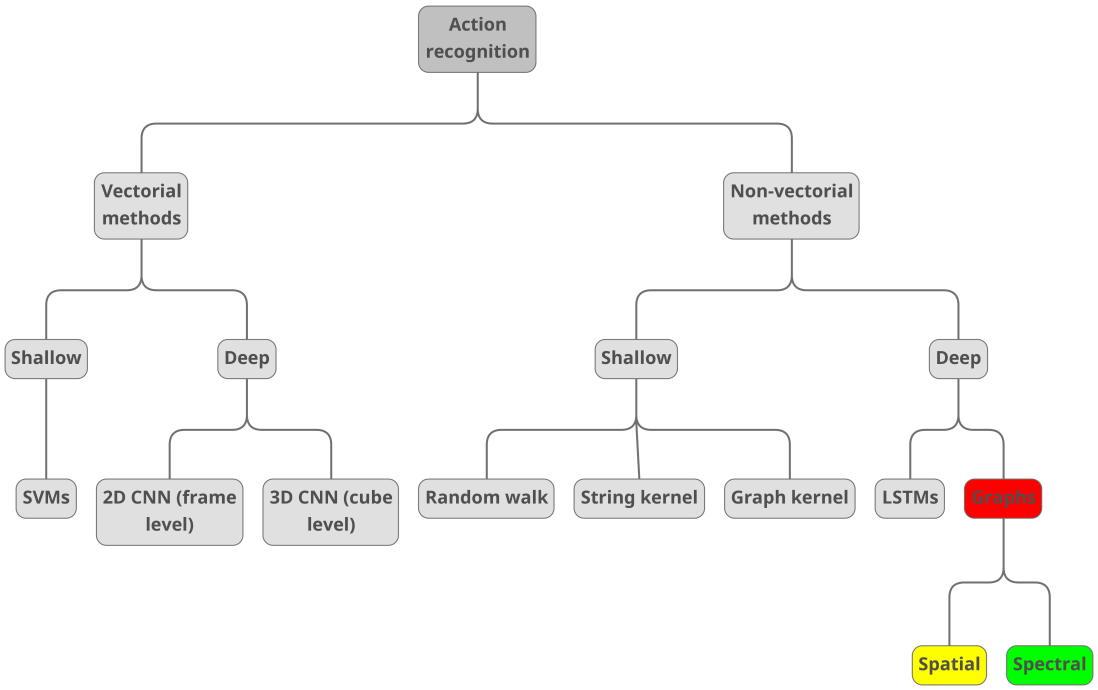


FIGURE 2.3 – This figure shows the different families of methods used in the literature to tackle the problem of action recognition. The red cell associated to graph methods represent the least investigated approaches for this particular task. Recently, few works based on graph methods have emerged to achieve action recognition relying on 2D/3D skeletons features (already provided). However, action recognition with graph methods operating on sequences of rgb frames has comparatively been less investigated and constitutes one of our contributions in this thesis. Moreover, spatial graph techniques shown in yellow cell are the most commonly used methods and relatively well explored (in general and in the particular task of action recognition) compared to spectral ones in green cell.

descriptors such as Enhanced Speeded-Up Robust Features (ESURF) [116], *N-jet* [117] or **global descriptors** such as Histogram of Oriented Gradients (HOG) [118] and Histogram of Optical Flow (HOF) [3] [119], as well as Motion Boundary Histogram (MBH) [119].

HOG encodes information related to appearance while HOF and MBH provide information about velocity and speed.

3. See modern computer vision approach (2008) based on hand-crafted video representations (HOG and HOF) in Figure 2.1.

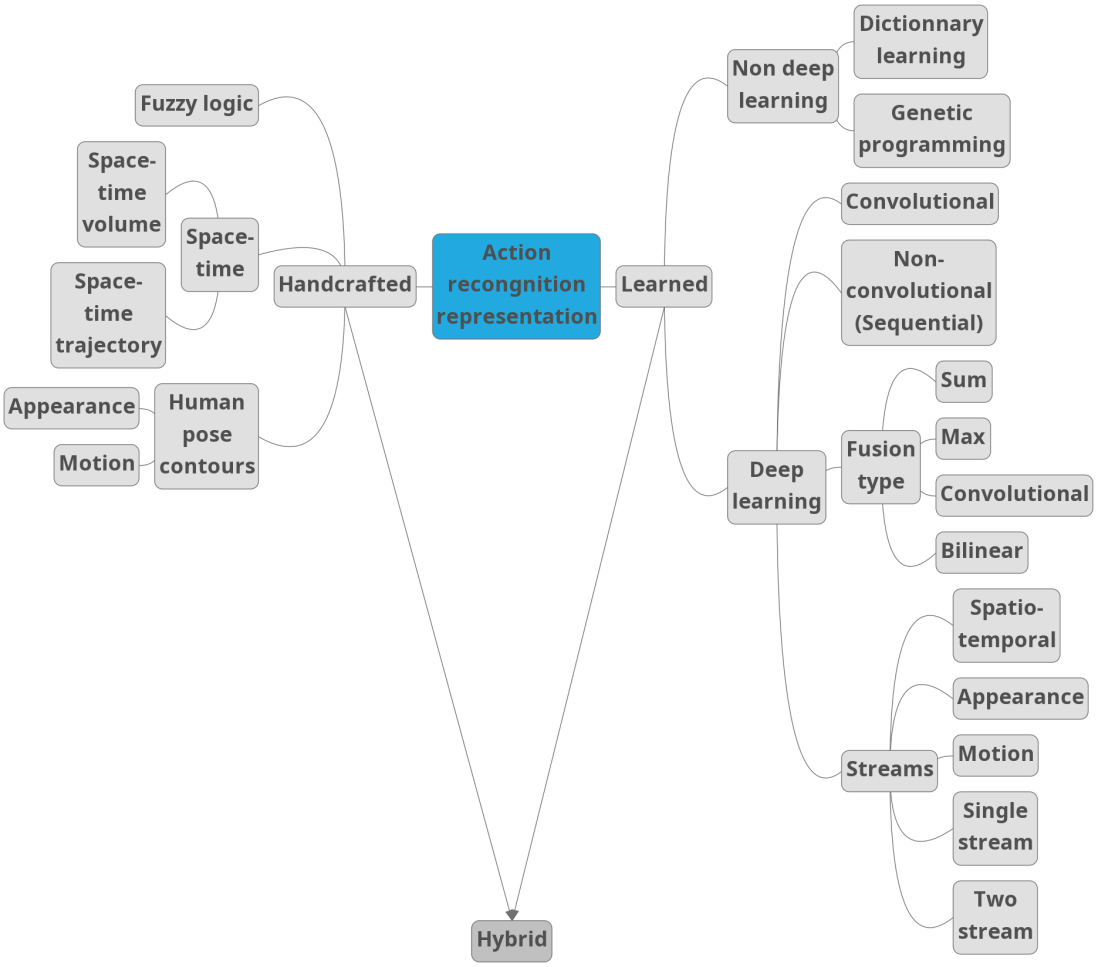


FIGURE 2.4 – This figure gives a general overview on the different approaches for video action representation, including handcrafted, learning and hybrid approaches.

The aggregator then combine[4] the resulted STIP features relying on models based on bag of words[5] such as bag of visual words [121], Fisher vector[6] [124, 125], improved dense trajectories[7] [126] or probabilistic models such as hidden

4. This step of aggregation is quite challenging since the semantic representations are heterogeneous. As a consequence, a simple combinations of histograms leads to poor classification.

5. It is introduced for textual contents and retrieval [120] then extended to computer vision tasks. The general principle consists at representing a documents with a set of words and their respective frequencies.

6. It is an encoding method inspired by Fisher kernel [122]. It relies on the assumption that local descriptors can be modeled by a probability density function. Common FV methods use concatenation or Spatial Pyramid Matching (SPM) [123] to encode coarse spatial information which can be detrimental to fine-grained actions.

7. Method designed to estimate camera motion and to improve dense trajectories. It aims at building a robust homography estimation in order to remove inconsistent matches in human trajectories.

Markov models, dynamic Bayesian action network [127], Gaussian mixture models followed by their classification using SVMs.
For instance, Fisher vector models the distribution of representations over the datastet using a probabilistic model to build features by keeping first and second order statistics.

Space-time approaches can be divided also into two families : **Space-time volume** and **Space-time trajectory**.

Space-time volume. Video representation in space-time domain can be seen as a volume that is represented in 3D spatio-temporal cuboids[8]. The latter are compared using a similarity measure which aims at computing the degree of their correspondence. Later on, space-time volume has been extended to multiple modalities, including motion energy image and motion history image [128] combined with HOG prior to their classification with multi-instance SVMs.

Space-time trajectory. Video action can be described by its 2D/3D spatio-temporal trajectories [129] given human joints. The trajectories are constructed relying on the displacement field of joints obtained with optical flow.
In order to better estimate camera motion, dense optical flow is combined with enhanced speeded-up robust features [130]. However, it is computationally expensive to build dense trajectories. For that reason, a sparse trajectory representation is introduced. It is called saliency map [131]. It allows to avoid the expensive computational cost of dense features by discarding several features without compromising the discrimination power of the resulted representations. Saliency map feature can be combined with HOG, HOF and MBH to build multi-source feature representations.

Despite the advantages that space-time approach offer, mainly its invariance to action speed, it remains challenging to track the joints and modeling multiple persons in the scenes.

2.3.2 Fuzzy logic Approach

A few works based on Fuzzy Logic (FL) systems (sub-domain of AI) attempt also to achieve action recognition. Fuzzy logic systems are of particular interest for this task, especially for real world applications which are characterized by complex scenes including multi-view variation, camera calibration and uncertainty hand-

8. Which can be sparse or dense.

ling. In order to tackle these difficulties, fuzzy logic could be the suitable choice. Its general principal consists at defining fuzzy sets of some parameters w.r.t the task to solve and its environment. The rules of fuzzy logic system are designed by an expert which is beneficial to build understandable and interpretable models.

[132] propose a fuzzy logic model to build human action representations based on log-polar histograms and on temporal self-similarites before their classification with SVM. Its inputs include human joints and the velocity of human actions. Similar to [132], in [133] a fuzzy logic model based on C-means clustering is proposed. It first extracts visual features such as human joints and their neighbors, as well as their speed followed by a fuzzy C-means clustering procedure which aims at learning different possibilities membership functions for these features.

Most of existing fuzzy logic models for action recognition [132, 133] provide view dependent representations and hence are limited to recognize actions from fixed view. However, complex action recognition applications require to recognize human action from any viewpoint. One may consider to setup different cameras and calibrate them which is a quite challenging solution when it comes to cope with real time and real world scenarios. As an extension to [132, 133], authors in [134] design a view invariant fuzzy logic model using only a single camera. It proceeds in four steps : human poses contours are extracted relying on qualitative Poisson human representation ; followed by the estimation of their views which are then clustered before their classification.

[135] propose a neuro-fuzzy model to build view invariant human action representation. This model results from the combination of a biologically inspired model and fuzzy logic model. This biologically inspired model detects motion features and optimizes them using a quantum particle swarm optimization procedure, initialized with centroidal voronoi tessellations followed by a fuzzy inference step which models feature clusters as Gaussian membership functions.

2.3.3 Human Pose Contours-based Approaches

Human pose is composed of a collection of joints. These joints can be described with their 2D/3D coordinates and with their surrounding joints w.r.t the human body connectivity to form a cylinder, ellipsoid or skeleton based surface mesh [9]. In this approach, action representations can be built relying on the appearance,

9. Known also as visual hulls. It is a geometric representation based on silhouette shape. Visual hull construction relies on the assumption that the foreground of an object can be separated by the background. It results into foreground and background binary image.

motion of human pose or their combination.

Appearance features are represented with the human joints and their contours [136]. First, foreground joints are estimated from frames using segmentation techniques [137] to detect contour points [138]. Then, each joint is described with the foreground features of the image w.r.t its spatial location and with its neighboring regions to enrich its context [138, 139]. [140] propose to extract scale-invariant features from the contours of human pose which are then put into clusters to build key poses. [141] suggest to divide a frame into a fixed number of cells and grids to build mutli-scale pose representations of joints, followed by their clustering before their classification.

Motion features are based on optical flow or its variants [142] such as motion history image [143], motion histogram volume [144] or histogram of motion intensity representations of joints. [145] use histogram of motion intensity to build multi-view human motion representation.

Appearance and motion features can be fused in 3D volume such as cuboid to build a video level representation which is view invariant [144, 146, 147]. Particularly [148] propose to represent human action as sequences of spatio-temporal human poses, using a distance measure for matching joints.

2.4 Learning Methods

In the previous section, we described action representation approaches based on handcrafted detectors and descriptors. Another family of approaches consists in learning action representation automatically from raw videos in a partial or complete end-to-end process [10].

We distinguish three learning approaches : (i) Evolutionary strategy, (ii) Dictionary learning and (iii) Deep learning

2.4.1 Evolutionary Strategy

Evolutionary strategy is bio-inspired optimization technique based on the natural evolution of biological populations.

10. End-to end process means that representations are learned and classified jointly and automatically directly from raw data

Genetic programming is a class of evolutionary strategy used mainly in data modeling, feature selection and as black-box optimization [149]. Its principle consists in optimizing a system without any prior knowledge on its results.

In action recognition, it is used to learn spatio-temporal motion feature representations [125, 150]. Given a sequence of frames and their associated optical flow components, a group of 3D operators such as Gabor filter and wavelet are combined to build data-dependent descriptors. These descriptors keep evolving over the group of 3D operators until meaningful representations are built and hence action recognition accuracy is maximized through an appropriate fitness function [11]. An example of the whole process is illustrated in Figure 2.5.

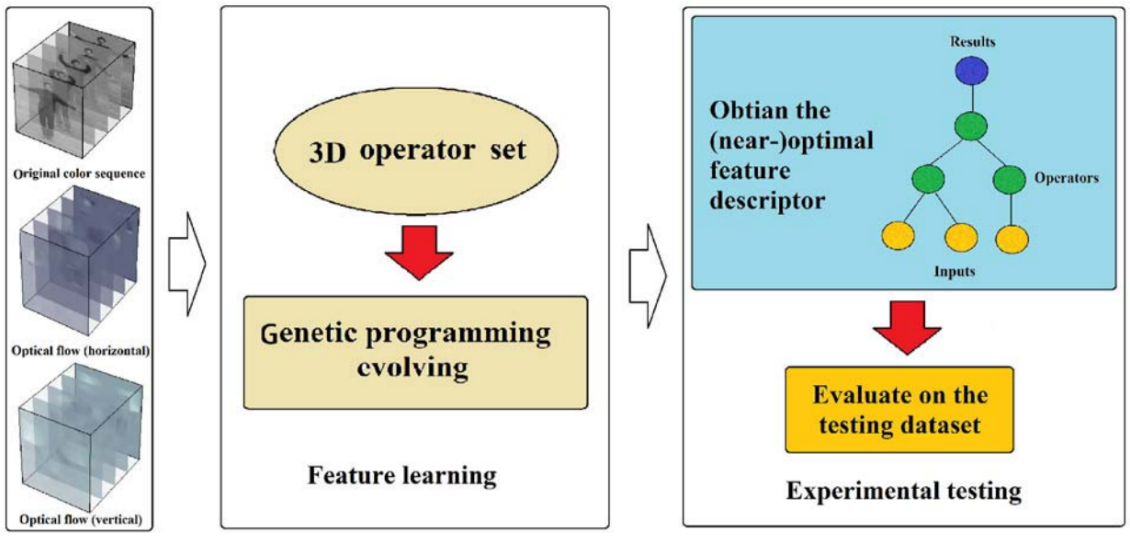


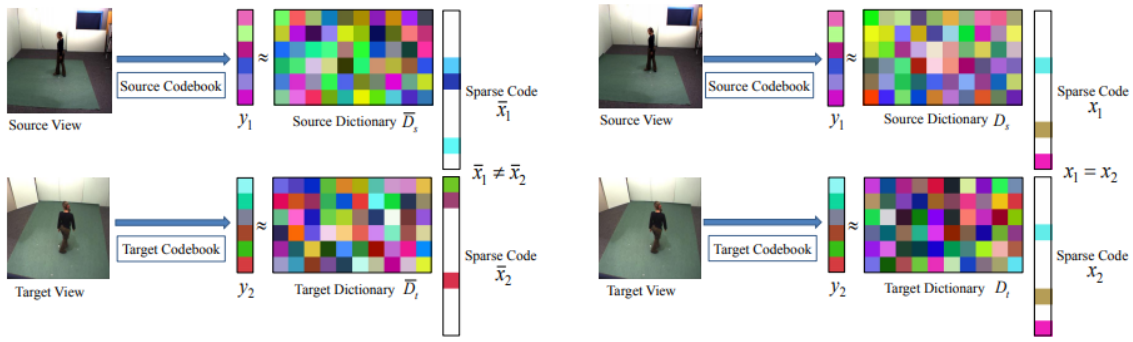
FIGURE 2.5 – Learning video action recognition with genetic programming. Genetic program is represented as tree structure of three components : selection, crossover and mutation which aim at selecting best performing spatio-temporal descriptors from a set of evolving candidates through generations. Picture credit for [125].

2.4.2 Dictionary Learning

Dictionary learning is a signal processing technique which aims at building sparse representation of data. In the context of action recognition, dictionary learning is used to learn sparse descriptors from spatio-temporal action representation, similar to BoVW methods. It can be divided into three methods. (i) Over

11. Fitness function is an objective function which takes candidate solution as input to the problem to solve and produces as output the fitting of the candidate solution.

complete dictionary basis [151] which is a linear combination of small dictionaries built upon spatio-temporal features. (ii) Dictionary based on hierarchical features [152] and (iii) transferable cross-view dictionaries [153] as illustrated in Figure 2.6 which provides invariant multi-view action representation. While (i), (ii) and (iii) are supervised models, [154] propose unsupervised dictionary learning variant relying on locality constrained linear coding [155] and on trajectories of features.



(a) Independent dictionary learning (b) Transferable dictionary learning

FIGURE 2.6 – Cross-view transferable dictionary to build invariant multi-view action representation. (a) Based on Bag of Visual Words (BoVW) where the source and target dictionaries are learned individually from two videos with different views but belonging to the same action. (b) Based also on BoVW but the source and target dictionaries are learned simultaneously. Picture credit for [153].

2.4.3 Resurgence of Neural Networks

Hand-crafted descriptors are data-dependent, their performances vary from one task to another following the intrinsic and extrinsic acquisition conditions of data. However, there is no universal hand-crafted descriptor. For that reason, learning descriptors from raw data may be more advantageous.

In this spirit, artificial neural networks approach has been introduced, starting from the formal neuron [156] and perceptron [157]. Artificial neural networks aim at learning representations directly from raw data and classifying them jointly in an end-to-end and differentiable manner. They are composed of a set of neurons interconnected to build a hierarchical multi-layered structure.
One of the most commonly used artificial neural network for the task of classification is feedforward network, which is a directed acyclic graph that maps inputs to outputs. Artificial neural networks are learned by using Stochastic Gradient

Descent (SGD) to minimize the loss function described in Equation 1.3, with gradients computed by back-propagation [12, 158].

Multi-Layer Perceptron (MLP) is one of the artificial neural networks composed of blocks of perceptron layers and non linear activation functions (eg. Sigmoid). It is modeled as complete bi-partite [12] directed acyclic graph, which takes input data (image) and outputs its label as depicted in Figure 2.7.

The key success of artificial neural networks resides in their ability to approximate any continuous function with enough neurons [159]. Deep neural networks should contain sufficient number of layers and enough neurons to learn expressive inputs-outputs mappings effectively. They learn hierarchical representations from raw data with an increasing level of abstraction before classifying them.

In particular ConvNets have outperformed state-of-art handcrafted methods in the task of image classification with a large margin (see Figure 1.6). The strength

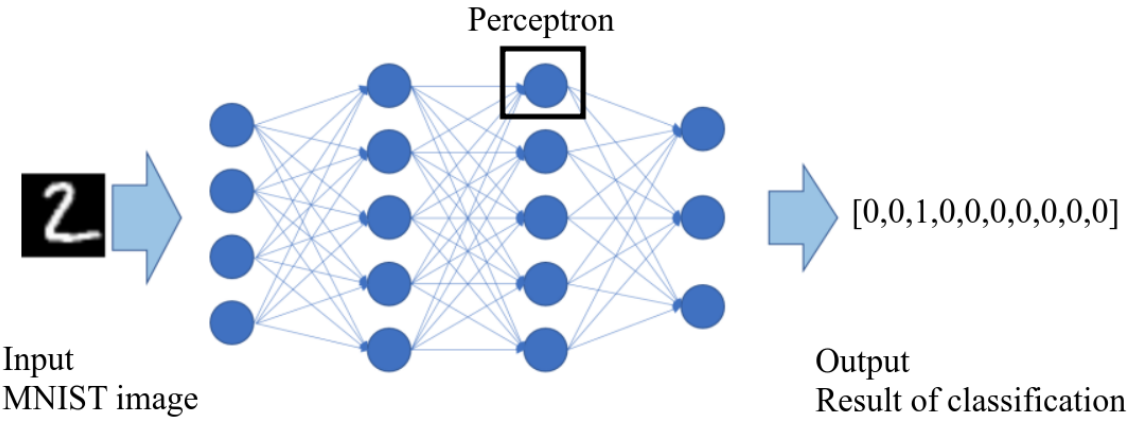


FIGURE 2.7 – Multi layer perceptron for handwritten digits classification. Given Digits ∈ [0,9], multi layer perceptron learns representations for each digit and outputs result of classification in a vector of 10 values. Picture credit for [160].

of ConvNets [11, 14, 158] resides in their ability to provide discriminant and stable feature representation for classification. Moreover, they are able to learn complex structures while being invariant to translations and rotations, as well as stability to small deformation [91]. Figure 2.8 shows examples of translated, rotated and deformed handwritten digits images that ConvNets are able to classify correctly.

12. Given two sets of vertices where every vertex of the first set is connected to vertices of the second set.

Common ConvNets architectures are built by stacking multiple blocks composed of convolutional operators whose filters are learned with backpropagation, a non-linear activation function such as Rectified Linear Unit (ReLU) and a pooling operator used to reduce the dimentionality of the resulting representations and to build local invariance [13] to transformations at different image locations.

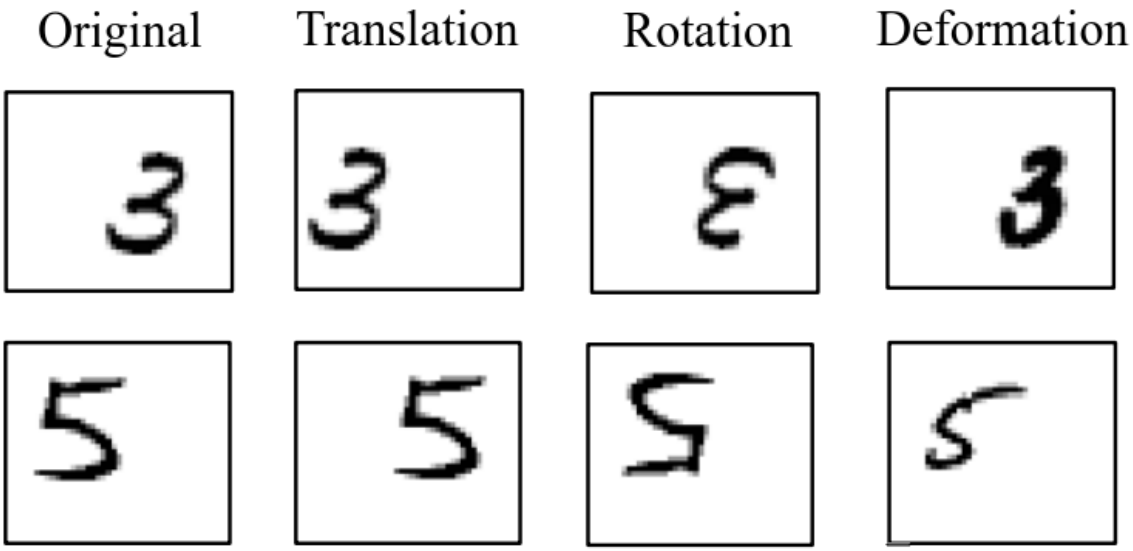


FIGURE 2.8 – This figure shows the different variations in representing a digit, including transformation such as translation, rotation and deformation. A classifier should be invariant to translation, to rotation and to relatively small deformation in order to classify them correclty.

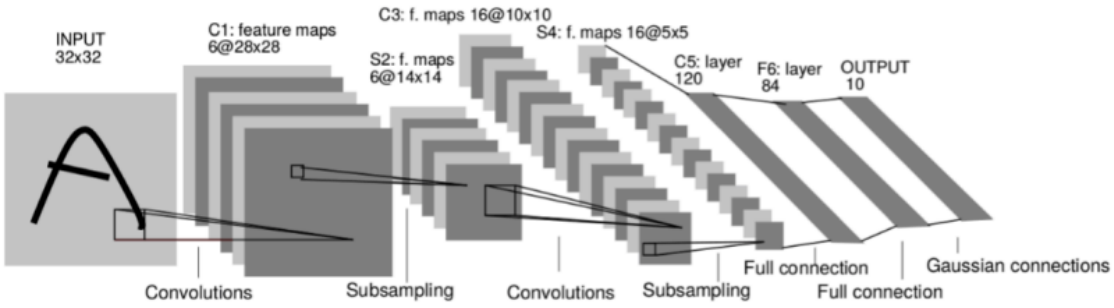


FIGURE 2.9 – Architecture of *LeNet*. A convolutional neural network for handwritten character recognition. Picture credit for [13].

The initial convolutional neural network architecture was shallow, it is composed of five layers, called *LeNet* [13] which is initially designed for document

13. This process is similar to Spatial Pyramid Matching (SPM) which achieves local invariance to transformations in the whole image.

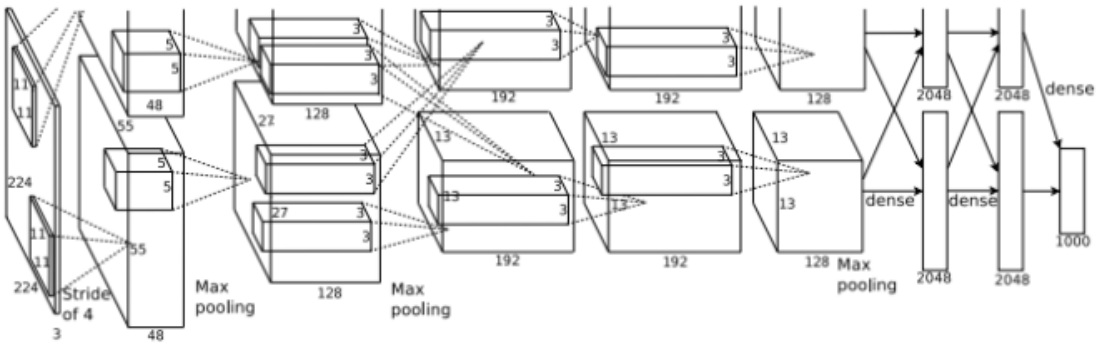


FIGURE 2.10 – This figure illustrates *AlexNet* architecture which won ILSVRC 2012 by outperforming all handcrafted methods on *ImageNet* dataset and hence initiating the DL revolution. Picture credit for [14].

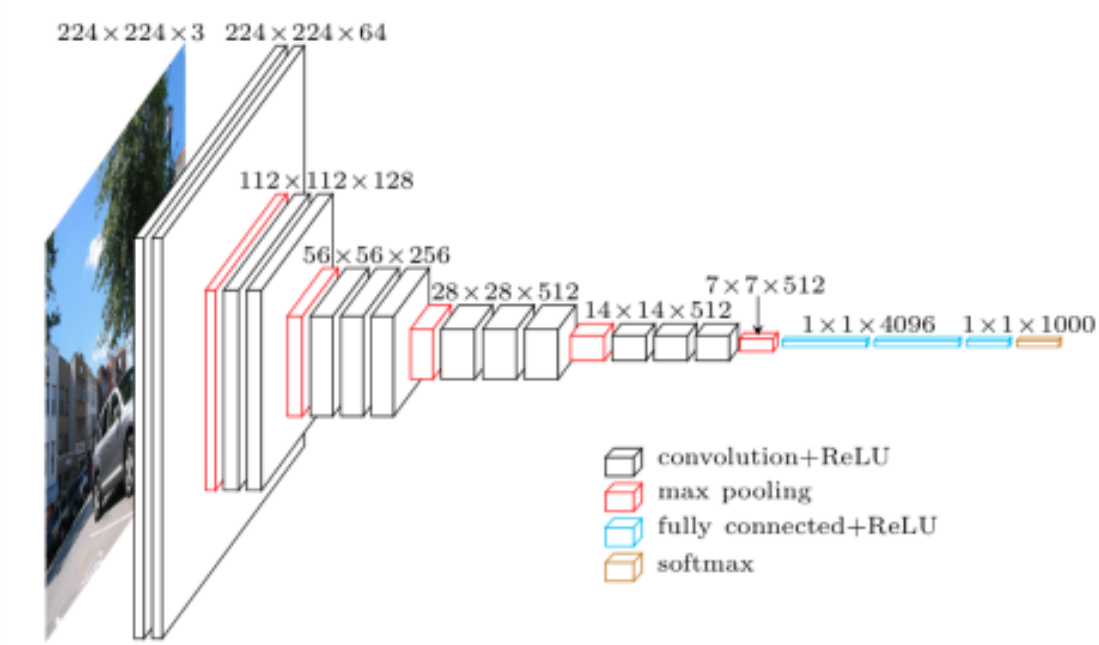


FIGURE 2.11 – *VGG-16*, an extension of *AlexNet* with deeper layers. Picture credit for [161]

recognition. It is depicted in Figure 2.9. However, the first successful deep ConvNet was *AlexNet*, composed of eight layers, appeared 14 years later thanks to the availability of large-scale datasets, namely *ImageNet* and efficient computational GPUs resources to train them in reasonable time. The *AlexNet* architecture is displayed in Figure 2.10.

It won ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012 with a large margin w.r.t state-of-the-art followed then by its extension to *ZF-Net* (the winner of ILSVRC 2013) [163], and *CNN-F/M/S* [164]. Later on, improved and dee-

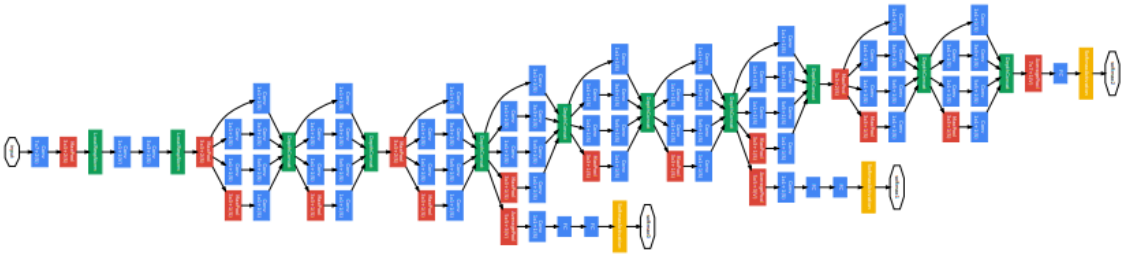


FIGURE 2.12 – *Inception V1* architecture, well known as GoogLeNet composed of 22 layers. It obtained state-of-the-art for ImageNet classification in ILSVRC 2014. Picture credit for [162].

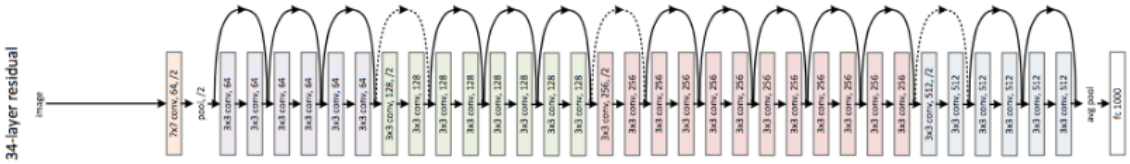


FIGURE 2.13 – This figure displays ResNet-34, a deep residual network. These residual connections ease the training process and allow to define deeper networks while avoiding vanishing/exploiding gradients problem. Its extension ResNet-152 won ILSVRC 2015. Picture credit for [16].

per networks have been proposed, called *VGG*-11/16/19 [14] [161]. *VGG*-16 is illustrated in Figure 2.11.

[161] Study the effect of network depth on the performance of classification. They showed that depth is an important property of networks, increasing it leads to clearly better performances. However, this requires a huge number of parameters which comes at a high computational cost. In order to control the complexity of networks and their depth while keeping the computational budget constant, successive *Inception* networks (1, 2, 3 and 4) were designed [104, 162]. The first version of *Inception* won ILSVRC 2014. It is illustrated in Figure 2.12. However, increasing depth of networks without a careful design can lead to optimization issues, resuling into vanishing/explosion of gradients [53]. To circumvent that, several networks have been designed to facilitate backpropagation including *ResNet*-18/34/50/101/152 [16]. *ResNet*-34 is displayed in Figure 2.13. The *ResNet* version with 152 layers won ILSVRC 2015 with a large margin as depicted in Figure 1.6. Table 2.1 summarizes the statistics of ConvNets including depth and the number of parameters.

14. 11,16,19 stand for the depth of the network

Network	Depth	#Parameters
LeNet [13]	5	60000
Alexnet [14]	8	61M
VGG-11 [161]	11	132M
VGG-16	16	138M
VGG-19	19	143M
Inception *V1* [104]	22	6M
Resnet-50 [16]	50	25M
Resnet-101	101	44M
Resnet-152	152	60M

TABLE 2.1 – Deep learning models complexity in depth and number of parameters. M stands for million.

Later on, several variants have been introduced to reduce the complexity of networks and their computational time, including DenseNet [165], Multi-connection width ConvNets [166, 167], Pyramidal-Net [168], Xception [169], ResNeXt [170] and SqueezeNet [171], as well as MobileNet [172].

After the success of deep ConvNets on image classification, extension to neighboring tasks such as action recognition have drawn a lot of attention. Nevertheless, the progress was not as significant as still image classification as deep ConvNets require huge amount of labeled (training) data which was not the case for action recognition. However, the representation learned on ImageNet [173] have shown their versatility and their transferablility to other tasks while outperforming handcrafted representations [174]. As a result, this makes it possible to generalize deep ConvNets to small datasets in two steps; pretraining them on ImageNet then fine-tuning them on target datasets for the specific tasks while mitigating the risk of overfitting, as achieved in early action recognition solutions based on deep learning [15].

Action recognition in videos. Several action recognition approaches based on deep neural networks are proposed. In the following sections, we discuss the major works proposed in the literature. They can be convolutional or sequence-based, including different streams (appearance and motion) and their type of aggregation : spatio-temporal, two-streams, as well different fusion type of sequences of frames representations : single fusion, early fusion, late fusion and slow fusion. Moreover, video actions can come from different natures such as raw frames and

their optical flow components or 2D/3D skeletons features. Figure 2.4 shows different deep action representation methods.

2.4.4 Convolutional Methods

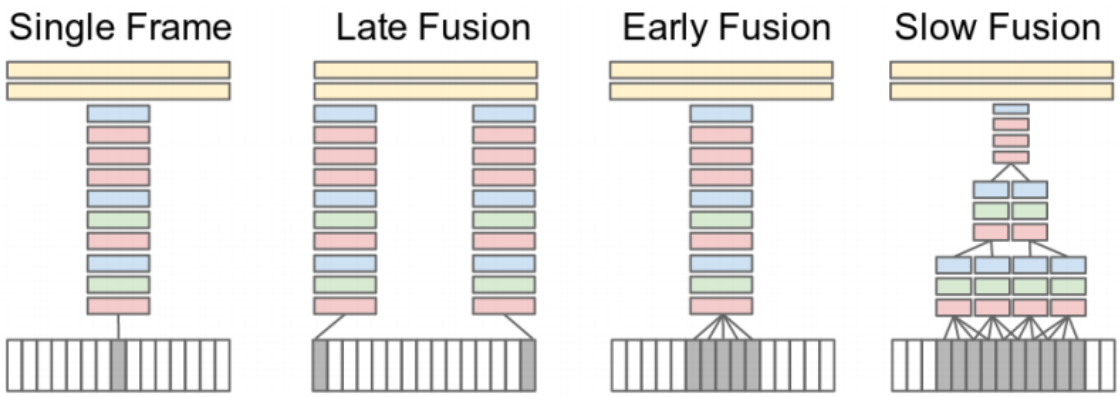


FIGURE 2.14 – Different approaches for fusing frames representations across temporal dimension. Red, green and blue boxes indicate convolutional, normalization and pooling layers respectively. In the Slow Fusion model, the depicted columns correspond to shared parameters. Picture credit for [56].

Videos are considered as 3D volumes including spatial and temporal dimensions. As videos have variable duration, a preprocessing step is needed to build a fixed-size representation to fit the input requirements of ConvNets and this is usually achieved by sampling a fixed number of frames from all the videos [15, 28].

Deep learning methods for action recognition are by essence based on successful ConvNets for image classification. We distinguish two approaches, namely single stream and two stream networks. The former operate on rgb frame modality while the latter on optical flow modality in addition to rgb frames.

2D single stream networks. [56] investigate different ways to fuse temporal information from a sequence of frames. These ways are illustrated in Figure 2.14, namely : single frame, early fusion, late fusion and slow fusion. **Single frame** approach relies on single frames representations which are fused at the last stage. **Early fusion** approach consists at combining in the first layer a set of successive frames representations. **Late fusion** approach uses two networks with shared parameters, their predictions are fused at the last stage. In contrast, **slow fusion**

approach is located between early and late fusion. It fuses frames representations at different stages.
The experimental study conducted by [56] shows that the results are worse compared to handcrafted state-of-the-art features with a large margin. One of the reasons is that the learned representations do not capture motion features. For that reason, two stream network has been introduced to encode explicitly motion information.

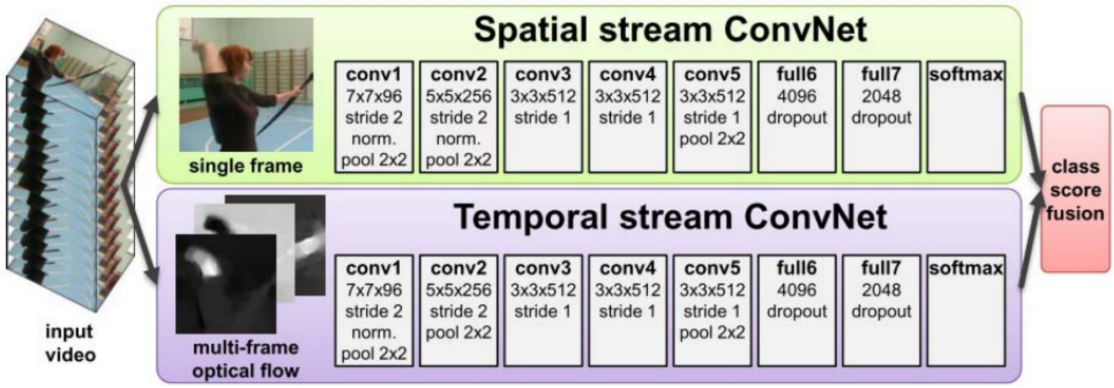


FIGURE 2.15 – This figure shows one of the first ConvNets for action recognition. It is composed of two streams. One for appearance information based on rgb frame input and another one for motion information based on the optical flow components of successive frames. Picture credit for [15].

2D two stream networks. The first successful extension of convolutional neural networks to tackle action recognition task is modeled as a two stream **2D convolutional network** [15] pretrained on ImageNet and then fine-tuned on the targeted task. One stream operates directly on rgb frames to capture static appearance representation, called appearance stream and another one is based on optical flow components of successive frames to describe the dynamic of motion information, called motion stream. The overall classification accuracy is the combination of the two stream scores as it is shown in Figure 2.15. This combination can be based on averaging or SVMs. It is also known as *late fusion* combination.

Averaging is an intuitive and straightforward fusion strategy for combining representations coming from appearance and motion stream, as well as for aggregating frames representations before the classification layer. It has the advantage of not needing any extra parameters to learn while keeping the computational cost reasonable, thanks to independent back-propagation calculations, one for each stream.
Despite its effectiveness and simplicity, it has the drawback of modeling spatial

and temporal information separately. [175] show that their fusion prior to their classification provide more discriminating representation and hence better results than the combination of their scores at the end of training. The scheme of this network is depicted in Figure 2.16. This approach takes the advantage of modeling appearance and motion information explicitly through the two streams and their fusion leads to good features abstraction capturing the spatio-temporal aspects of video. This network is able to learn representations specific to each class while being generic. Moreover, the hierarchy of the network allows to learn progressively invariance to speed motion.

[176] investigate different methods of fusing appearance and motion representations in order to best encode spatio-temporal information [15]. It includes sum fusion, max fusion, concatenation fusion and convolutional fusion, as well as bilinear fusion. **Sum fusion** computes the sum of two feature maps at the same spatial location and channel. **Max fusion** takes the maximum of the two feature maps. **Concatenation fusion** aims at stacking two feature maps at the same spatial location across the feature channels. Based on the resulted concatenated feature maps, **convolutional fusion** convolves them with a collection of trainable filters to reduce the dimensionality of representations and to weight the combination of the two feature maps. **Bilinear fusion** consists at computing a matrix outer product of the two feature maps at pixel level followed by their summation w.r.t spatial location.

Temporal aggregation through the sequence of frames representation has also been studied. While temporal average pooling is the most simple and less expensive operation to aggragate convolutional feature maps over time, two methods are evaluated to take advantage of complex operations that provide efficient feature maps while preserving the temporal structure of videos [176].**3D pooling** applies directly max-pooling on the stacked feature maps. **3D convolution and 3D pooling** convolves feature maps with a collection of filters, followed by a 3D pooling. One of benefit of 3D pooling is its ability to provide invariance to small changes of the features location across the frames.

[177] study different pooling strategies including convolutional pooling, late pooling, slow pooling, local pooling and time-domain convolutional pooling and find out that convolutional based max pooling outperforms the other pooling operators. **Convolutional pooling** aims at performing max-pooling over the last convolutional layer across the sequence of frames. **Late pooling** consists at pas-

15. Note that spatial fusion of networks is a well known problem in different applications and it is not tied to the particular task of action recognition in videos.

sing the last convolutional representations through fully connected layers before applying max-pooling across video frames. **Slow pooling** is a hierarchical combination of frames representations divided into segments. Max-pooling is first applied over the convolutional features of the frames of a given segment followed by a fully connected layer. Then, the last max-pooling layer combines the representations of all the fully connected layers. **Local pooling** is relatively similar to slow pooling. It combines frames representations after the last convolutional layer followed by two fully connected layers. However, it contains only a single layer of max-pooling after the convolutional layers compared to slow pooling. **Time domain convolutional pooling** contains a temporal convolutional layer which captures local relationships between successive frames over temporal window before performing max-pooling in the temporal domain. The different operators are illustrated in Figure 2.17.

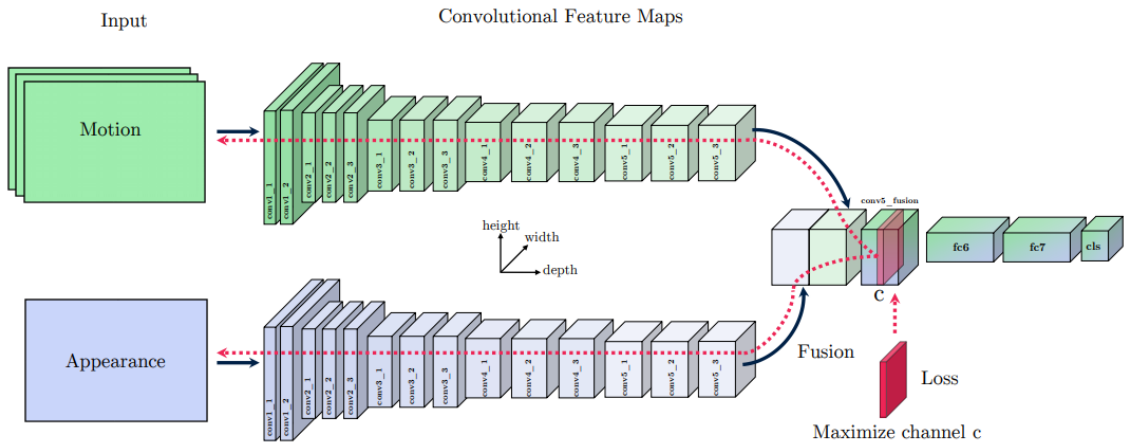


FIGURE 2.16 – This figure displays a two stream convolutional neural network based on the fusion of spatio-temporal representations prior to the first fully connected layers . Activation maximization in red is used to visualize the spatio-temporal convolutional representations. Picture credit for [175].

Handling videos of variable length.The ConvNets designed for action recognition operate on fixed-size inputs. Hence, it requires to sample a fixed-size of frames to fit the input dimensions of ConvNets. Defining a sampling strategy is challenging since the resulted sample should preserve the temporal structure of video. [178] propose ConvNet architecture based on Inception-V1 network [162] (initially designed for image classification. see Figure 2.12) to build long range temporal modeling for videos. It is called temporal segment network composed of appearance and motion stream. They propose a sparse strategy to sample frames across the video instead of random sampling. It consists at dividing the video into several segments of equal duration. From each segment, snippets are sampled

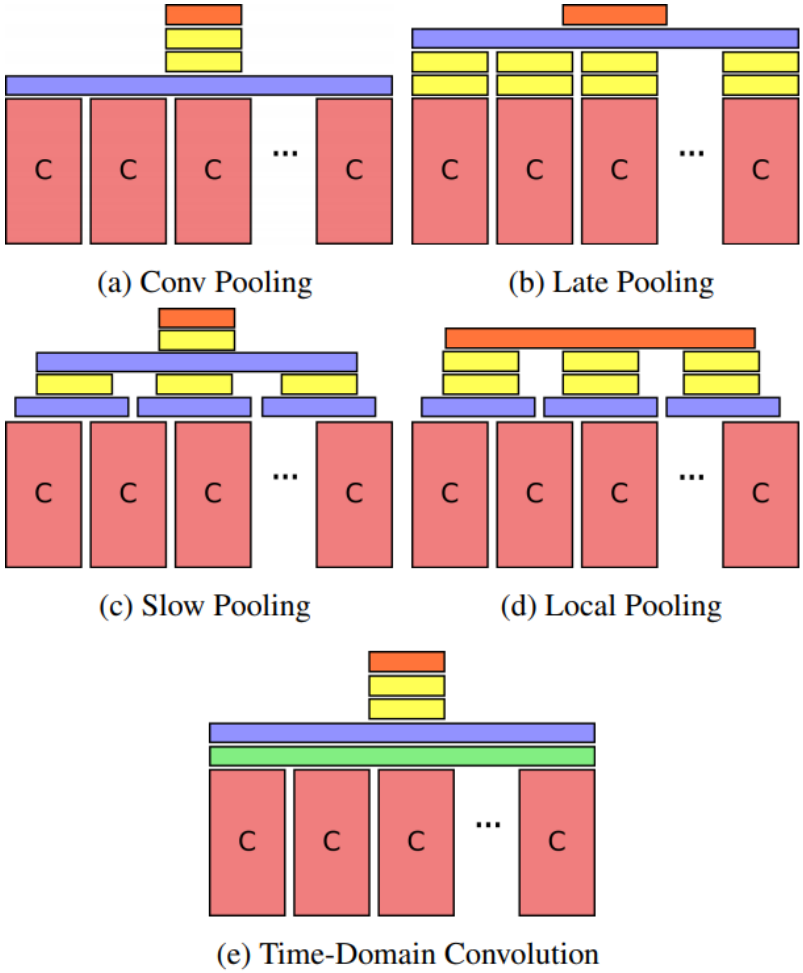


FIGURE 2.17 – This figure displays the different pooling operator architectures. C stands for stacked convolutional layers. Purple, green, yellow and orange rectangles represent max-pooling, time-domain convolutional, fully-connected and softmax layers respectively. Picture credit for [177].

at random. The latter are then fed to appearance and motion stream network. Figure 2.18 displays the temporal segment network.
Moreover, in addition to rgb frames and their optical flow components modalities, rgb difference and warped optical flow modalities have been evaluated, showing their complementary aspects for action classification and hence establishing new state-of-the-art.

Two-stream residual networks. After the success of ResNet-152 on image classification task, its two stream variant has been introduced for the task of action recognition [179, 180] to model rich interaction between the two streams and to provide discriminant local spatiotemporal features. The idea consists at adding

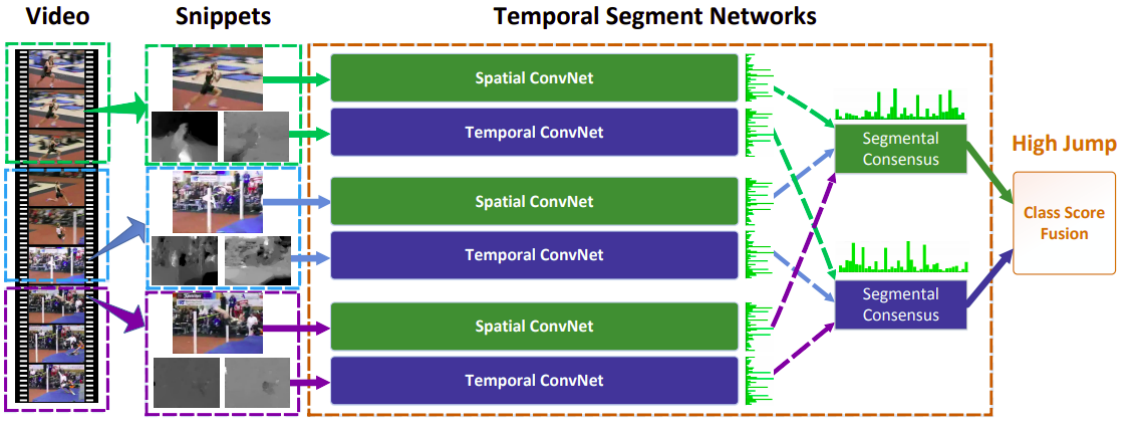


FIGURE 2.18 – Temporal segment network (TSN), each input video is divided into several segments and a short snippet is randomly selected from each segment. The class scores of different snippets are fused by the segmental consensus function to yield segmental consensus, which is a video-level prediction. Predictions from all modalities are then fused to produce the final prediction. Segmental consensus function aims at combining the outputs resulted from multiple snippets to obtain a consensus of class hypothesis among them. Based on this consensus, the probability distribution of action category is predicted for the whole video sequence. Picture credit for [178].

unidirectional residual connections from motion stream to appearance stream only as depicted in Figure 2.19. The reason of the unidirectionality of residual connections is the possible bias of both losses towards appearance information because both streams are initialized with pre-trained *ResNet* weights of *ImageNet* targeted to image classification.

[180] explore different cross stream residual connections to appropriately model the spatiotemporal interactions which is mainly important for discriminating actions of similar appearance and motion patterns such as brushing teeth and applying lipsticks. The different type of motion and appearance streams connections are displayed in Figure 2.20. The ablation study proposed by [180] shows that simple cross residual connections (see (a) in Figure 2.20) between same layers of the two streams decreases classification performance compared to separated two stream network. One of the reason of this decrease in performance could be related to a large change of signal distribution of given layer of one stream after fusing it with signal of other stream. As an alternative, extra additive or multiplicative residual operators are integrated, separated or not with ReLU activation function (see (b), (c) and (d) in Figure 2.20).

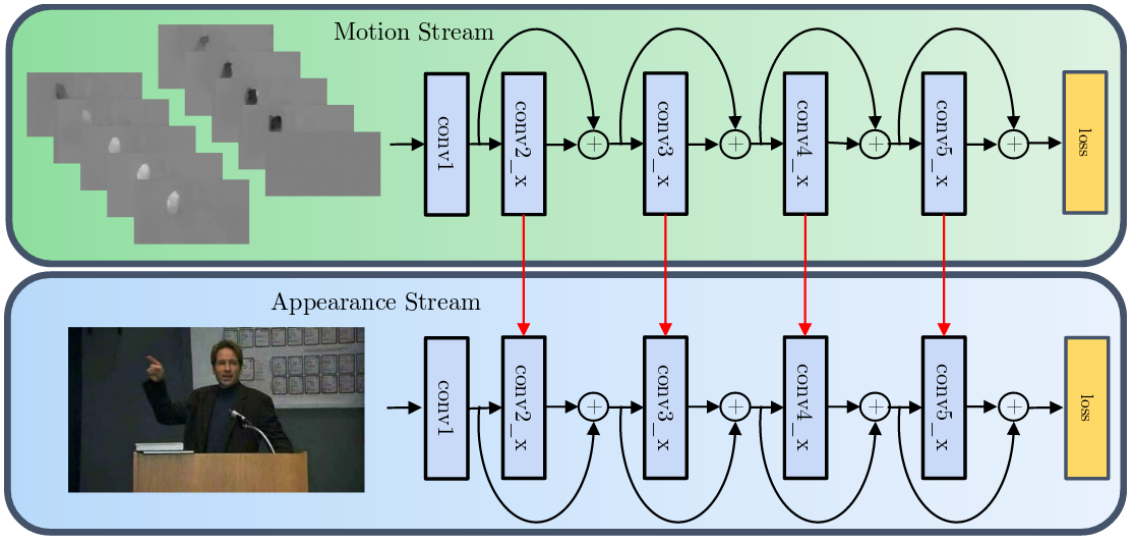


FIGURE 2.19 – Two stream ConvNet with residual connections for action recognition in videos. Picture credit for [179].

It turns out that multiplicative residual connection encodes better spatio-temporal interaction and provides rich and discriminating features compared to additive residual connection, as well as increasing the classification performance. The former better scale the appearance information from motion information and model efficiently the spatio-temporal interaction thanks to the factorization of gradients during the backpropagation with the other stream. The last case (see (e) in Figure 2.20) shows bidirectional residual connections between the two streams. The performance results show that this bidirectionality lead to a loss in classification performance which can be explained by the domination of motion stream by appearance stream since both streams are initialized with *ImageNet* weights.

End-to-end optical flow generation. Motion stream of two stream networks operates on optical flow components of successive frames. These components are computed offline using the traditional optical flow algorithm. This preprocessing step is required to build motion inputs while being computationally expensive and storage demanding. [181] propose an unsupervised ConvNet network, called *MotionNet* to generate optical flow components on-the-fly for a stack of successive multiple frames. Given two successive frames f_1 and f_2, *MotionNet* generates a flow field I. f_2 and I are then used to reconstruct f_1 as $f_1^{'}$ relying on inverse warping which consists at minimizing the difference between f_1 and $f_1^{'}$. Figure 2.21 displays the whole two stream network composed of appearance stream and motion stream. The latter is composed of two blocks *MotionNet* and motion stream respectively.

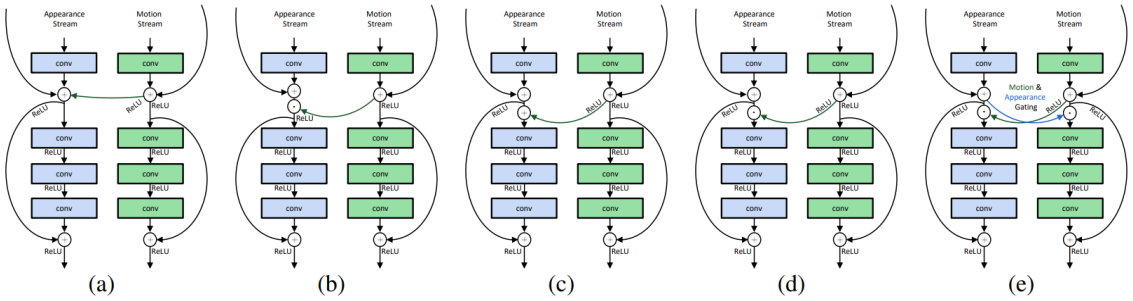


FIGURE 2.20 – This figure shows the different types of interaction created between appearance and motion streams for learning rich spatio-temporal features. In the four first blocks (a), (b), (c) and (d) we observe four different unidirectional connections going from the motion to the appearance stream while in the last block (e), bidirectional gating connections between the two streams are created. Picture credit for [180].

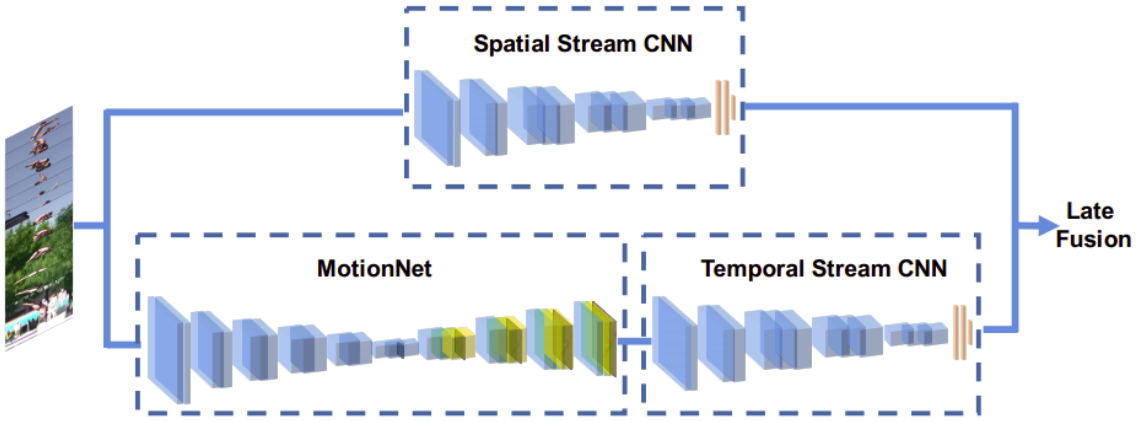


FIGURE 2.21 – This illustration shows the hidden two-stream network. Spatial stream operates on a stack of frames to build appearance representations which are projected to action categories. MotionNet takes consecutive video frames as input and estimates motion in an unsupervised manner followed by temporal (motion) stream that maps the motion information to action categories. Finally, late fusion is performed through the weighted averaging of the prediction scores of the two streams. Picture credit for [181].

3D Convolutional neural networks. Several successful pretrained 2D ConvNets models are used as image feature extractors. The features are mainly extracted from the last fully connected layer, known as classification layer. The latter provides discriminating features ready for classification while being also well suited for transfer learning tasks [182, 183]. However, these features are not directly suitable for videos for the following reasons : i) images are static while videos are

dynamic, ii) deep image feature representations lack of motion modeling and this results into less expressive spatio-temporal features.

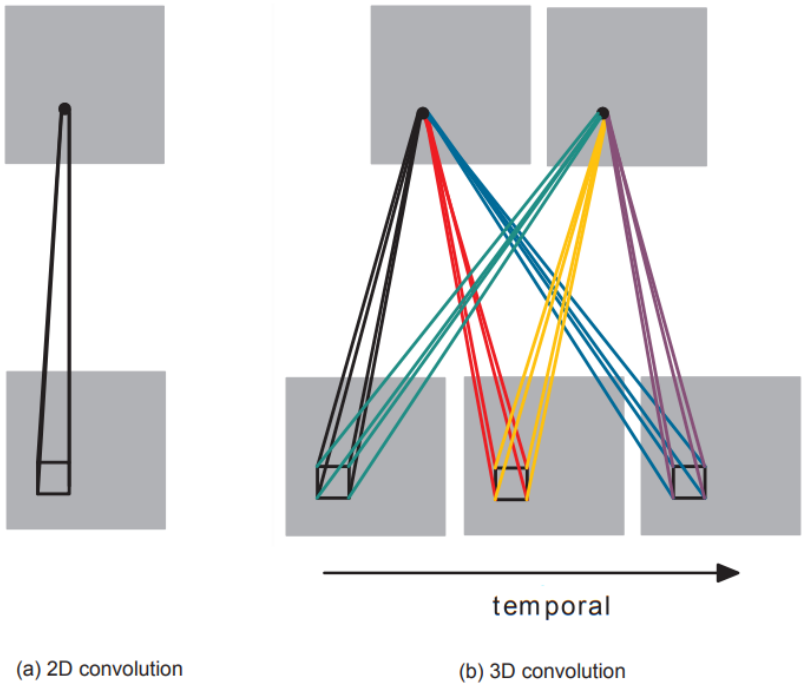


FIGURE 2.22 – This figure shows a comparison of 2D (a) and 3D (b) convolutional filters. The former is initially designed for static images while the latter is well suited for videos. The sets of connections are color-coded so that the shared weights are in the same color. Note that all the 6 sets of connections do not share weights, resulting in two different feature maps on the right. Picture credit for [55].

[55, 184] propose a 3D ConvNet to learn spatio-temporal features and to encode explicitly the motion information in videos. 3D ConvNet is an extention of 2D ConvNet achieved by adding a temporal dimension to 2D convolutional filters and to 2D pooling kernels. [55] conduct the first study for extending 2D ConvNet to 3D ConvNet by designing 3D convolutional filters. The 3D convolution is achieved by convolving the 3D filters with the cube of multiple frames rather than a single frame as in the case of 2D filters. Figure 2.22 shows a comparison of 2D and 3D convolutional filters.
The 3D ConvNet proposed in [55] is relatively shallow, composed of 7 layers. It operates on small-scale datasets and short videos duration of 3 classes (Trecvid) and of 6 classes (KTH).

[184] suggest an improved version targeted for large-scale datasets (Sports-1M about 1.1 million sports videos belonging to 487 sport categories). It first studies extensively the design of 3D convolutional filters, including convolution on single frame, on multiple frames and on a cube of frames as depicted in Figure 2.23. It shows that the choice of their temporal depth is crucial and requires a careful design. From the experimental study, it is concluded that 3D ConvNet learns rich

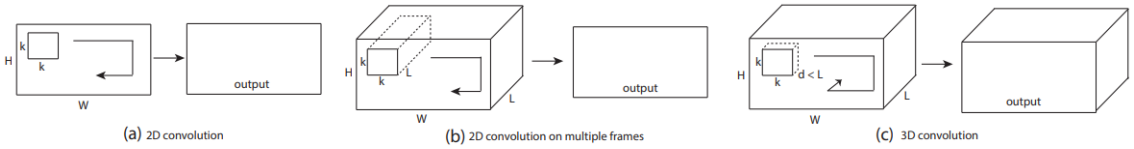


FIGURE 2.23 – This figure shows the different possible types of convolutional filters targeted for video frames. a) Applying 2D convolution on an image results in an image. b) Applying 2D convolution on a video volume (multiple frames as multiple channels) also results in an image. c) Applying 3D convolution on a video volume results in another volume, preserving temporal information of the input signal. Picture credit for [184].

descriptors, including jointly appearance and motion information in contrary to 2D ConvNet. Interestingly, 3D ConvNet captures appearance information from the first few frames then track the motion information in the remaining frames [16]. This study confirms that ImageNet features (based on 2D ConvNet) are not directly well adapted for video data. Figure 2.24 illustrates a feature embedding comparison of ImageNet based on 2D ConvNet and UCF-101 (video dataset described in Section 2.5.1) and based on 3D ConvNet using t-distributed Stochastic Neighbor Embedding (t-SNE) [185]. From this visualization, we observe that 3D ConvNet features are well separated compared to those of ImageNet.

Two stream 3D convolutional networks. Despite the rich spatiotemporal features that 3D ConvNets are able to learn from rgb frames, providing extra motion modality could be advantageous.
Following the success of **Two stream 2D convolutional network**, [28] propose its 3D version. Unlike single 3D ConvNet based on rgb frames, two streams 3D ConvNet operates on rgb frames and on their optical flow components. One stream for each modality. However, training 3D convolutional networks from scratch is challenging and can easily get the model exposed to over-fitting due to the lack of labeled video data. One alternative is to take advantage of the successful 2D ConvNets on image classification, using their pre-trained models.

16. This result is obtained by adding deconvolutional layers [163] to the 3D ConvNet model to interpret its decision

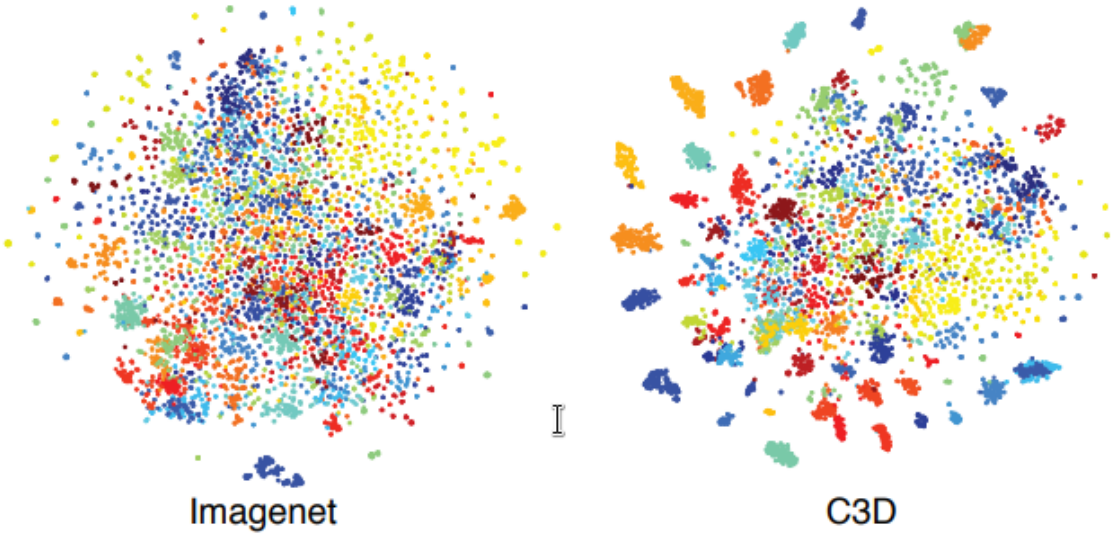


FIGURE 2.24 – Feature embedding visualization of ImageNet (images dataset in the left) based on 2D ConvNet and UCF-101 (videos dataset in the right) and based on 3D ConvNet. Each video is visualized as a point and videos belonging to the same action have the same color. Picture credit for [184].

[28] propose to repeat the (learned) weights of the 2D filters across the time dimension and to rescale them in order to ensure that convolutional filter response is the same. This study compares different 2D/3D ConvNets architectures, including **a) 3D ConvNet on a cube** of rgb frames, **b) separated two-stream 2D ConvNets** respectively on single rgb frames and on a stack of multiple optical flow components of frames, **c) 3D fused two stream ConvNet** similar to the latter, but followed by a 3D convolutional layer that fuses the appearance and motion feature representations resulted from the 2D two streams, before their classification and a **d) two-stream 3D ConvNet based on late score fusion**. It is composed of an appearance stream which operates on multiple rgb frames and a motion stream which takes as input a cube of optical flow components. Figure 2.25 depicts these different cases a), b), c) and d) respectively.

The Experimental results show that d) two stream 3D CNN performs better than other two stream variants. Particularly, it outperforms single 3D ConvNet based only on rgb frames, confirming the complementary aspect of optical flow components even if 3D ConvNet should be able to capture motion information from rgb frames directly. The need of motion modality can be explained by the fact that optical flow is a recurrent algorithm that performs iterative optimization for the flow fields while 2D/3D ConvNets lack of recurrence.

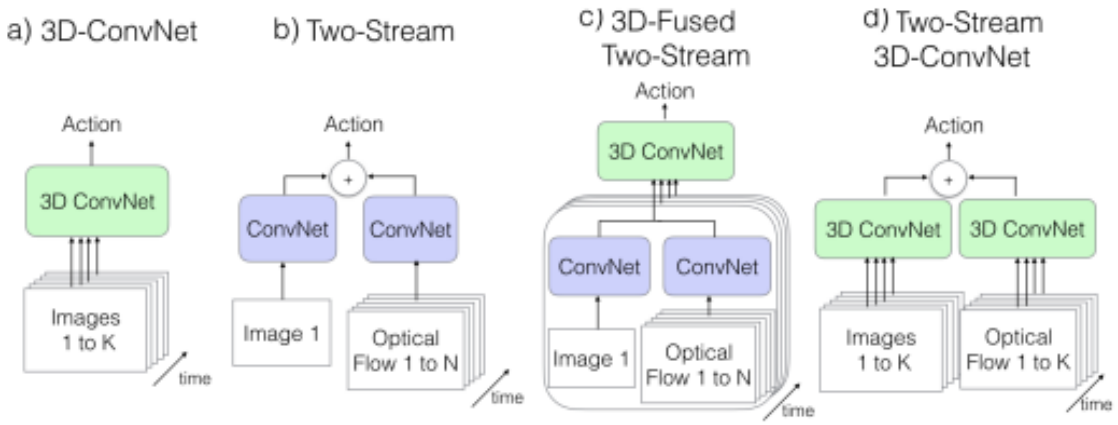


FIGURE 2.25 – This figure shows the different video architectures based on 2D/3D ConvNets, including rgb frames and optical flow based modalities. K stands for the total number of frames in a video, whereas N stands for a subset of neighboring frames of the video. Picture credit for [28].

Another 3D ConvNet based on the extension of DenseNet [165] to 3D is introduced [186]. Unlike [28], [186] rely on a single 3D stream. It is particularly composed of temporal pooling layer with multiple depths in the purpose of capturing different temporal depths. This is achieved by pooling kernels of variable temporal sizes.

Two-stream ConvNets understanding. Despite the success of 2D/3D two streams ConvNets on action recognition task, the understanding of their decision remains unclair. [175] study the spatio-temporal features learned by two stream ConvNets by the visualization of their convolutional filters. Two key conclusions are reported. Early layers show similar spatial structure for appearance and motion while higher layer at fusion level, filters are broadly tuned to multiple speeds and can be specific but also generic across classes.

Visually, it is shown that the filters of the last convolution layer which fuses appearance and motion convolutional representations (see Figure 2.16) are activated by different coloured blobs in the appearance input and by linear motion of circular regions in the motion input (see section 4.2 in [175]) which reflect the spatiotemporal representation of an action. Figure 2.26 displays the activation of a single filter of the last convolutional layer on appearance and motion inputs. The study confirms the importance of rgb and optical flow modalities, as well as their interaction for learning rich video representation.

FIGURE 2.26 – Studying a single filter at layer conv5 fusion : (a) and (b) show what maximizes the unit at the input : multiple coloured blobs in the appearance input (a) and moving circular objects at the motion input (b). (c) shows a sample clip from the test set, and (d) the corresponding optical flow (where the RGB channels correspond to the horizontal, vertical and magnitude flow components respectively). Picture credit for [175].

Hybrid descriptors. [187] assess the complementary of handcrafted and learned video descriptors. Their combination improve action classification performances with an important gap. The method is based on convolutional features extracted from a targeted layer of two stream ConvNets to build a video representation. Different convolutional layers are exploited during the extraction to take the advantage of the different levels of abstraction that each layer provides. The obtained appearance and motion convolutional feature maps are normalized to have the same spatial extent. These feature maps along with trajectories points [17] (handcrafted features) in the video are pooled over the 3D volume. The resulted video representation is called *trajectory-pooled deep convolutional descriptors (TDDs)*. These TDDs are then encoded in Fisher vector prior to their classification with linear SVMs. The process of TDDs construction is depicted in Figure 2.27.

2.4.5 Sequence Models

ConvNets were designed to deal with spatial data. Despite their success on image classification task, they have the issue of ignoring the temporal structure in the case of spatio-temporal data such as videos.
As an alternative, RNNs are proposed to model temporal dependencies in data. However, standard RNNs are not appropriate for videos due to their difficulty of learning over long sequences (videos of several frames), well known as vanishing and exploding gradients problem [53]. To circumvent that, Long Short Term Memory (LSTMs) which is a variant of RNN is introduced. They have the ability to use memory cells in order to store, modify and access internal state for discovering

17. Based on improved dense trajectories [126] described in Section 2.3.1.

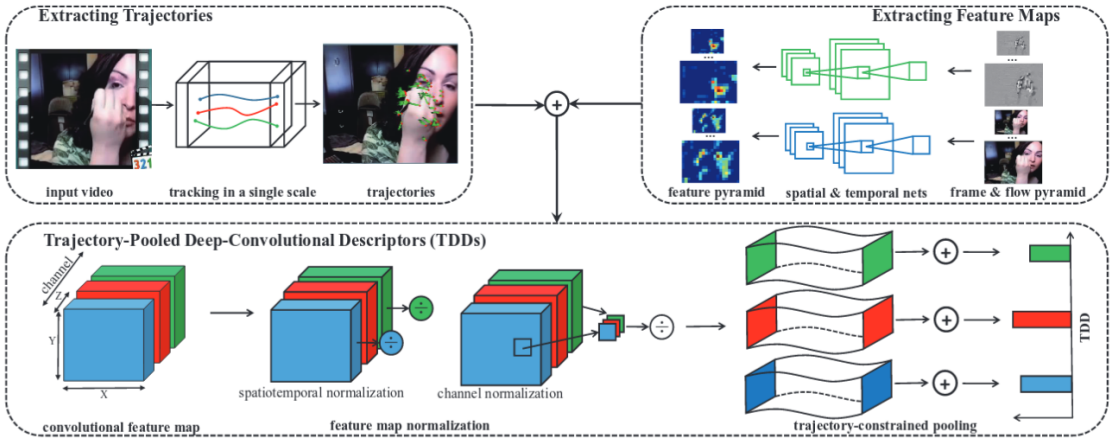


FIGURE 2.27 – The overall scheme of trajectory-pooled deep convolutional descriptors (TDDs) construction. It is composed of three steps. 1) trajectories extraction using improved dense trajectories [126]. 2) Multi-scale convolutional feature maps extraction relying on two stream network [15]. 3) computation of TDDs. Picture credit for [187].

long range temporal dependencies while providing invariant representations to ordering.

[188] suggest LSTMs based model to tackle the problem of action recognition in videos. This model operates on handcrafted features built using Scale-Invariant Feature Transform (SIFT) and BoVW descriptors which impede learning and fine-tuning the features in an end-to-end manner.
In contrast, [189] propose to stack an LSTM layer on the top of 2D ConvNet layers to incorporate explicit motion information modeling, fine-tuning convolutional and recurrent layers jointly in an end-to-end manner directly from (raw) rgb frames and their optical flow components.

In [190], the authors use a 3D ConvNet-LSTM model for video captioning. This model is composed of encoder (ConvNets) and decoder (LSTM) layers designed for the purpose of capturing local spatiotemporal information and an attention mechanism to provide the global context of video action. This attention mechanism is an alternative to global average pooling. Unlike the latter which averages the representations across all the frames, the former applies a weighted average operator to aggregate the representations. The weights associated to the attention layer are based on the LSTM outputs. This encoder-decoder is then extended for the specific task of action recognition [28, 191].

2.4.6 Skeleton based Action Recognition

In the previous sections we presented state-of-art methods for action recognition. These methods operate on sequences of (rgb) frames and their respective optical flow components modalities. Particularly, human actions can be represented by the different trajectories of skeleton joints based on human body connectivity. These joints are described either with their 2D or 3D coordinates. They are used then to build motion information by tracking their trajectories.

Sequence based models. Human action recognition based on skeletons can be seen as time series [192]. Different motion characteristics for each joint are extracted over time to represent human actions. One of the successful models to represent time series is recurrent neural network which is capable of modeling at some extent long range contextual information of variable temporal sequences such as skeletons.

[193] propose a hierarchical recurrent neural network for skeleton classification. The skeleton is divided first into five parts, including left arm, right arm, trunk, left leg and right leg. These parts are then fed into five bidirectional recurrent neural network BDRNN. The respective representations extracted from these BDRNN are fused hierarchically. The last BDRNN before the classification layer contains Long Short Term Memory (LSTMs) units employed in the purpose of overcoming the exploding/vanishing gradient problem [53]. Figure 2.28 illustrate the whole network.

Convolutional models based human pose estimation. Skeleton features based 2D/3D coordinates can be handcraftedly designed or estimated from video frames. This is a tedious task which requires lots of human involvement to annotate them and may result into inaccurate labeling. This task aims at inferring the position of a person and its joints from images or video frames. It rises several challenges.

An image can contain a variable and unknown number of people at any location with different scales. Dynamic visual scenes suffer from complex spatio-temporal overlapping of people induced by their interactions which results into crowded scenes, occlusion of body parts and the difficulty to associate correctly the body parts.

[194] propose a ConvNet model for estimating 2D multi-person pose. This model learns to detect body parts and their association jointly in an end-to-end

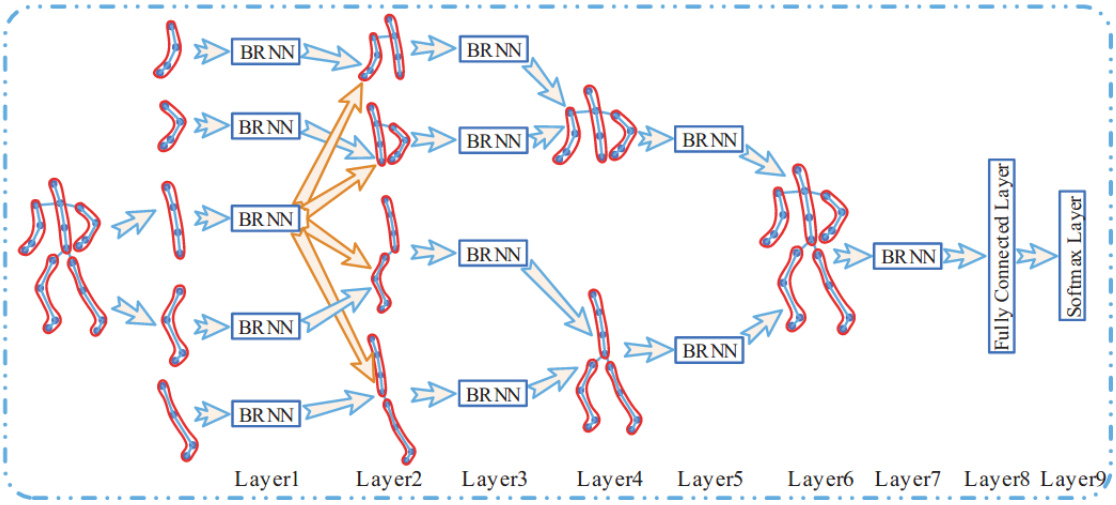


FIGURE 2.28 – This figure shows the proposed hierarchical recurrent neural network for skeleton action recognition. BRNN stands for bi-directional recurrent neural network. Picture credit for [193].

manner. It outputs joint heatmaps indicating the estimated probability of each joint at every pixel and the affinity between pairs of joints. This affinity is useful to associate the different estimated joints into human skeleton. Figure 2.29 displays examples of estimated poses in different environments.

Based on this 2D pose estimator, [195] propose a novel motion representation of video action that encodes the probability distribution of human joints over sequence of frames, as an alternative to optical flow which gives uniform importance to all the pixels independently of the context. This novel representation is based on heatmaps associated to every joint across the frames. They are colorized using a color indicating the time of the frame. The resulted colorized heatmaps of each joint are summed across the video frames to build a fixed-size, high dimensional and sparse video representation. The latter is fed to a ConvNet to achieve action classification. Figure 2.30 depicts the process of poses estimation and their colorization to build video level representation for action classification.
This representation has shown its complementary aspects to optical flow modality [28]. The late fusion of their scores brings important gain in performance. Moreover, the combination of the scores of colorized heatmaps representation, optical flow components and rgb frames modalities allows to achieve state-of-the art results.

Graph ConvNets on skeletons. The generalization of ConvNets on irregular domains [63] such as graphs and manifolds has drawn the attention of computer

FIGURE 2.29 – Examples of poses estimated in different environments. Picture credit for [194].

vision community, particularly the one of human action recognition.

2D/3D human skeletons can be seen as graphs where the nodes and the edges represent respectively the joints and their spatio-temporal interaction through video frames. One of the advantages of graph methods over vectorial ones is their ability to explicitly encode the geometric structure of objects into scenes, in contrary to vectorial methods which vectorize scenes prior to their classification.

Despite the fundamental challenges in designing convolutional and pooling operators on graphs [100], a few solutions have emerged, including spatial methods [85] and spectral ones [18] (see the different families of methods in Figure 2.3) [99].

[196] propose a spatial graph ConvNet, called *STGCN* to achieve action classification on graph skeleton. They represent skeleton as a spatio-temporal graph

18. Spectral methods consist in achieving convolution in the Fourier domain

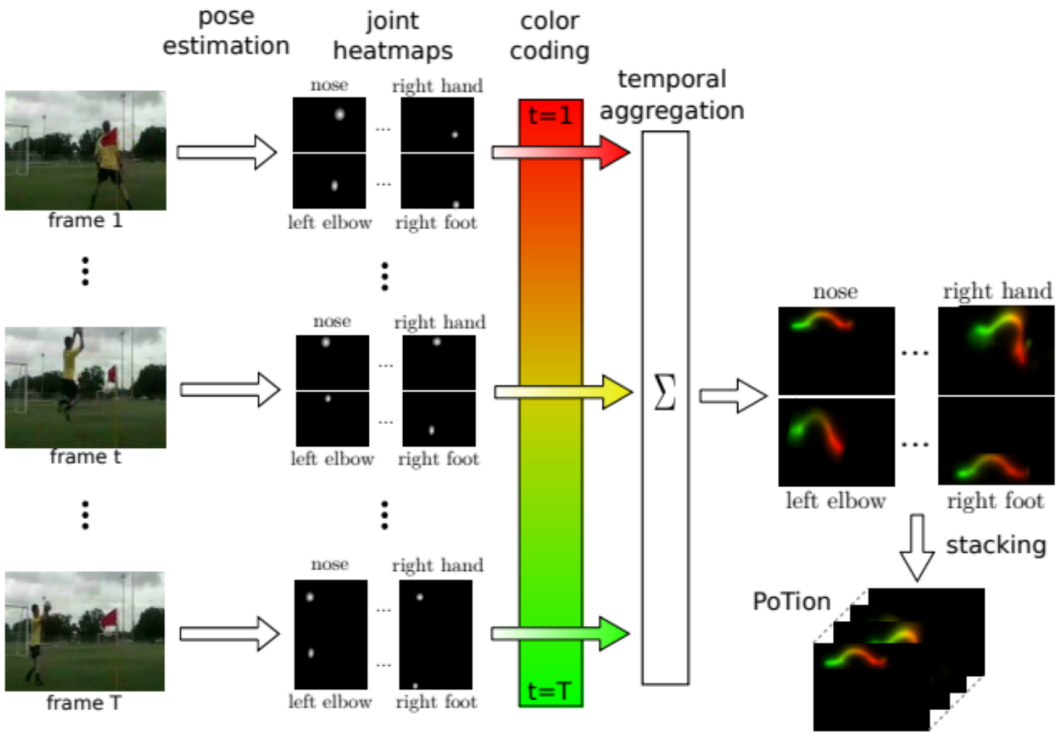


FIGURE 2.30 – This figure illustrates a motion representation of human action. Given a video, joint heatmaps are extracted for each frame and colorized using a color that depends on the relative time in the video clip. For each joint, its colorized heatmaps across the sequence of frames are aggregated to obtain the clip-level video representation with fixed dimension. Picture credit for [195]

where joints are connected spatially according to the human body connectivity and temporally connecting the same joints between successive frames. The resulted graph is then fed to *STGCN* for classification.
STGCN is composed respectively of 2D and 1D convolutional layers to deal with spatial skeletons and temporal ones. Figure 2.31 displays the architecture of *STGCN*.

2.5 Evaluation Datasets

Datasets are crucial to train and to assess the relevance of ML models on the targeted tasks along with corresponding evaluation metrics. The choice of appropriate dataset depends on multiple factors. (i) A suitable dataset should reflect

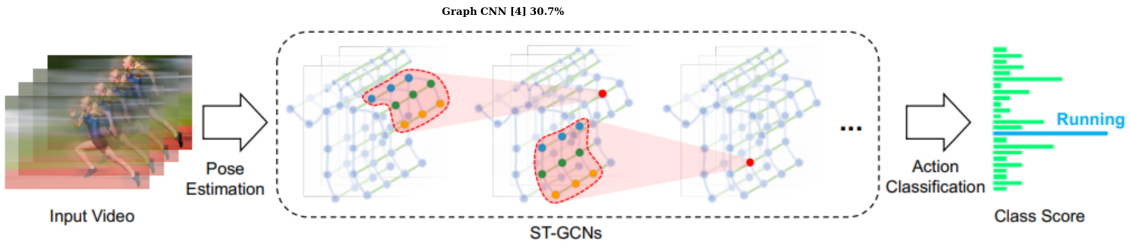


FIGURE 2.31 – This figure shows the architecture of spatio-temporal graph ConvNet (*STGCN*). Inputs of *STGCN* consist of a collection of skeletons estimated from rgb video frames, relying on a ConvNet for pose estimation. Multiple layers of spatial-temporal graph convolution *STGCN* will be applied and gradually generate higher-level feature maps on the graph. It will then be classified by the standard Softmax classifier to the corresponding action category. Picture credit for [196].

the constraints and challenges faced in concrete applications, as well as ensuring that it covers large enough variability of real and complex environments. (ii) Its size plays a crucial role in evaluating the validity of ML models and their scalability. Moreover, ML models are data-hungry, particularly DL ones which are quickly exposed to overfitting when trained on small datasets. (iii) Budget is a major bottleneck to cope with. Successful models are data and computational resources consuming. Finding a trade-off between training complex models (on large datasets) and computational efficiency requires a careful design.

In this thesis, we choose datasets that cover the challenges discussed in Section 1.3.3 of reasonable size to fit our computational resources. Including different modalities, ranging from 2D/3D skeletons to RGB frames in order to show the flexibility of our models and hence best fit real data that come from different sources. Moreover, this choice relies also on the input requirements of models : vectorial deep learning and geometric deep learning. Table 2.2 gives an overview of the selected datasets.

2.5.1 UCF-101

UCF-101 is an action recognition dataset of realistic videos collected from YouTube. It is an extension of UCF-50 dataset. Table 2.3 summarizes the its characteristics. It is particularly challenging due to its large variation in camera motion, large intra-class variability, the presence of multiple persons (with different poses) and objects at different scales. The action categories are divided into five types as depicted in Figure 2.32 : human-object interaction, body-motion only, human-human interaction, playing musical instruments and sports.

Dataset	#Actions	Clips	Background	Camera motion	Modalities		Resource	#Splits	Evaluation metric
					RGB	Skeletons			
UCF	101	13320	D	✓	✓	✗	Y	3	Accuracy
HMDB	51	6766	D	✓	✓	✗	M, Y, W	3	Accuracy
JHMDB	21	928	D	✓	✓	2D	M, Y, W	3	Accuracy
SBU	8	282	S	✗	✓	3D	L	5	Accuracy

TABLE 2.2 – Summary of datasets : UCF [197], HMDB [198], JHMDB [199] and SBU [200]. From RGB frames, optical flow modality is computed to represent the motion information in videos. D, S, M, Y, W and L stand respectively for Dynamic, Static, Movies, YouTube, Web and Laboratory environment. The splitting process of each dataset is described in Section 2.5.4.

Actions	101
Clips	13320
Groups per action	25
Clips per group	4-7
Mean clip length	7.21 sec
Total duration	1600 mins
Min clip length	1.06 sec
Max clip length	71.04 sec
Frame rate	25 FPS
Frame size	320×240
Audio	Yes (51 actions)

TABLE 2.3 – Summary of the characteristics of UCF-101.

2.5.2 HMDB-51 and JHMDB-21

HMDB-51 is an action category dataset of realistic videos collected from multiple sources including movies, Prelinger archive database, YouTube and Google videos. Each action category contains at least 101 clips. The actions are grouped into five types : general facial actions, facial actions with object manipulation, general body movement, body movement with object interaction, body movements for human interaction. Table 2.4 summarizes the different characteristics of HMDB-51 and its subset JHMDB-21.

Clips have diverse contents and are taken under extremely uncontrolled conditions such the ones of UCF-101 described in Section 2.5.1. Moreover, HMDB-51 has its specific conditions including clip quality, visibility of body parts as illustrated in Figure 2.33. The 51 action categories belonging to HMDB and 21 ones of JHMDB are illustrated in Figure 2.34.

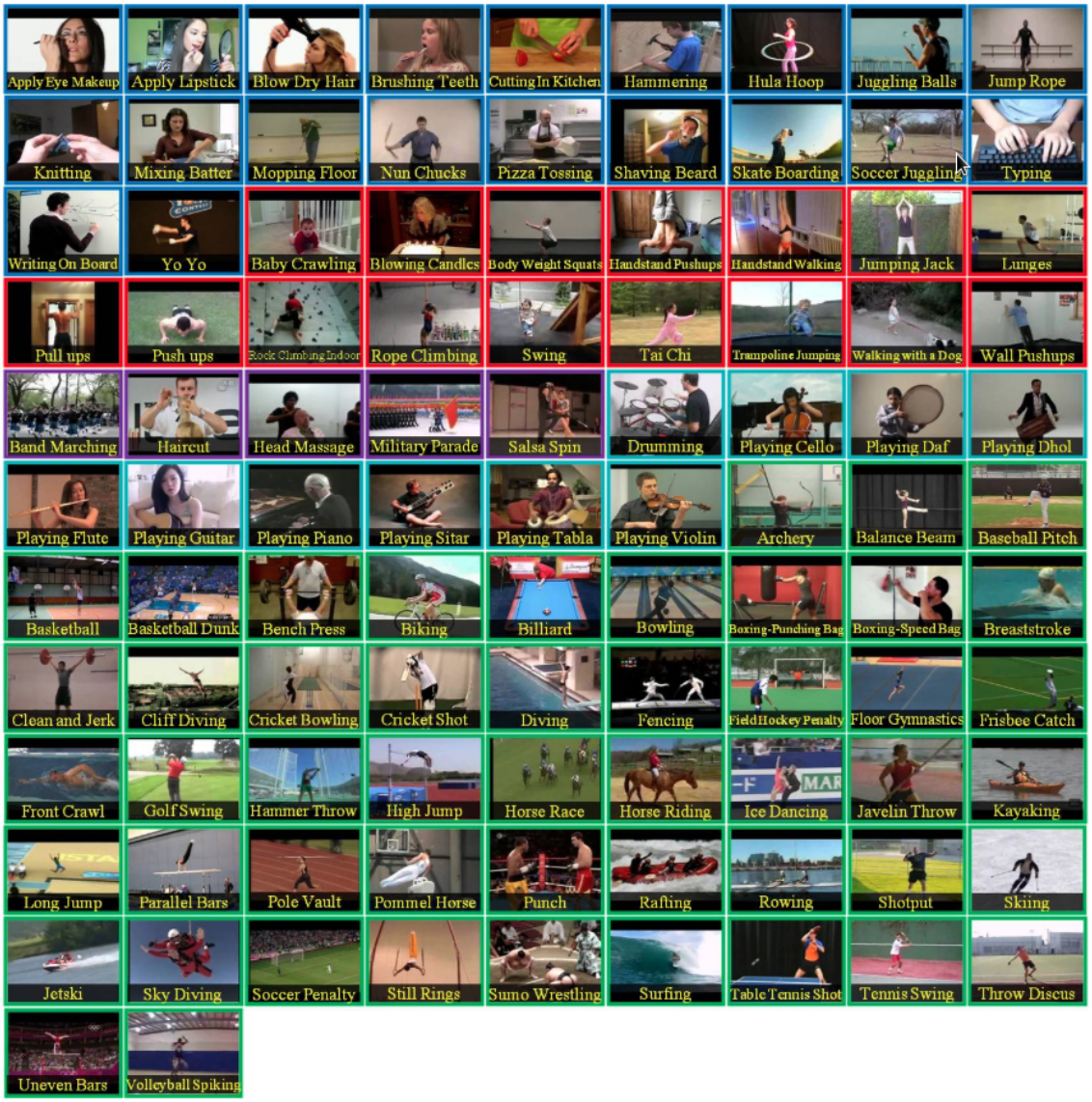


FIGURE 2.32 – UCF-101 dataset. The color of frame borders indicates to which action category they belong : Human-object Interaction, Body-Motion only, Human-Human interaction, Playing musical instruments, Sports. Picture credit for [197].

One of the particularity of **JHMDB-21** compared to HMDB-51 is its extra 2D skeleton modality which can serves also as pose estimation dataset. However, annotating all the clips of HMDB-51 to get the 2D coordinates of persons at different frames is time consuming and adds extra difficulty due to variations in pose, human sizes, motion blur, partial body visibility. For that reason, 21 action categories from HMDB-51 are considered (in JHMDB-21) in the way that they correspond to single person actions such as run, throw, shoot, etc. as illustrated in Figure 2.34.

	HMDB	JHMDB
Actions	51	21
Clips	6766	928
Frame rate	30* FPS	30* FPS
Frame size	320 × 240	320 × 240
Skeletons	✗	2D

TABLE 2.4 – This table summarizes the characteristics of HMDB-51 dataset and of its subset JHMDB-21. * stands for variable FPS. This variability is due to the collection of videos from different sources. FPS is then converted to 30 FPS for all the clips.

In spite of this, it remains challenging to achieve action classification as only 2D joints coordinates through the sequences of frames are provided, deprived from rich appearance and motion information.

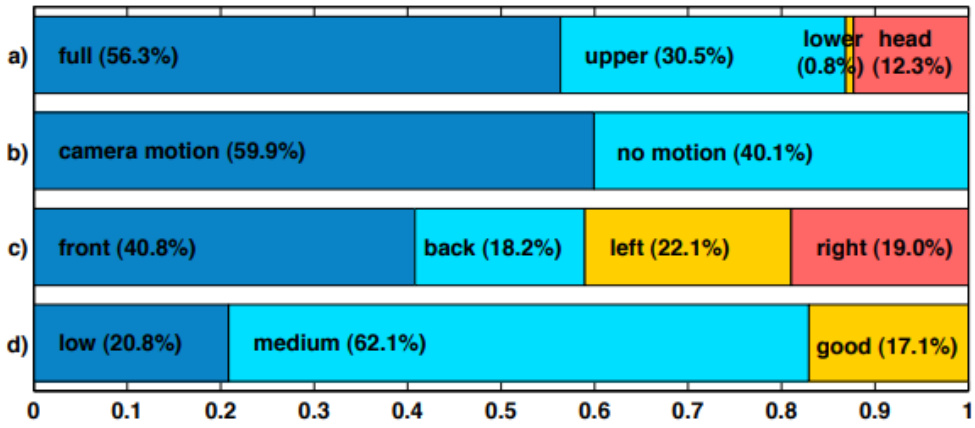


FIGURE 2.33 – Specific characteristics of HMDB-51. a) Visible body part, b) Camera motion, c) Camera view point and d) Clip quality. Picture credit for [198].

2.5.3 SBU

SBU-8 is a two-person interaction dataset acquired in a laboratory environment using the Microsoft kinect sensor which provides an adequate accuracy of real-time full-body tracking with low cost [200]. It includes eight types of actions performed by seven participants, namely : approaching, departing, pushing, kicking, punching, exchanging objects, hugging and shaking hands as shown in Figure 2.35. These actions are relatively challenging (but less challenging than UCF-101, HMDB-51 and JHMDB-21) because they are non-periodic actions and

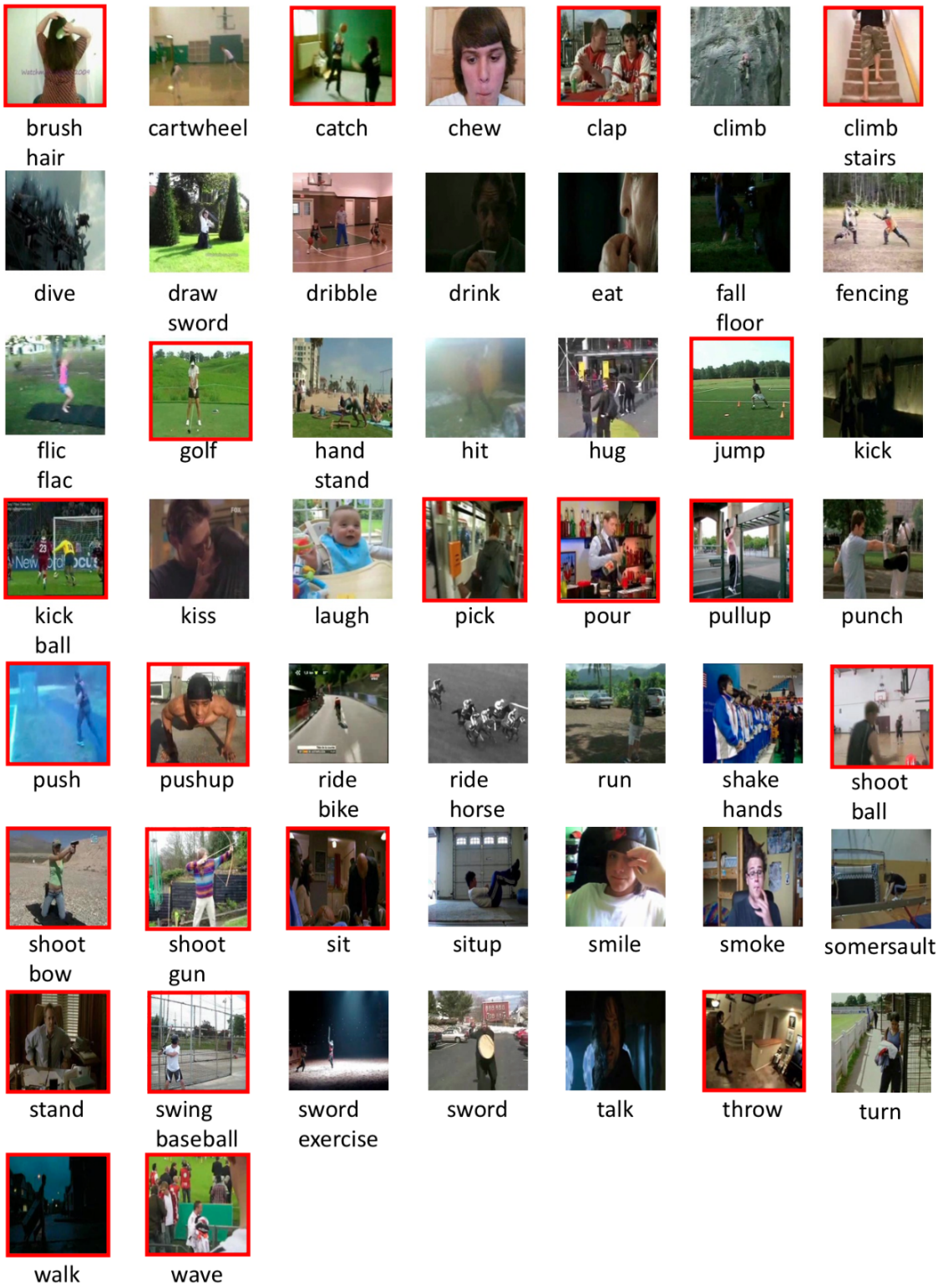


FIGURE 2.34 – HMDB-51 dataset. The red color of frame borders indicates the action categories that belong to JHMDB-21 dataset.

Actions	8
Number of participants	7
Clips	282
Frame rate	15 FPS
Frame size	640×480
Skeletons	3D

TABLE 2.5 – Summary of characteristics of SBU.

have similar body movements such as shaking hands and exchanging objects. The entire dataset has 282 clips including three modalities, RGB images, depth map and 3D skeletons (which are more accurate than the 2D skeletons of JHMDB-21). The general characteristics of SBU are summarized in Table 2.5.

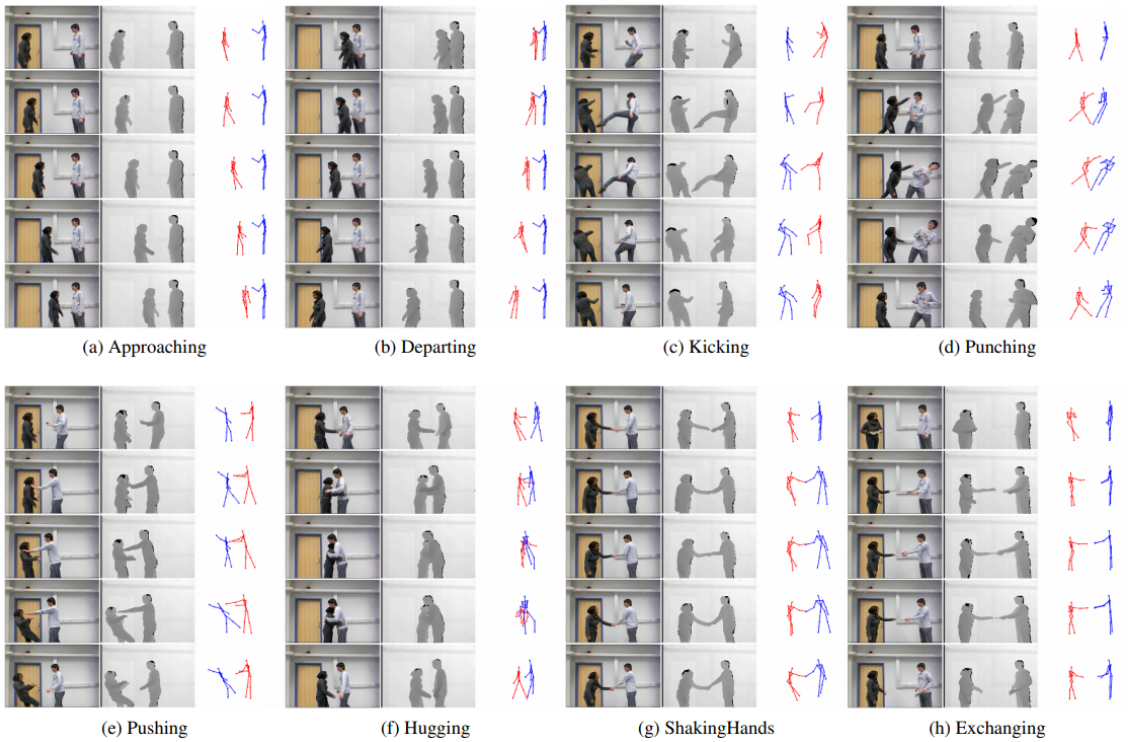


FIGURE 2.35 – Visualization of the eight two-persons interaction actions belonging to SBU-8 dataset. Picture credit for [200].

In our work, SBU-8 and JHMDB-21 based respectively on 3D and 2D skeletons are of particular interest to achieve action recognition with geometric deep learning models, where skeletons represent graphs.

2.5.4 Train and Test Splits Construction

UCF-101. Three distinct pairs of training and test splits are generated. They have been constructed in a way to keep the action groups separate, ensuring that action clips from the same group are not shared in the same train and test split since the clips which belong to the same group are obtained from a single long video [197]. The clips of one action category are divided into 25 groups which contain four up to seven clips each. Each test split has 7 different groups and their respective remaining 18 groups are used for training [197].

HMDB-51. It is composed of three distinct pairs of train and test splits. They have been built in a way to guarantee that clips from the same video are not used in the same train and test split. In addition to, for each action category, 70 training and 30 testing clips are selected w.r.t to the 70/30 balance for each characteristic (see Figure 2.33)[198].
JHMDB-21 follows the same splitting protocol as the one proposed for HMDB-51 [199].

SBU-8. From 282 clips, divided into 21 sets [19] of two actors, five folds of 4-5 two-actor sets are constructed. Four folds are used for training and one for testing. This partitioning ensures that each two-actor set appears only in training or only in testing [200].

Accuracy metric. It is the most commonly used metric to evaluate the performances of supervised models in the task of classification. It is defined as the ratio of the number of correct test predictions to the total number of examples in the test set. In the case of several splits, the overall performance corresponds to the average accuracy over all the test splits.

2.6 Conclusion

In this chapter, we have provided the development of human action understanding, and we have reviewed several video representation methods including handcrafted and learned approaches. We have also discussed several classes of standard deep learning architectures and particularly those targeted for the task of action recognition, which are the most relevant to our work. This discussion includes the emerging field of geometric deep learning and its recent applications

19. Each set contains clips of a pair of different persons performing all the 8 actions. In each action, one person is acting and the other person is reacting

in action recognition. Moreover, we have presented different video action modalities including rgb frames and their optical flow components, as well as 2D/3D joint skeletons on which machine learning and deep learning models can operate. Finally, we have described the different datasets, and their proposed training and test split procedures used to assess our video representations and our models.

Despite the different modalities and the multifariousness of action recognition datasets we have, it is still challenging to achieve action recognition. The challenges include lack of understanding of the most important video features, and the properties of machine learning and deep learning models to achieve action recognition.

In the next chapter, we present our first contribution which consists in handling videos with varying length and context and also the variable temporal granularity in action categories. Our solution is based on temporal pyramid representation of videos, and on multiple aggregation learning.

CHAPITRE 3

MULTIPLE AGGREGATION NETWORKS FOR ACTION RECOGNITION

Contents

Chapitre abstract

Deep Convolutional Neural Networks (ConvNets) are nowadays achieving significant leaps in different pattern recognition tasks including action recognition. ConvNets stack multiple convolutional, pooling and fully connected layers. These networks are increasingly deeper, data-hungrier and this makes their success tributary to the abundance of labeled training data. While convolutional and fully connected operations have been widely studied in the literature, the design of pooling operations that handle action recognition, with different sources of temporal granularity in action categories, has comparatively received less attention, and existing solutions rely mainly on max or averaging operations. The latter reduce dimensionality of output layers (and hence attenuate their sensitivity to the availability of labeled data); however, this process may dilute the information of upstream convolutional layers and thereby affect the discrimination power of the trained representations, especially when the learned categories are fine-grained.

Therefore, these existing pooling operators are clearly powerless to fully exhibit the actual temporal granularity of action categories and thereby constitute a bottleneck in classification performances.
In this chapter, we introduce a novel hierarchical pooling design that captures different levels of temporal granularity in action recognition. Our design principle is coarse-to-fine and achieved using a tree-structured network; as we traverse this network top-down, pooling operations are getting less invariant but timely more resolute and well localized. Learning the combination of operations in this network which best fits a given ground-truth is obtained by solving a constrained minimization problem whose solution corresponds to the distribution of weights that capture the contribution of each level (and thereby temporal granularity) in the global hierarchical pooling process. Besides being principled and well grounded, the proposed hierarchical pooling is also video-length and resolution agnostic. Extensive experiments conducted on the challenging UCF-101, HMDB-51 and JHMDB-21 databases corroborate all these statements.

The work in this chapter has led to the publication of two conference papers :

- Ahmed Mazari and Hichem Sahbi. Deep Temporal Pyramid Design for Action Recognition. In the 44th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Brighton, United Kingdom, 12-19 May 2019, pp. 2077-2081.
- Ahmed Mazari and Hichem Sahbi. Coarse-To-Fine Aggregation For Cross-Granularity Action Recognition. In the 27th IEEE International Conference on Image Processing (ICIP). Abu Dhabi, United Arab Emirates, 25-28 October 2020.

3.1 Introduction and Related Work

Action recognition is standing as one of the most challenging problems in video processing which consists in assigning one or multiple semantic categories to moving objects. The challenge in this task stems from (i) the difficulty to learn mapping models that assign action categories to frames while being resilient to the intrinsic properties of actions (human appearance and motion, articulation, velocity, etc.) and also their extrinsic acquisition conditions (camera motion and spatial-temporal resolution/scale/length, illumination, occlusion, cluttered

background, etc.), as well as (ii) the hardness in hand-labeling large collections of training videos prior to build these mapping models. Therefore, this affects the accuracy of multiple related applications such as scene understanding [30, 201, 202], video surveillance [203, 204], video caption generation and retrieval [31, 205-213] as well as human computer interaction and robotics [214-217]. Most of the existing action recognition solutions are based on machine learning (ML) [17, 19-22, 26, 126, 152]; their general recipe consists in learning functions that map visual content representations of frame sequences (either handcrafted or learned [218-221]) into categories using widely known ML algorithms such as random forests, support vector machines [20, 21, 222] and deep networks [15, 28, 175, 176, 179, 180, 195].

Among the ML solutions for action recognition those based on deep networks are currently witnessing a major interest [16, 88, 223-226] but their success is tributary to the availability of large amount of labeled training data and also the appropriate choice of their architectures including convolutional and recurrent ones [15, 28, 175, 176, 179, 180, 195, 227]. In particular, convolutional networks are designed by stacking multiple convolutional, pooling and fully connected layers; successful architectures for action recognition include two-stream 2D/3D ConvNets [15, 176] operating on appearance and motion flows, and ConvNets combined with Long Short-Term Memory (LSTM) networks [228] that capture coarse temporal structure of actions as well as 3D ConvNets [28] which capture fine (local) temporal structures. However, and beside issues related to scarcity of labeled data [1] and the large number of training parameters [2] (especially in 3D ConvNet models), the effort in the design of deep networks, that capture the relevant motion information in videos, has been focused essentially on optimizing their convolutional and fully connected layers [3] while *comparatively* the design of optimized pooling layers received less attention especially on non-vectorial data including video sequences. The difficulty in designing architectures with suitable pooling (a.k.a aggregation) operators, particularly on video sequences stems from the eclectic properties of videos (namely their duration, temporal resolution and velocity of moving objects as well as the granularity of their action categories) and this makes pooling design very challenging. This challenge is further exacerbated by the lack of labeled video data (covering all the variability) compared to other

1. Labeled video data are usually difficult to collect and expensive even at reasonable frame rates

2. Training and fine-tuning ConvNets (together with their hyper-parameters) for the challenging task of action recognition is known to be memory and time demanding even when using highly efficient GPU resources and reasonable size videos

3. Convolutions and multi-layer perceptron have been largely studied since the early age of artificial neural networks and also in other problems in image processing including wavelet filter design

neighboring problems such as image classification that benefit from labeled sets which are at least an order of magnitude larger compared to the current action recognition datasets while the task is inherently far more challenging; as a result, these action recognition models are more subject to overfitting.

In order to attenuate such effect, pooling methods [69, 71, 229] have been designed, and most of them are based on global measures including max and averaging operators. Pooling plays a key role in reducing the dimensionality of convolutional feature maps and thereby the number of training parameters and enhances the resilience, of the learned ConvNet representations, to the lack of training data and to the acquisition conditions. However, it comes at the detriment of some relative loss in the discrimination power especially when video data belong to fine-grained action categories. Indeed, pooling contributes in diluting (averaging) convolutional features which are highly important in discriminating fine-grained categories and these averaging operators are rather more appropriate for coarse-grained actions (see Figure 3.1). Alternative and more recent solutions [15, 28, 175, 176, 179, 180] rely on sampling and stacking ConvNet features in order to build *spectrogram-like* fixed length representations that also preserve the granularity of video actions. Nonetheless, both methods suffer from several drawbacks; on the one hand, pooling methods based on global statistical measures are time/duration agnostic (and hence invariant) but less discriminating while spectrogram-like (see Figure 3.7) methods are discriminating but time/duration aware (less invariant) and highly sensitive to the aforementioned video acquisition conditions and may result into a loss of information, especially when videos are not well resolute.

A more suitable pooling should gather the advantages of these two families of methods while discarding their inconvenients. Following this goal, we consider in our work a hierarchical aggregation scheme that describes moving scenes at multiple temporal granularities while also being resilient to their highly variable acquisition conditions. Top levels in this hierarchical aggregation provide orderless (invariant) but less discriminating motion and appearance representations which capture coarse-grained action categories (as global averaging technique) *while* bottom levels correspond to fine-grained, timely resolute and order-sensitive video representations [227, 230]. The design principle of our proposed solution is coarse-to-fine and allows us to capture a gradual change of invariance and granularity; as we traverse the hierarchy top-down, our video representations are getting less invariant but timely more resolute and fine-grained. However, knowing a priori which levels in this hierarchy are the most appropriate in order to capture the actual granularity of our video data is challenging and also *combinatorial*; hence, learning this combination "end-to-end" and in a differentiable

manner is rather more appropriate.

Considering this line of research, other related works [231-235] try to model granularity of actions in videos by incorporating specific modules into ConvNets. The method in [231] samples, from each video, frames as well as their associated optical flow components and adds a spatio-temporal pyramid module to ConvNet in order to capture hierarchical relationships between appearance and motion features. The method in [232] stacks a temporal pyramid pooling layer on the top of motion and appearance ConvNet streams in order to build fixed-length video representations. In [233], authors sample a set of frames by first splitting videos into segments and taking frames from each segment, and build a spatial pyramid to extract multi-scale appearance features from different convolutional layers. These features are then concatenated and fed to a three level temporal pyramid. The work in [234] samples video frames at different temporal resolutions, and feeds them to a 3D ConvNet in order to extract their respective features followed by a temporal pyramid which down-samples and concatenates the resulting features. Finally, the method in [235] achieves frame sampling followed by a temporal pyramid pooling in order to build features at different pyramidal levels; the resulting features are afterwards fed to a temporal relational layer that groups these features at different scales. While all these methods rely on a hierarchical temporal aggregation scheme, none of them considers the issue of learning the best combination of levels in these temporal aggregation hierarchies, and this turns out to be highly effective as shown in the following sections.

In this chapter we introduce a novel scheme for action recognition based on Multiple Aggregation Networks. Given a hierarchy of aggregation operations, the goal is to learn a combination of these operations that best fits a given action recognition ground-truth. We solve this problem by minimizing a constrained objective function whose parameters correspond to the distribution of weights through multiple aggregation levels; each weight captures the granularity of its level and its contribution in the global learned video representation. Besides handling aggregation at different levels, the particularity of our solution resides in its ability to handle variable length videos (without any up or down-sampling) and thereby makes it possible to fully benefit from the whole frames in videos.

The rest of this chapter is organized as follows. First, we describe in Section 3.2 our motion and appearance streams used to build frame-wise representations. Then, we introduce in Section 3.3 our main contributions; a method based on "linear/nonlinear kernel" combination as well as "end-to-end" two stream ConvNets that aggregate and combine the obtained frame-level representations into tempo-

FIGURE 3.1 – Examples of *fine* and *coarse-grained* actions. The first row shows three action categories from the MLB-YouTube dataset [236] : "No swing", "Swing" and "Bunting" which are difficult to distinguish as they have very small differences. The second row shows two instrument playing actions from the UCF-101 dataset [197] : "cello" and "violin" which are also difficult to distinguish as their arm/hand locations and directions are similar. In contrast, the third row shows "Pat on back", "Butt kick" and "Shaking hand" actions (taken from NTU RGB+D dataset [237]) which are relatively easier to distinguish.

ral pyramids in order to achieve action recognition. Finally, we show, in Section 3.4, the validity of these contributions through extensive experiments using standard and challenging video datasets including UCF-101, HMDB-51 and JHMDB-21.

3.2 Frame-wise Two-Stream Video Description at a Glance

We consider a collection of videos $\mathcal{S} = \{\mathcal{V}_i\}_{i=1}^{n}$ with each one being a sequence of frames $\mathcal{V}_i = \{f_{i,t}\}_{t=1}^{T_i}$ and a set of action categories (a.k.a classes or categories) denoted as $\mathcal{C} = \{1, ..., C\}$. In order to describe the visual content of a given video $\mathcal{V}_i$, we rely on a two-stream process (see Figure 3.2); the latter provides a complete description of appearance and motion that characterizes the spatio-temporal aspects of moving objects and their interactions. The output of the appearance

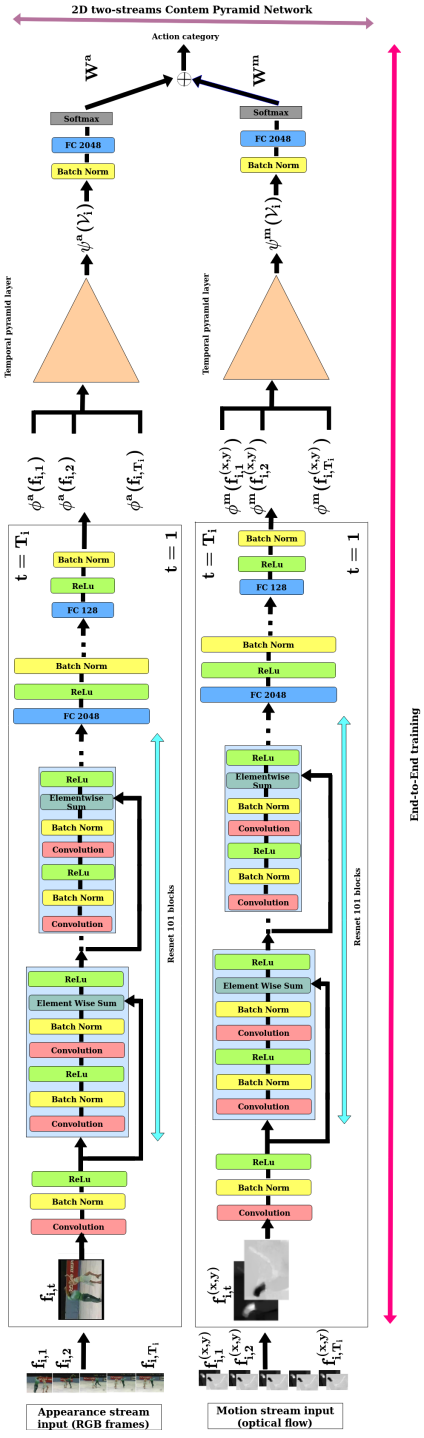


FIGURE 3.2 – Our two stream network including a ResNet block, a temporal pyramid block and "batch norm+fully connected+softmax+late fusion" layers. The temporal pyramid block achieves pooling either by weighted averaging or weighted concatenation (see Equation 3.1 and also Figure 3.3) (**Better to zoom the PDF version**).

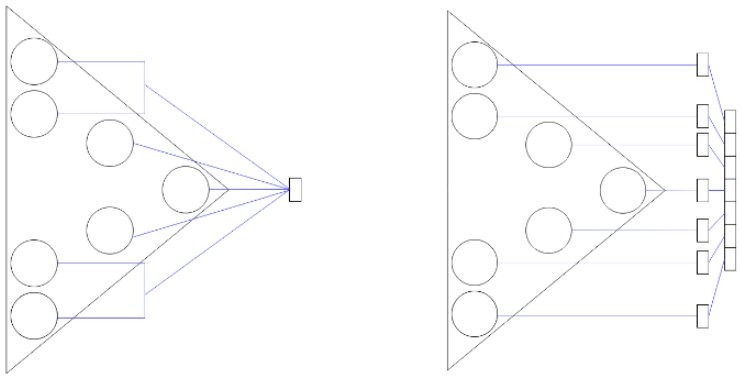


FIGURE 3.3 – Aggregation by "averaging" vs. aggregation by "concatenation".

stream (denoted as $\{\phi^a(f_{i,t})\}_{t=1}^{T_i} \subset \mathbb{R}^{2048}$) is based on the deep residual network (ResNet-101) [4] trained on ImageNet [223] and fine-tuned on UCF-101 [197] while the output of the motion stream (denoted as $\{\phi^m(f_{i,t})\}_{t=1}^{T_i} \subset \mathbb{R}^{2048}$) is also based on the ResNet 101 network but trained on optical flow image pairs [178, 238]; these pairs correspond to the horizontal and the vertical displacement fields which are linearly transformed in order to make their ranges between 0 and 255.
Besides the high performances reported in ImageNet classification [239], the particularity and the strength of ResNet resides in its skip connections which (i) reduce the sensitivity of the network to its architecture and (ii) reduce the effect of gradient collapse/explosion thereby making the optimization and fine-tuning of this network parameters (through stochastic gradient descent) effective and numerically more stable.

Following the line in [15, 178] and in order to adapt the pretrained ResNet-101 to optical flow data, we slightly update the input layer of the original ResNet [5]. Indeed, the number of channels is reset to 20 instead of 3 in the original ResNet; the initial weights of these 20 channels are obtained by averaging the 3 original (appearance) channel weights and by replicating their values through the 20 new motion channels. During training, closely related methods (namely [178]) split each video into N continuous segments, and for each segment, a frame f is randomly selected to feed an appearance stream ResNet and a stack of optical flow is also taken (starting from f) as an input to the motion stream. In the setting of [178], scores obtained from the softmax layers of motion and appearance streams are fused through different frames using a *segmental consensus* function in order to make class prediction at the video level; in other words, for each

4. Or on ResNet-152 trained on ImageNet only. See Table 3.1.
5. Already available/pretrained on ImageNet to capture the appearance.

test video, 19 frames[6] are uniformly sampled and passed through appearance and motion streams and their scores are combined as votes for all the action categories. As shown subsequently, and in contrast to [178], our proposed method relies on a different aggregation scheme that models *coarse as well as fine grained action categories*; our method does not require any frame (re)sampling which may degrade performances (as also shown later in experiments) indeed, our method effectively leverages the entire set of video frames.

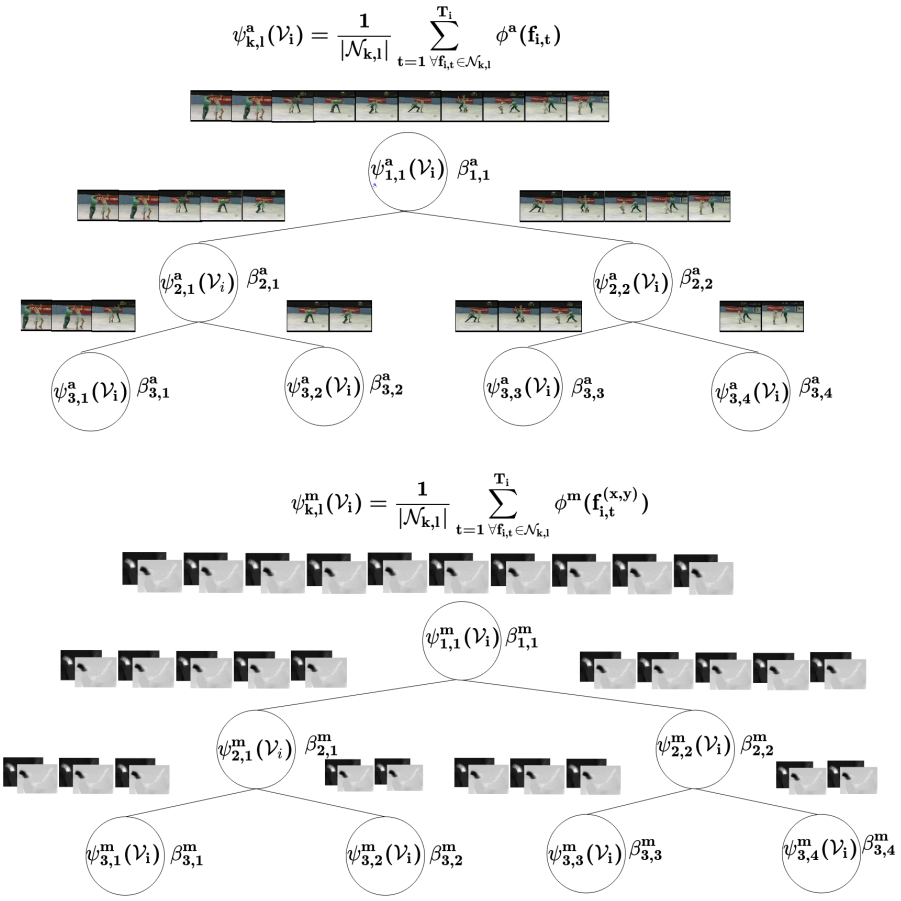


FIGURE 3.4 – This figure shows frame aggregation at each node of the temporal pyramid for appearance (top) and motion streams (down). $\phi^a(f_{i,t})$ stands for the appearance representation of the t^{th} frame of video $\mathcal{V}_i$. It can be based on the deep residual network (ResNet-152) trained on ImageNet or on ResNet-101 trained on ImageNet and then fine-tuned on UCF-101. $\phi^m(f_{i,t})$ is the motion representation of the t^{th} frame of video $\mathcal{V}_i$ obtained with ResNet-101 trained on optical flow data of UCF-101 dataset.

6. The reason for choosing 19 frames is explained by the fact that the minimum number of video frames in UCF-101 is 28, hence 19 is the maximum number from which a stack of 10 optical flow frames can be taken.

3.3 Multiple Aggregation Learning

Given a video $\mathcal{V}_i$ with T_i frames, we define $\mathcal{N}$ as a tree-structured network with depth up to D levels and width up to 2^{D-1}. Let $\mathcal{N} = \cup_{k,l}\mathcal{N}_{k,l}$ with $\mathcal{N}_{k,l}$ being the k^{th} node of the l^{th} level of $\mathcal{N}$; all nodes belonging to the l^{th} level of $\mathcal{N}$ define a partition of the temporal domain $[0, T_i]$ into 2^{l-1} equally-sized sub-domains. A given node $\mathcal{N}_{k,l}$ in this hierarchy aggregates the frames that belong to its underlying temporal interval. Each node $\mathcal{N}_{k,l}$ also defines an appearance and a motion representation respectively denoted as $\psi^a_{k,l}(\mathcal{V}_i)$, $\psi^m_{k,l}(\mathcal{V}_i)$ and set as $\psi^a_{k,l}(\mathcal{V}_i) = \frac{1}{|\mathcal{N}_{k,l}|}\sum_{t\in\mathcal{N}_{k,l}}\phi^a(f_{i,t})$, $\psi^m_{k,l}(\mathcal{V}_i) = \frac{1}{|\mathcal{N}_{k,l}|}\sum_{t\in\mathcal{N}_{k,l}}\phi^m(f_{i,t})$; see Figure 3.4). Depending on the level in $\mathcal{N}$, each representation captures a particular temporal granularity of motion and appearance into a given scene; it is clear that top-level representations capture coarse visual characteristics of actions while bottom-levels (including leaves) are dedicated to fine-grained and timely-resolute sub-actions. Knowing a priori which levels (and nodes in these levels) capture the best – a given action category – is not trivial. In the remainder of this section, we introduce a novel learning framework which achieves multiple aggregation design and finds the best combination of levels and nodes in these levels that fits different temporal granularities of action categories.

Considering the motion stream, we define – for each node $\mathcal{N}_{k,l}$ – a set of variables $\beta_m = \{\beta^m_{k,l}\}_{k,l}$ (with $\beta^m_{k,l} \in [0,1]$ and $\sum_{k,l}\beta^m_{k,l} = 1$) which measure the importance (and hence the contribution) of $\psi^m_{k,l}(\mathcal{V}_i)$ in the global motion representation of $\mathcal{V}_i$ (denoted as $\psi^m(\mathcal{V}_i)$). Precisely, two variants are considered for ψ^m

$$\begin{aligned} &(*) \quad \psi^m(\mathcal{V}_i) &&= \left(\beta^m_{1,1}\psi^m_{1,1}(\mathcal{V}_i)\dots\beta^m_{k,l}\psi^m_{k,l}(\mathcal{V}_i)\dots\right)^\top \\ &(**) \quad \psi^m(\mathcal{V}_i) &&= \sum_{k,l}\beta^m_{k,l}\psi^m_{k,l}(\mathcal{V}_i). \end{aligned} \tag{3.1}$$

As shown in Equation 3.1, the variant in (*) corresponds to a concatenation scheme while (**) corresponds to averaging. The former relies on the hypothesis that nodes in $\mathcal{N}$ (and hence sub-actions in different videos) are well aligned whereas the latter relaxes this hypothesis (See Figure 3.5 and Equation 3.2 later). Similarly to motion, we define the aggregations and the set of variables $\beta_a = \{\beta^a_{k,l}\}_{k,l}$ associated to appearance stream. In the remainder of this section, and unless explicitly mentioned, the symbols m, a are omitted in the notation and all the subsequent formulation is applicable to motion as well as appearance streams.

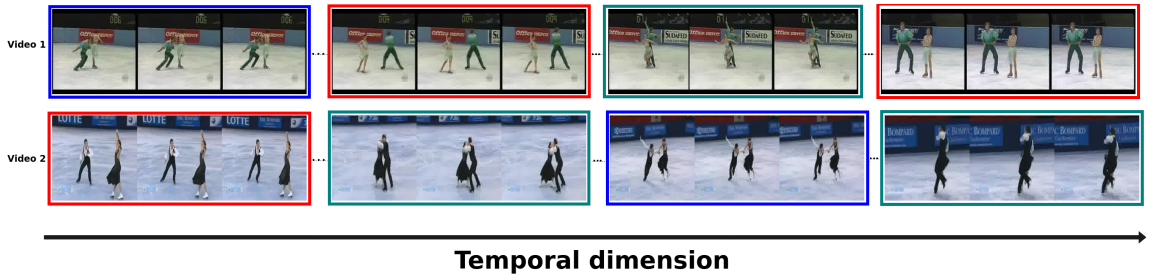


FIGURE 3.5 – Two actions belonging to the Ice-dancing category. Aligned/similar sub-actions are surrounded with red, blue and green rectangles **(Better to zoom the pdf version)**.

3.3.1 Shallow Multiple Aggregation Learning

In this section, we consider all the representations $\{\psi_{k,l}(.)\}_{k,l}$ fixed on all the video frames, and only the mixing parameters in β are allowed to vary. Given the set of action categories $\mathcal{C} = \{1, \ldots, C\}$; we train multiple classifiers (denoted $\{g_c\}_{c\in\mathcal{C}}$) on top of these level-wise representations. In practice, we use maximum margin classifiers whose kernels correspond to combinations of elementary kernels dedicated to $\{\mathcal{N}_{k,l}\}_{k,l}$. These classifiers are suitable choices as they allow us to weight the impact of nodes in the hierarchy $\mathcal{N}$ and put more emphasis on the most relevant granularity of the learned representations. Hence, depending on the granularity of action categories, these classifiers will prefer top or bottom levels of $\mathcal{N}$.

Considering a training set of videos $\{(\mathcal{V}_i, y_{ic})\}_i$ associated to an action category c, with $y_{ic} = +1$ if $\mathcal{V}_i$ belongs to the category c and $y_{ic} = -1$ otherwise, the max margin classifier associated to this action category c is given by $g_c(\mathcal{V}) = \sum_i \alpha_i^c y_{ic} \mathcal{K}(\mathcal{V}, \mathcal{V}_i) + b_c$, here b_c is a shift, $\{\alpha_i^c\}_i$ is a set of positive parameters and $\mathcal{K}$ is a positive semi-definite (p.s.d) kernel [240]. In order to combine different nodes in the hierarchy $\mathcal{N}$ and hence design appropriate aggregation, we consider multiple representation learning that generalizes [241] both to linear and nonlinear combinations. Its main idea consists in finding a kernel $\mathcal{K}$ as a combination of p.s.d elementary kernels $\{\kappa(.,.)\}$ associated to $\{\mathcal{N}_{k,l}\}_{k,l}$. Considering the two map variants in Equation 3.1, we define these kernels as

$$\begin{aligned} \mathcal{K}(\mathcal{V}, \mathcal{V}') &= \sum_l \sum_k \beta_{k,l}\ \kappa(\psi_{k,l}(\mathcal{V}), \psi_{k,l}(\mathcal{V}')) \\ \mathcal{K}(\mathcal{V}, \mathcal{V}') &= \sum_{l,l'} \sum_{k,k'} \beta_{k,l}\beta_{k',l'}\ \kappa(\psi_{k,l}(\mathcal{V}), \psi_{k',l'}(\mathcal{V}')). \end{aligned} \tag{3.2}$$

As $\beta_{k,l} \in [0, 1]$, the kernel $\mathcal{K}$ is p.s.d resulting from the closure of the p.s.d of κ w.r.t the sum and the product. Using $\mathcal{K}$, we train the max margin classifiers $\{g_c\}_{c\in\mathcal{C}}$

whose kernels (in Equation 3.2) correspond to level-wise linear (resp. cross-wise nonlinear) combinations of elementary kernels dedicated to $\{\mathcal{N}_{k,l}\}_{k,l}$. Hence, using a maximum margin formulation, we find the parameters $\beta = \{\beta_{k,l}\}_{k,l}$ and $\{\alpha_i^c\}_{i,c}$ by solving

$$\begin{array}{ll} \min\limits_{0\leq\beta\leq 1, \|\beta\|_1=1, \{\alpha_i^c\}} & \frac{1}{2}\sum\limits_c \sum\limits_{i,j} \alpha_i^c \alpha_j^c y_{ic} y_{jc} \mathcal{K}(\mathcal{V}_i, \mathcal{V}_j) - \sum\limits_i \alpha_i^c \\ \text{s.t.} & \alpha_i^c \geq 0, \quad \sum\limits_i y_{ic}\alpha_i^c = 0, \quad \forall i, c. \end{array} \tag{3.3}$$

As the problem in Equation 3.3 is not convex w.r.t β, $\{\alpha_i^c\}$ taken jointly and convex when taken separately, an EM-like iterative optimization procedure can be used : first, parameters in β are fixed and the above problem is solved w.r.t $\{\alpha_i^c\}$ using quadratic programming (QP), then $\{\alpha_i^c\}$ are fixed and the resulting problem is solved w.r.t β using either linear programming for (*) and QP for (**). This iterative process stops when the values of all these parameters remain unchanged or when it reaches a maximum number of iterations.

3.3.2 Deep Multiple Aggregation Learning

In this section, we consider an end-to-end framework that learns the parameters β_m and β_a together with (i) the ResNet parameters (denoted as α_a, α_m)[7], (ii) the MLP+softmax parameters (denoted as γ_a, γ_m) as well as (iii) the mixing parameters (referred to as $\mathbf{w}_a$ and $\mathbf{w}_m$) which respectively capture the importance of appearance and motion streams in action recognition. Considering E as the regularized cross-entropy loss[8] associated to our complete network (in Figure 3.2), we find the optimal $\alpha = \{\alpha_m, \alpha_a\}$, $\beta = \{\beta_m, \beta_a\}$, $\gamma = \{\gamma_m, \gamma_a\}$ and $\mathbf{w} = \{\mathbf{w}_m, \mathbf{w}_a\}$ by solving the following constrained minimization problem

$$\begin{array}{ll} \min\limits_{\alpha,\beta,\gamma,\mathbf{w}} & E(\alpha, \beta, \gamma, \mathbf{w}) \\ \text{s.t.} & 0 \leq \beta_{k,l}^m \leq 1, \quad \sum\limits_{k,l} \beta_{k,l}^m = 1 \\ & 0 \leq \beta_{k,l}^a \leq 1, \quad \sum\limits_{k,l} \beta_{k,l}^a = 1. \end{array} \tag{3.4}$$

In spite of having many differences w.r.t usual losses used in deep learning, this objective function can still be solved using gradient descent and backpropagation. However, many differences exist and should be carefully tackled ; indeed, whe-

7. In the rest of this chapter, the notation α refers to the ResNet parameters and not the max margin classifiers anymore.

8. Regularization is achieved using ℓ_2 weight decay.

reas the forward step can be achieved, gradient backpropagation (through our multiple aggregation layer) should be achieved while considering videos with a varying number of frames. Besides, constraints on $\beta's$ should also be handled. In what follows, we discuss all these updates in the optimization process.

Optimization. Considering $\rho()$ as the output of the final layer of our deep network and considering $\frac{\partial E}{\partial \rho}$ available, the gradients $\frac{\partial E}{\partial \mathbf{w}}$, $\frac{\partial E}{\partial \gamma}$ (w.r.t the preceding mixing and MLP layers) could easily be obtained using a straightforward application of the chain rule (as already available in the used PyTorch tool). However, $\frac{\partial E}{\partial \beta}$, $\frac{\partial E}{\partial \alpha}$ cannot be obtained straightforwardly ; on the one hand, any step following the gradient $\frac{\partial E}{\partial \beta}$ should preserve equality and inequality constraints in Equation 3.4 while a direct application of the chain rule provides us with a surrogate gradient which ignores these constraints. On the other hand, the variable number of frames for different training videos requires a careful update of $\frac{\partial E}{\partial \alpha}$ as shown subsequently.

Constraint implementation. In order to implement the equality and inequality constraints during the optimization of the objective function in Equation 3.4, we consider a re-parametrization as $\beta^m_{k,l} = h(\hat{\beta}^m_{k,l})/\sum_{k',l'} h(\hat{\beta}^m_{k',l'})$ for some $\{\hat{\beta}^m_{k,l}\}_{k,l}$ with h being strictly monotonic positive function and this allows free settings of the parameters $\{\hat{\beta}^m_{k,l}\}_{k,l}$ during optimization while guaranteeing $\beta^m_{k,l} \in [0,1]$ and $\sum_{k,l} \beta^m_{k,l} = 1$. During back-propagation, the gradient of the loss E (now w.r.t $\hat{\beta}$'s) is updated using the chain rule as

$$\begin{aligned} \frac{\partial E}{\partial \hat{\beta}^m_{k,l}} &= \sum_{p,q} \frac{\partial E}{\partial \beta^m_{p,q}} . \frac{\partial \beta^m_{p,q}}{\partial \hat{\beta}^m_{k,l}} \\ \text{with} \quad \frac{\partial \beta^m_{p,q}}{\partial \hat{\beta}^m_{k,l}} &= \frac{h'(\hat{\beta}^m_{k,l})}{\sum_{k',l'} h(\hat{\beta}^m_{k',l'})} . (\delta_{p,q,k,l} - \beta^m_{p,q}), \end{aligned} \tag{3.5}$$

and $\delta_{p,q,k,l} = 1_{\{(p,q)=(k,l)\}}$. In practice $h(.) = \exp(.)$ and $\frac{\partial E}{\partial \beta^m_{p,q}}$ is obtained from layer-wise gradient backpropagation (as already integrated in standard deep learning tools including PyTorch). Hence, $\frac{\partial E}{\partial \hat{\beta}^m_{k,l}}$ is obtained by multiplying the original gradient $\left[\frac{\partial E}{\partial \beta^m_{p,q}}\right]_{p,q}$ by the Jacobian $\left[\frac{\partial \beta^m_{p,q}}{\partial \hat{\beta}^m_{k,l}}\right]_{p,q,k,l}$ which simply reduces to $\left[\beta^m_{k,l}(\delta_{p,q,k,l} - \beta^m_{p,q})\right]_{p,q,k,l}$ when $h(.) = \exp(.)$. Similarly, we implement the constraints associated to the appearance stream.

ResNet update. As discussed earlier, motion and appearance ResNets are *recurrently* (iteratively) applied frame-wise prior to pool the underlying feature maps using multiple aggregation. It is clear that the number of frames intervening in

this aggregation is video-dependent, and thereby the number of terms in these aggregations (and the number of ResNet branches/instances) is also varying. Hence, a straightforward application of the chain rule in the whole architecture – in order to update $\frac{\partial E}{\partial \alpha}$ – becomes possible only when this architecture is unfolded, and this requires fixing the maximum number of frames (denoted as T) and sampling temporally all the videos in order to make T_i constant and equal to T. Note that beside requiring all the ResNet instances to share the same parameters (as in Siamese nets), this results into a cumbersome architecture even for reasonable T values. Furthermore, frame sampling requires interpolation techniques which are highly dependent on quality, duration and temporal resolution of videos and this may result into spurious motion/appearance details (especially on short videos; even when timely well resolute) which ultimately leads to a significant drop in action recognition performances.

In order to avoid these drawbacks and to fully benefit from the available number (and also temporal resolution) of frames — without using multiple instances of "Siamese-like" ResNets and without resampling — we consider an alternative gradient estimation. The latter relies on a membership measure μ which assigns each frame $f_{i,t}$ to nodes in the temporal pyramid as $\mu_{i,t}^{k,l} = 1_{\{t \in \mathcal{N}_{k,l}\}}$. Using this membership measure together with the chain rule, the gradient of the loss E w.r.t the parameters of the ResNet α can be updated as

$$\frac{\partial E}{\partial \alpha_m} = \sum_{k,l} \sum_{i,t} \mu_{i,t}^{k,l} \frac{\partial E}{\partial \psi_{k,l}^m} \frac{\partial \psi_{k,l}^m}{\partial \phi^m(f_{i,t})} \frac{\partial \phi^m(f_{i,t})}{\partial \alpha_m}. \tag{3.6}$$

Similarly, we evaluate the gradient for the appearance stream. From the above equation, it is clear that when $k = l = 1$, all the frames $\{f_{i,t}\}_{i,t}$ contribute in the estimation of the gradient, while for other nodes, only a subsets of frames (belonging to these nodes) are used. Nonetheless, all the frames contribute evenly through all the nodes and hence in gradient estimate, without any sampling. Note also that this formulation implicitly implements *weight sharing* as the above gradient can equivalently be written as the sum of gradients, shared through multiple streams of an unfolded architecture, with each stream being dedicated to one frame. However, the advantage of the above formulation resides again in its computational efficiency and also its ability to leverage all (possibly variable numbers of) frames in videos while an unfolded architecture requires sampling a fixed number of frames and handling multiple ResNet branches which may clearly lead to intractable training.

3.4 Experiments

In this section, we evaluate the impact of our multiple aggregation design on the performance of action recognition and we compare it against other aggregation strategies as well as the related work using three standard datasets : UCF-101 [197], HMDB-51 [198] and JHMDB-21 [199]. UCF-101 — used to comprehensively study the different settings of our model — is the largest and most challenging ; it includes 13,320 video shots belonging to 101 categories with variable duration, poor frame resolution, viewpoint and illumination changes, occlusion, cluttered background and eclectic content ranging from multiple and highly interacting individuals to single and completely passive ones. We also consider HMDB-51 and JHMDB-21 for further comparisons ; the latter include 6766 (resp. 928) videos belonging to 51 (resp. 21) action categories. The particularity of these three datasets also resides in the fact that actions are misaligned as their videos are endowed with large context while others are precisely trimmed and contain only the actions of interest (see the example of misalignments in Figure 3.5) ; for more details about the datasets, see again Section 2.5. In all these experiments, we process all the videos using ResNet-101 (as a backbone network [9]) in order to extract all the underlying appearance and motion representations framewise. Then, we apply different aggregation schemes prior to assign those videos to classes. We use the same evaluation protocols as the ones suggested in [197-199] (i.e., train/test splits) and we report the average accuracy over all the categories of actions.

The purpose of our evaluation is to show the performance of the hierarchical aggregation design of our temporal pyramid compared to different coarse and fine aggregations as well as other baselines. We also extend the comparison of action classification against reported results in the related work. Different settings are considered in order to assess the performance of our method : i) multiple depths of our hierarchical aggregation network, ii) two streams (motion and appearance) as well as their fusion, and iii) the two types of aggregations namely "concatenation" and "averaging".

We train our complete temporal pyramid-based networks (in Figure 3.2) for respectively 130, 100 and 65 iterations on UCF-101, HMDB-51 and JHMDB-21 using the PyTorch SGD optimizer. For appearance stream, we set the learning rate to 0.001 and reduce it by a factor of 10 every 25, 20, 10 iterations for resp. UCF-101, HMDB-51 and JHMDB-21. For motion stream, we set the learning rate to 0.005 and we reduce it by the same factor after “80 and 110”, “60 and 80”,

9. ResNet-101 is trained on ImageNet and then fine-tuned on UCF-101. We also build appearance representations based on ResNet-152 trained only on ImageNet.

Deep convolutional networks	UCF-101	# parameters (millions)
Pretrained AlexNet [14]	58.14	61M
Pretrained VGGNet11 [161]	63.12	132M
Pretrained VGGNet19 [161]	63.42	143M
Pretrained ResNet18 [16]	68.32	11M
Pretrained ResNet50 [16]	68.39	25M
Pretrained ResNet101 [16]	68.47	44M
Pretrained ResNet152 [16]	68.58	60M

TABLE 3.1 – Action classification performances using the temporal pyramid described in Section 3.3.1 (based on concatenation (*). See Equation 3.2) combined with different deep network architectures pretrained with ImageNet (these networks were initially designed to extract appearance features).

"50 and 60" iterations on the three sets respectively. Experiments on individual streams are run using 4 Titan X Pascal GPUs (with 12 Gb) and last 72h for UCF101, 36h for HMDB-51 and 15h for JHMDB-21 (on the appearance stream) and 96h for UCF101, 48h for HMDB-51 and 24h for JHMDB-21 (on the motion stream) while on the joint stream experiments are run using 4 Tesla P100 GPUs (with 16 Gb) and last 100h, 55h and 30h on the three sets respectively.

3.4.1 Convolutional Network Selection

The choice of the initial pretrained backbone convolutional network – that operates at the frame-level — should consider two factors ; its baseline classification performances and the number of training parameters. The latter is particularly crucial for action recognition as the size of training data is limited compared to other neighboring tasks (such as image classification) on which these convolutional networks were initially trained. Hence, in order to select the most appropriate convnet among a collection of existing ones (namely [14, 16, 161]), we measure the performance of our temporal pyramid based on the design in [227]. The results in Table 3.1 show that the deeper the network, the better are the performances. However, in our experiments, we consider *ResNet-101*, which provides descent action recognition performances while being relatively less memory and time demanding compared to the other networks and particularly *ResNet-152* (see again Table 3.1).

Motion stream UCF-101	Shallow design concatenation	Shallow design averaging	Deep design concatenation	Deep design averaging
TP (level 1)	78.40	78.40	**78.66**	78.66
TP (level 2)	79.53	79.54	**79.86**	79.76
TP (level 3)	79.70	79.71	**79.93**	79.83
TP (level 4)	79.76	79.77	**81.14**	80.66
TP (level 5)	80.23	80.24	**81.43**	80.84
TP (level 6)	79.96	79.98	**81.69**	80.12

TABLE 3.2 – This table shows level-wise performances using the motion stream both for shallow and deep models. These performances are reported both for "averaging" and "concatenation". In these initial experiments – in order to compare the performances of shallow and deep designs under comparable conditions – we fine-tune only the last fully connected layer of *ResNet-101* along with the parameters of the temporal pyramid (TP).

3.4.2 Performances

Firstly, we show a comparison of action recognition performances using different settings. Extensive experiments, reported in Table 3.2 and Table 3.3, show that our hierarchical aggregation design makes it possible to select the best configuration (combination) of level representations in order to improve the performance of classification ; indeed, the results show a clear gain as the depth of the hierarchy increases and compared to global average pooling (level 1). This gain results from the match between the temporal granularity of the learned level-wise representations in the hierarchy and the actual granularity of action categories. Note that in all these performances, multi-level node concatenation provides a clear gain compared to averaging, especially on deeper levels of the temporal pyramid, both on motion and appearance streams. The rational is that multi-level node concatenation preserves better the temporal granularity of actions compared to averaging. Hence, in the remainder of these experiments, we keep concatenation when learning "end-to-end" joint combination of appearance and motion streams.

Secondly, we compare the performance of the two settings (shallow and deep) of our multiple aggregation design using both motion and appearance streams taken individually and combined ; as already discussed, the parameters $\mathbf{w}_a$, $\mathbf{w}_m$ of this fusion are optimized as a part of the end-to-end learning process. Results reported in Table 3.4 show the complementary aspects of the two streams in all the settings as their fusion brings a clear gain in performance. Moreover, we observe that the contribution of the motion stream is strictly increasing (and *a contrario*

Appear stream	Shallow design		Deep design	
UCF-101	concatenation	averaging	concatenation	averaging
TP (level 1)	80.28	80.28	**80.31**	80.31
TP (level 2)	81.77	81.78	82.16	**82.21**
TP (level 3)	82.17	82.17	82.74	**82.89**
TP (level 4)	82.51	82.50	**83.52**	83.38
TP (level 5)	82.50	82.51	**83.63**	80.83
TP (level 6)	81.96	81.96	**83.92**	80.83

TABLE 3.3 – This table shows level-wise performances using the appearance stream both for shallow and deep models. These performances are reported both for "averaging" and "concatenation". In these initial experiments – in order to compare the performances of shallow and deep designs under comparable conditions – we fine-tune only the last fully connected layer of *ResNet-101* along with the parameters of the temporal pyramid.

Fusion	Shallow design (concat)			Deep design (concat)			Stream importance	
UCF-101	Motion	Appear	Joint	Motion	Appear	Joint	$\mathbf{w}_m$	$\mathbf{w}_a$
TP (level 1)	78.40	80.28	88.91	78.74	80.69	89.69	0.46	**0.54**
TP (level 2)	79.53	81.77	89.10	79.97	82.78	90.00	0.49	**0.51**
TP (level 3)	79.70	82.17	89.34	80.69	83.12	90.26	**0.52**	0.48
TP (level 4)	79.76	82.51	89.37	81.74	83.78	90.92	**0.52**	0.48
TP (level 5)	80.23	82.50	84.49	82.86	84.10	91.45	**0.56**	0.44
TP (level 6)	79.96	81.96	89.26	83.41	84.92	**92.37**	**0.60**	0.40

TABLE 3.4 – This table shows level-wise performances of joint (2-stream) fusion for both shallow and deep methods. These results are shown only for "concatenation" as the underlying baseline performances reported in Table 3.2 and Table 3.3 are better than "averaging". In contrast to Table 3.2 and Table 3.3, all the parameters of the whole network (including ResNet) are allowed to vary.

strictly decreasing for appearance stream) as the level of the temporal pyramid increases (see the distribution of $\mathbf{w}$ in Table 3.4). This clearly corroborates the highest impact of motion (compared to appearance) when modeling the temporal granularity of action categories (see later Figure 3.6). We also observe a higher positive impact on performances as the depth of our temporal pyramids increases; again, these results are obtained using "concatenation" instead of "averaging", as the former already globally overtakes the latter on motion and appearance streams when taken individually (see again Table 3.2 and Table 3.3).

We further investigate the potential of our method using multiple instances of temporal pyramids both for motion and appearance streams as well as their joint fusion. The rational – from this setting – resides in the heterogeneity of action

# of temporal pyramids per stream	Accuracy (concatenation) Appearance stream	Motion stream	Joint stream
1	83.92	81.69	90.78
2	83.95	81.73	90.79
4	**83.97**	81.79	90.84
8	83.92	**81.86**	**90.89**
16	83.89	81.83	90.85

TABLE 3.5 – This table shows the evolution of the performances w.r.t different # of temporal pyramids per stream. In order to combine the outputs of these multiple pyramids (when using concatenation), we add a succession of FC+ReLU+BatchNorm to reduce the dimensionality **from** "63 (number of nodes in TP of 6 levels) $\times$ 128 (node dimension) $\times$ # TPs" **to** "128". All these results correspond to temporal pyramids of 6 levels.

categories and their dynamics which may affect the accuracy ; indeed, the apex of some actions appears early in video clips while for others later or spread through all the video duration. Hence, instead of learning a single monolithic temporal pyramid per stream, we stack multiple instances of temporal pyramids with different weights β, each one dedicated to a subclass of actions whose dynamics (not category) are similar [10]. We learn the parameters of these pyramids "end-to-end" as discussed earlier for single pyramids. Table 3.5 shows the performances w.r.t the number of pyramids. In spite of an increase of the number of training parameters in these multiple pyramids (without any increase of training data), we observe an improvement ; we believe that adding extra training data will bring a further and clearer gain in performances.

3.4.3 Sampling, Surrogate Gradient and Efficiency

Table 3.6 shows the impact of our method – with and without frame sampling – on the performance of action recognition. These results are obtained using a single pyramid. From these results, it is easy to see that performances get better as the number of sampled frames increases reaching asymptotically the best performances when all the frames are used. This behavior is similar both on motion and appearance streams. However, we notice that motion stream which is based on optical flow data is more sensitive to sampling than appearance stream so the accuracy of the former is clearly proportional to the number of frames. Put differently, motion stream builds a better representation and hence becomes more

10. These subclasses of actions are not explicitly defined in a supervised manner but implicitly by allowing enough flexibility in the multiple instances of temporal pyramids in order to capture different (unknown) subclasses of action dynamics.

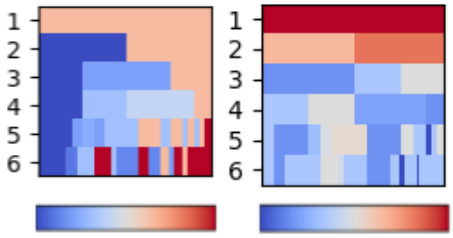
Sampling strategies	# frames (train) RGB	OF	# frames (test) RGB	OF	Accuracy Appearance	Motion	Fusion
#1	25	25	25	25	84.23	81.27	91.65
#2	25	25	25	250	84.23	81.27	91.64
#3	25	50	25	50	84.23	81.86	91.69
#4	25	50	25	250	84.23	81.89	91.78
#5	64	64	250	250	84.62	82.05	91.89
#6	64	64	all	all	84.81	82.77	92.09
#7	64	all	all	all	84.81	83.41	92.29
#8	all	all	all	all	84.92	83.41	92.37

TABLE 3.6 – This table shows the evolution of the performance w.r.t to different sampling strategies (i.e., number of frames in training and test videos). RGB and OF stand for the number of input RGB frames and the number of optical flow frames used in the appearance and the motion streams respectively. These performances are obtained using a temporal pyramid of six levels.

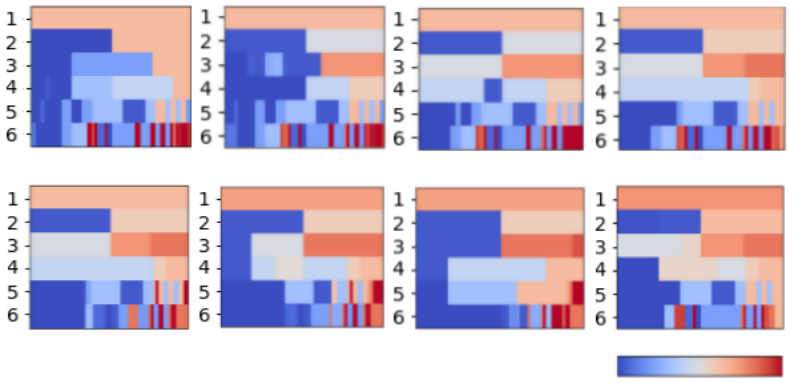
important for the overall action classification when it is fed with more optical flow data as shown again in Table 3.6 (settings #6 and #7). However, taking all the frames during backpropagation, comes at the expense of a substantial increase of computation; when considering all the 2.5 millions frames of our videos on UCF-101, training costs 72h (resp. 96h) for appearance (resp. motion) stream using 4 Titan X GPUs (with 12 Gb) and 100h on the joint stream using 4 Tesla P100 GPUs (with 16 Gb). This high cost results from the large number of visited frames when (re)estimating the gradient, in Equation 3.6 w.r.t the parameters of the ResNet, through the epochs of backpropagation. In order to make the evaluation of Equation 3.6 (and hence training) more tractable (with a controlled loss in classification performances), we consider a surrogate gradient defined as

$$\frac{\partial E}{\partial \alpha_m} = \sum_{k,l,i} \sum_{t \in \mathcal{P}_r^i} \mu_{i,t}^{k,l} \frac{\partial E}{\partial \psi_{k,l}^m} \frac{\partial \psi_{k,l}^m}{\partial \phi^m(f_{i,t})} \frac{\partial \phi^m(f_{i,t})}{\partial \alpha_m}, \tag{3.7}$$

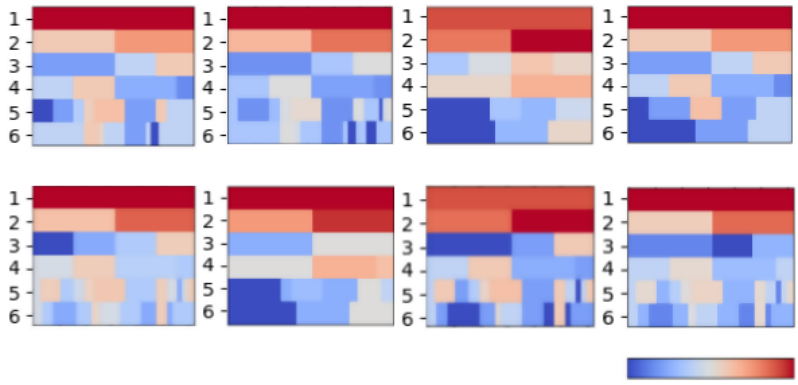
here $\mathcal{P}_r^i$ stands for a subset of selected frame time-stamps, in a given video $\mathcal{V}_i$, that contribute to gradient estimation at the r^{th} epoch. We consider a periodic selection mechanism which guarantees that all the frames are evenly used through epochs; in practice, $\mathcal{P}_r^i = \{t \in [0, T_i],\ t \equiv r \pmod K\}$ with $1/K$ being the fraction of frames used per epoch. With this mechanism, gradient evaluation still relies on the entire set of frames in the training set, but their use is distributed through epochs and this makes the evaluation and training process far more efficient while maintaining close performances (see Table 3.7). For instance, when $K = 24$, training is $24\times$ faster compared to the most accurate setting (strategy #8 in Table 3.6) as only 8 frames are used (on average "per epoch-per video") in Equation 3.7 instead of 185; furthermore, since all the frames contribute equally through all



(a) Single temporal pyramid



(b) Multiple temporal pyramids (motion stream)



(c) Multiple temporal pyramids (appearance stream)

FIGURE 3.6 – (a) Weight distribution of motion and appearance streams obtained when learning the parameters of a single temporal pyramid (corresponding to the first row of Table 3.5). (b-c) Weight distribution of multiple temporal pyramids of motion and appearance streams (corresponding to the fourth row in the same table). Warmer colors correspond to higher weights while cooler colors to lower ones.

the epochs, the loss in accuracy is contained. These performances are obtained on individual and joint streams using the same aforementioned hardware resources.

Speed up factor (K)	Avg. # frames per "epoch and training video"	Accuracy		
		Appearance	Motion	Joint
1×	185	84.92	83.41	92.37
4×	92	84.27	82.59	91.74
8×	46	84.10	82.07	91.39
16×	23	83.96	81.23	90.70
24×	8	83.89	80.95	90.35

TABLE 3.7 – This table shows the performance of "surrogate back-propagation" with different acceleration factors. Note that motion stream performances are more sensitive to this acceleration compared to appearance stream.

3.4.4 Comparison Against Related Work

Finally, we compare the performance and the complementary aspects of our method against related state of the art action recognition methods [16, 28, 195, 227, 238] on UCF-101, HMDB-51 and JHMDB-21. The closely related methods in [227, 230] are based on deep framewise representations which are aggregated and classified using a hierarchy of multiple temporal granularities. However, this method differs from the one proposed, in [227], in different aspects : first, framewise representations are extracted using ResNet-152 pretrained only on ImageNet and not fine-tuned on UCF-101. Besides, the method in [227] is based only on appearance stream and more importantly, the design principle of the proposed method in Section 3.3.2 is deep and consists in weighting the contribution of each level in the temporal pyramid as a part of an "end-to-end" learning process while in [227, 230] this weighting scheme is relatively shallow and excludes the ResNet from training. Note that the variant in [230] – referred to as Deep Multiple Kernel Learning (DMKL)) – relies on a contrastive loss design that makes training more efficient and also still effective compared to the EM-like procedure in [227] ; however, ResNet is also excluded from training in [230]. All these differences explain the significant under-performances of [227, 230] compared to our "end-to-end" framework (framewise representations obtained with ResNet trained on ImageNet and then fine-tuned on UCF-101) as observed in Table 3.8.

Extra comparisons in Table 3.8 also include global averaging techniques as well as spectrogram-like representations. The former produces a global representation that averages all the frame representations while the latter keeps all the frame representations and concatenate them prior to their classifications (see Figure 3.7). Note that these two settings are related to the two extreme cases of our hierarchy, i.e., the root and the leaves. In particular, the spectrogram of a video $\mathcal{V}$ with T frames is obtained when the number of leaf nodes, in the hierarchy, is exactly

Methods	UCF-101	HMDB-51	JHMDB-21	Batch size	# frames (RGB,OF)	ImageNet pretrain	Kinetics pretrain
2D colorized heatmaps [195]	64.38	54.90	60.5	32	(all,all)	✗	✗
2D motion + GAP [238]	79.4	59.13	61.39	32	(none,64)	✓	✗
2D appearance + GAP [238]	82.1	60.24	62.71	32	(3,none)	✓	✗
2D 2-streams + GAP [238]	88.5	63.31	64.11	32	(3,64)	✓	✗
3D motion [28]	96.41	80.39	✗	15	(none,64)	✓	✓
3D appearance [28]	95.60	76.47	✗	15	(64,none)	✓	✓
3D two-streams [28]	97.94	80.65	✗	15	(64,64)	✓	✓
GAP-A of [227] (on ResNet152 [16])	66.15	✗	✗	✗	(all,none)	✓	✗
GAP-A of [230]	81.14	✗	✗	✗	(all,none)	✓	✗
GAP-M of [230]	79.10	✗	✗	✗	(none,all)	✓	✗
GAP-2S of [230]	89.16	✗	✗	✗	(all,all)	✓	✗
Spect-A (on ResNet152+ResNet18 [16])	64.41	54.85	60.61	32	(all,all)	✓	✗
Spect-A (on ResNet101+ResNet18 [16])	78.40	57.76	61.26	32	(all,all)	✓	✗
Spect-M (on ResNet101+Resnet18 [16])	76.46	55.38	60.66	32	(all,all)	✓	✗
Spect-2S (on ResNet101+Resnet18 [16])	80.10	58.28	62.14	32	(all,all)	✓	✗
TP-A (EM) of [227] (on ResNet152 [16])	68.58	58.63	62.16	✗	(all,none)	✓	✗
TP-A (EM) of [230]	83.36	✗	✗	✗	(all,none)	✓	✗
TP-M (EM) of [230]	81.07	✗	✗	✗	(none,all)	✓	✗
TP-2S (EM) of [230]	89.91	✗	✗	✗	(all,all)	✓	✗
TP-A (DMKL) of [230]	83.44	✗	✗	✗	(all,none)	✓	✗
TP-M (DMKL) of [230]	81.17	✗	✗	✗	(none,all)	✓	✗
TP-2S (DMKL) of [230]	89.95	✗	✗	✗	(all,all)	✓	✗
Our "2D appearance + TP"	84.92	62.23	63.51	1	(all,all)	✓	✗
Our "2D motion + TP"	83.41	61.04	62.97	1	(all,all)	✓	✗
Our "2D two-streams + TP"	92.37	65.14	66.96	1	(all,all)	✓	✗
2D col-heatM[195] + our "2D motion + TP"	80.41	65.21	69.93	✗	✗	✗	✗
3D motion[28] + our "2D motion + TP"	96.61	80.54	✗	✗	✗	✗	✗
3D appear[28] + our "2D appear + TP"	96.05	76.56	✗	✗	✗	✗	✗

TABLE 3.8 – This table shows a comparison of our temporal pyramid (TP) w.r.t different related works; in this table, "col-heatM" stands for colorized heatmaps, "Spect" for spectrograms, "A" for appearance, "M" for motion, "2S" for two-streams, "GAP" for global averge pooling and "OF" for optical flow. In our experiments, (i) *ResNet-152* is pretrained on ImageNet, (ii) *ResNet-101* is pretrained on ImageNet and fine-tuned on UCF-101 (for both appearance and motion) and (iii) ResNet18 is pretrained on ImageNet and fine-tuned on UCF-101 (again for appearance and motion). In these results, the symbol "✗" stands for "a method does not apply or was not applied (results not available)" in the underlying works.

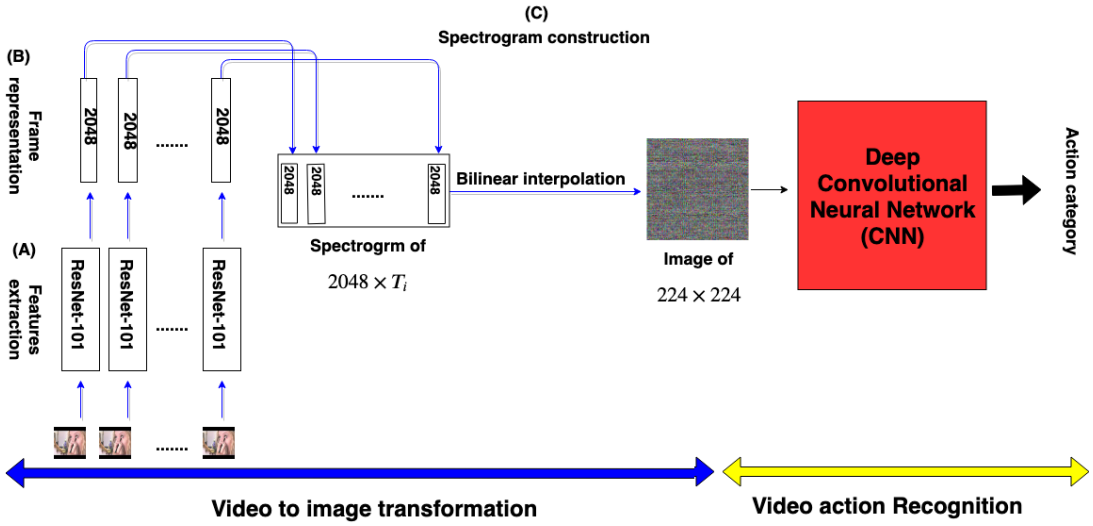


FIGURE 3.7 – Different steps of spectrogram construction.

equal to T. Global averaging techniques (shown in Table 3.8) include [195]; the latter is based on colorized heatmaps and corresponds to timely-stamped and averaged framewise probability distributions of human keypoints. These colorized heatmaps are fed to a 2D CNN for classification; note that colorized heatmaps provide video-level representations which capture globally the dynamics of video actions without any scheme to emphasize the most important temporal granularities of these actions and this results into low accuracy as again displayed in Table 3.8.

The last category of methods (shown in Table 3.8) includes convolutional networks based on 2D and 3D spatio-temporal filters [28, 238]. These methods are based either on one or two streams; one for motion and another one for appearance followed by a global average pooling. Both methods are similar to ours; they combine motion and appearance streams and their design is end-to-end but clearly differ in their pooling mechanisms and the way frames are exploited. Indeed, these related techniques rely on sampling strategies that vectorize video sequences into fixed length inputs while our method keeps all the frames in order to build temporal pyramids. Another major difference w.r.t our method resides in the huge set used in order to train these related architectures. Nevertheless, while these streams are highly effective their combination with our hierarchical aggregation, through a late fusion [11], brings a noticeable gain in performances. We also observe the same behavior on all the combinations of our two stream

11. Late fusion is applied (instead of early one) as our video inputs are different from those of 2D colorized heatmaps and convolutional 3D filters which are spatio-temporal while ours are only spatial. We also exclude, from fusion, 2D methods+GAP as they correspond to a particular setting of our method (namely temporal pyramid of level 1).

model with other baselines and other related methods (including two stream 3D CNNs [28] and spectrograms [227]); indeed, from the results shown inTable 3.8, our hierarchical method brings a clear gain w.r.t most of these methods. Note that some of these models rely on extra datasets (including Kinetics) in order to pretrain their CNNs while our method is trained only on the original datasets.

3.5 Conclusion

We introduce in this chapter a temporal pyramid approach for video action recognition. The strength of the proposed method resides in its ability to learn hierarchical pooling operations that capture different levels of temporal granularity in action recognition. This is translated into learning the distribution of weights in the temporal pyramid, that capture these granularities, by solving constrained minimization problems. Two settings are considered : shallow and deep. The former relies on solving a constrained quadratic programming problem while the latter on optimizing the parameters of a deep network including a temporal pyramid module both on motion and appearance streams as well as their combination. We also consider variants of the deep learning framework that design multiple instances of temporal pyramids each one dedicated to a particular subcategory of action granularities and also a procedure that allows us to efficiently train the network at the detriment of a slight decrease of its classification accuracy. The advantages of these contributions are established, against different baselines as well as the related work, through extensive experiments on challenging action recognition benchmarks including UCF-101, HMDB-51 and JHMDB-21 datasets.

ConvNets, including those developed in this chapter, are designed to operate on vectorial data including image sequences and videos. ConvNets rely on the assumption that videos come in the form of 2D/3D regular grids. However, actions in videos can be seen as constellations of interacting body parts which can be represented in the form of skeletons (or graphs) where their joints (nodes) describe the body parts and edges their interactions resulting into an explicit encoding of the geometric structure of video actions. As ConvNets cannot directly operate on skeletons, they require a careful update of convolution and pooling operations on graphs, and this will be investigated in the subsequent chapter.

CHAPITRE 4

SPECTRAL GRAPH CONVOLUTIONAL NEURAL NETWORKS FOR ACTION RECOGNITION

Contents

Chapitre abstract

Convolutional neural networks (ConvNets) are nowadays witnessing a major success in different pattern recognition problems. These learning models were basically designed to handle vectorial data such as images but their extension to non-vectorial and semi-structured data (namely graphs with variable sizes, topology, etc.) remains a major challenge, though a few interesting solutions are currently emerging.

In this work, we introduce MLGCN ; a novel spectral Multi-Laplacian Graph Convolutional Network. The main contribution of this method resides in a new design principle that learns graph-Laplacians as

convex combinations of other elementary Laplacians, each one dedicated to a particular topology of the input graphs.
Moreover, we generalize this MLGCN to tree-structured temporal pyramids referred to TP-MLGCN. The latter captures different levels of granularity in the learned classes. It is inspired by the work in *Chapter 3* and by inception network [104] to design effective convolutional operators on graphs.

We also introduce a novel pooling operator, on graphs, that proceeds in two steps : context-dependent node expansion is achieved, followed by a global average pooling ; the strength of this two-step process resides in its ability to preserve the discrimination power of nodes while achieving permutation invariance.

Experiments conducted on JHMDB (2D skeletons), SBU (3D skeletons) and UCF-101 (video rgb frames) datasets, show the validity of our methods for the challenging task of action recognition.

The work in this chapter has led to the publication of a conference paper :

- Ahmed Mazari and Hichem Sahbi. MLGCN : Multi-Laplacian Graph Convolutional Networks for Human Action Recognition. In the 30th British Machine Vision Conference (BMVC). Cardiff, Wales, United Kingdom, 2019

4.1 Introduction and Related Works

Video action recognition is a major task in computer vision which consists in classifying sequences of frames into categories (or classes) of actions. This task is known to be challenging due to the intrinsic properties (appearance and motion) of moving objects and also their extrinsic acquisition conditions (occlusions, background clutter, camera motion, illumination, length/resolution, etc.). Most of the existing action recognition methods are based on machine learning [17, 19, 21, 22, 26, 123, 126, 152] ; their general recipe consists in extracting (handcrafted or learned) features prior to classifying them using inference techniques such as kernel methods and deep networks [15, 28, 175, 176, 178-180, 195, 242].

Among the machine learning techniques for action recognition those based on deep networks are particularly performant ; successful methods include two-

stream 2D convolutional neural networks (CNNs) [15, 176], two-stream 3D CNNs and simple 3D CNNs [28]. However, and beside being data-hungry, these models rely on a strong assumption that videos are described as vectorial data ; in other words, these methods assume that videos come only in the form of regular (2D or 3D) grids. This assumption may not hold in practice : on the one hand, one may consider moving objects as constellations of interacting body parts (such as 2D/3D skeletons or joints in human actions) and this requires processing only these joints without taking into account holistically cluttered background or other parts in the scenes. On the other hand, moving objects may be occluded with spurious details which are not necessarily related to the moving object parts. Hence, for these particular settings, graph convolutional networks (GCNs) [243] are rather more appropriate where nodes, in these models, capture object parts and links their spatio-temporal interactions.

Early GCNs are targeted to graphs with known/fixed topology [1] (fixed number of nodes/edges, constant degree, etc.) [62, 99] ; in existing solutions pixels are considered as nodes and edges connect neighboring pixels. Despite their relative success for some pattern classification tasks including optical character recognition (on widely used benchmarks such as MNIST), these methods do not straightforwardly extend to general graphs with arbitrary topological characteristics (variable number of nodes/edges, heterogeneous degrees, etc.) and this limits their applicability to other challenging tasks such as action recognition.

Recent attempts, to extend these methods to action recognition [103, 244, 245], include [103] which models connectivity of moving joints in videos using graphs where nodes correspond to joints (described by spatial coordinates and their likelihoods) and edges characterize their spatio-temporal interactions. One of the drawbacks of these extensions resides in the limited representational power of joints and also the difficulty in achieving permutation invariance ; in other words, parsing and describing joints while being invariant to arbitrary reordering of objects especially for highly complex scenes with multiple interacting objects/persons.

From the machine learning point of view, GCN operates either directly in the spatial domain [101, 242, 246-256] or require a preliminary step of spectral decomposition of graphs using Fourier basis [60, 100, 257, 258] prior to achieve convolution [61-63, 99, 102, 259-262]. While graph convolution in the spectral domain is well defined, its success heavily relies on the choice of the Laplacian operators [263] that capture the topology of the manifolds enclosing data. These

1. as 2D regular grids

Laplacians, in turn, depend on many hyper-parameters which are difficult to set using tedious cross-validation especially when training GCNs on large-scale datasets.

In this chapter, we address the aforementioned issues (mainly Laplacian design in GCNs and permutation invariance) for the particular task of action recognition. Our solution achieves convolution in the spectral domain using a new design principle that considers a convex combination of several Laplacian operators; each Laplacian is dedicated to a particular (possible) topology of our graphs. We also introduce a novel context-dependent pooling operator that proceeds in two steps : node features are first expanded with their contexts and then globally averaged; the strength of this two-step pooling process resides in its ability to preserve/enhance the discrimination power of node representations while achieving permutation invariance. The validity of these contributions is corroborated through extensive experiments, in action recognition, using the challenging SBU-skeleton and UCF-101 datasets.

4.2 Graphs Construction

In this section, we describe the video processing used to build our graph inputs for sequences of skeleton joints and sequences of rgb frames. Note that skeleton joint features are based on their 2D/3D coordinates while those of rgb frames are based on the convolutional representations of raw frames.
This processing step consists in extracting and grouping joints (a.k.a keypoints) into trajectories prior to modeling their spatio-temporal interactions with graphs.

Given a raw video, skeletons are obtained by detecting human joints in successive frames using the state of the art human pose extractor[2] [194]; as these keypoints are labeled (see Figure 4.1), their trajectories are extracted by simply tracking keypoints with the same labels.

Considering a finite collection of trajectories, we build an adjacency graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where each node $v \in \mathcal{V}$ corresponds to a labeled trajectory and an edge $(v, v') \in \mathcal{E}$ exists between two nodes iff the underlying trajectories are spatially neighbors. Each trajectory (i.e., node in $\mathcal{G}$) is described by aggregating motion and appearance streams as shown subsequently.

2. This processing is only reserved to raw video datasets (including UCF [197]) while for other databases, such as SBU [264] and JHMDB [199], skeletons are already available.

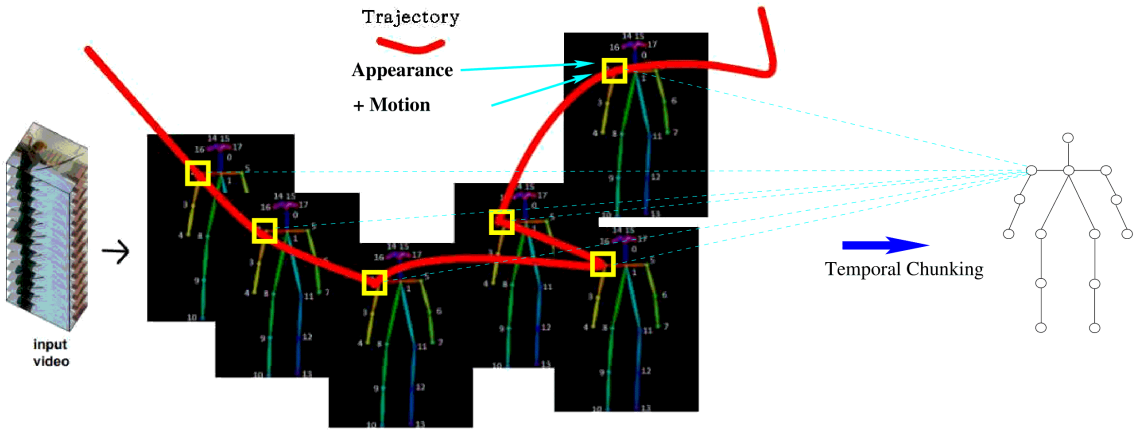


FIGURE 4.1 – This figure shows the whole keypoint extraction, tracking and description process on motion and appearance streams

4.2.1 Graph Representation for Skeleton Joints

Considering a sequence of skeletons, we process the underlying trajectories using *temporal chunking* : first we split the total duration of a video into C equally-sized temporal chunks ($C = 4$ in practice), then we assign the keypoint coordinates of a given trajectory v to the C chunks (depending on their time stamps) prior to concatenate the averages of these chunks and this produces the description of v denoted as $\psi(v)$. The whole process is displayed in Figure 4.1
Trajectories with similar keypoint coordinates but arranged differently in time, will be considered as very different. Note that beside being compact and discriminant (as shown later in Table 4.11), this temporal chunking gathers advantages while discarding drawbacks of two widely used families of techniques mainly *global averaging techniques* (invariant but less discriminant) and *frame re-sampling techniques* (discriminant but less invariant). Put differently, *temporal chunking* produces discriminant descriptions that preserve the temporal structure of trajectories while being *frame-rate* and *duration* agnostic.

4.2.2 Graph Representation for RGB Frames

In contrast to skeleton data where 2D/3D joint coordinates through video sequences are provided, skeleton joints on rgb frames are not available. They are estimated before aggregating their trajectories in order to build a graph representation at the video level.

Appearance features. Given a video, ResNet [16] is applied framewise in order to collect convolutional features associated to different keypoints. The steps of the overall pipeline are depicted in Figure 4.2. Given a frame, local and global convolutional features are extracted (as shown in step A); first, human poses in a given frame are estimated using the 2D pose extractor of [194] which provides the coordinates of 18 different joints. Afterwards, each region around a joint is rescaled to a fixed width (set in practice to 50 pixels), and a 10 x 10 pixel region, around this joint, is cropped and endowed with a convolutional feature (as shown in step B). In order to enhance the discrimination power of each joint, global convolutional features are also appended to the local ones (see step C). Finally, the obtained joint features are hierarchically aggregated (through frames) resulting into a global node representation at the video level (as shown in step D).

Motion features. Given a video, the method in [194] is used in order to extract 18 colorized heatmaps corresponding to 18 human joints tracked through different frames. These heatmaps correspond to timely-stamped framewise probability distributions of these joints. Similarly to appearance features, 10x10 pixel regions are first extracted around each joint at different instants, in the underlying colorized heatmaps, and then aggregated resulting into a global node representation at the video level. The steps of the overall pipeline are depicted in Figure 4.3. Note that the choice of heatmaps, instead of optical flow (e.g.,[28]), is motivated by the fact that the horizontal and the vertical displacement fields of optical flow are noisy and less discriminating, compared to colorized heatmaps, especially on scenes with imperceivable motion.

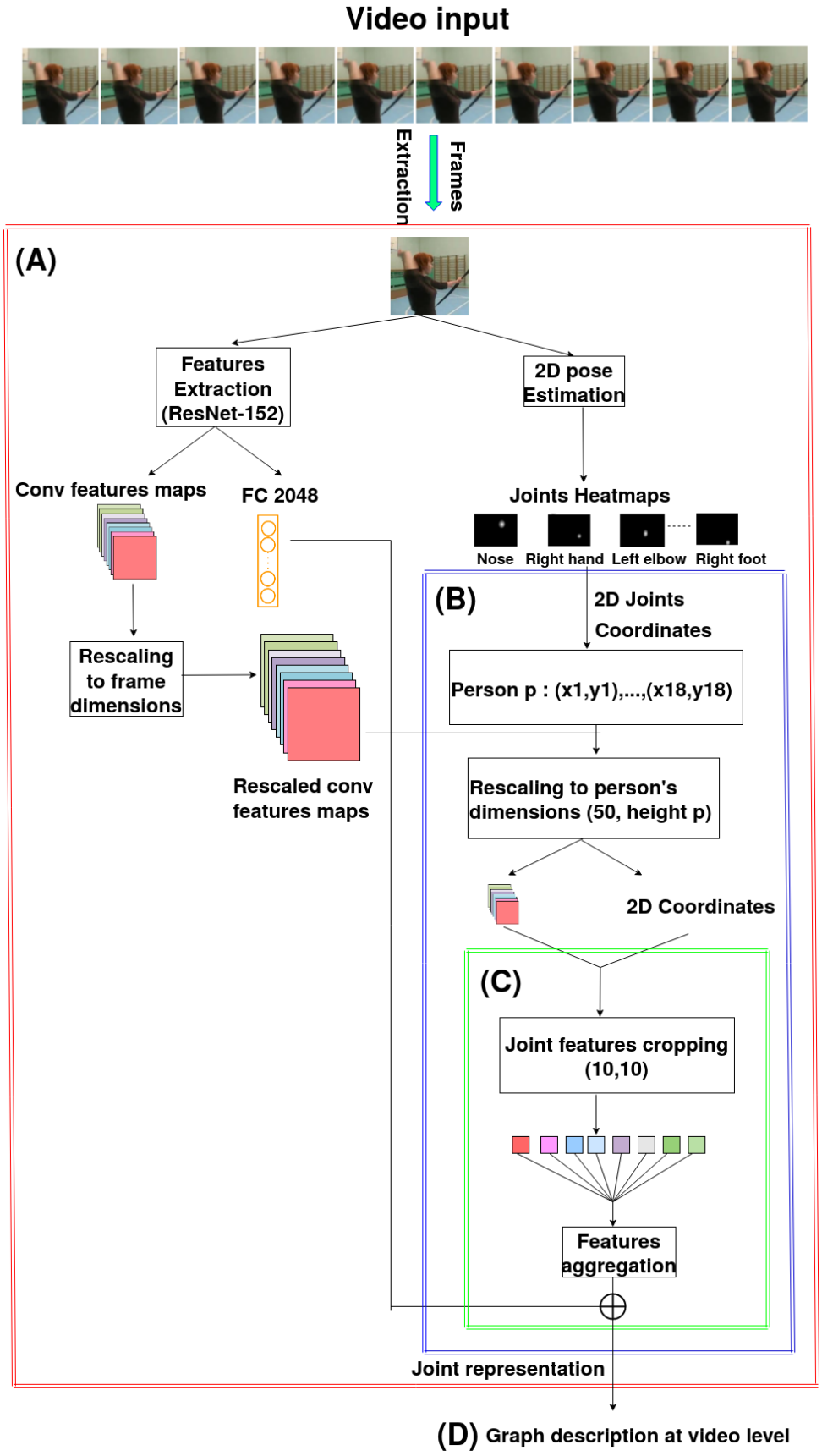


FIGURE 4.2 – Appearance graph representations at frame level.

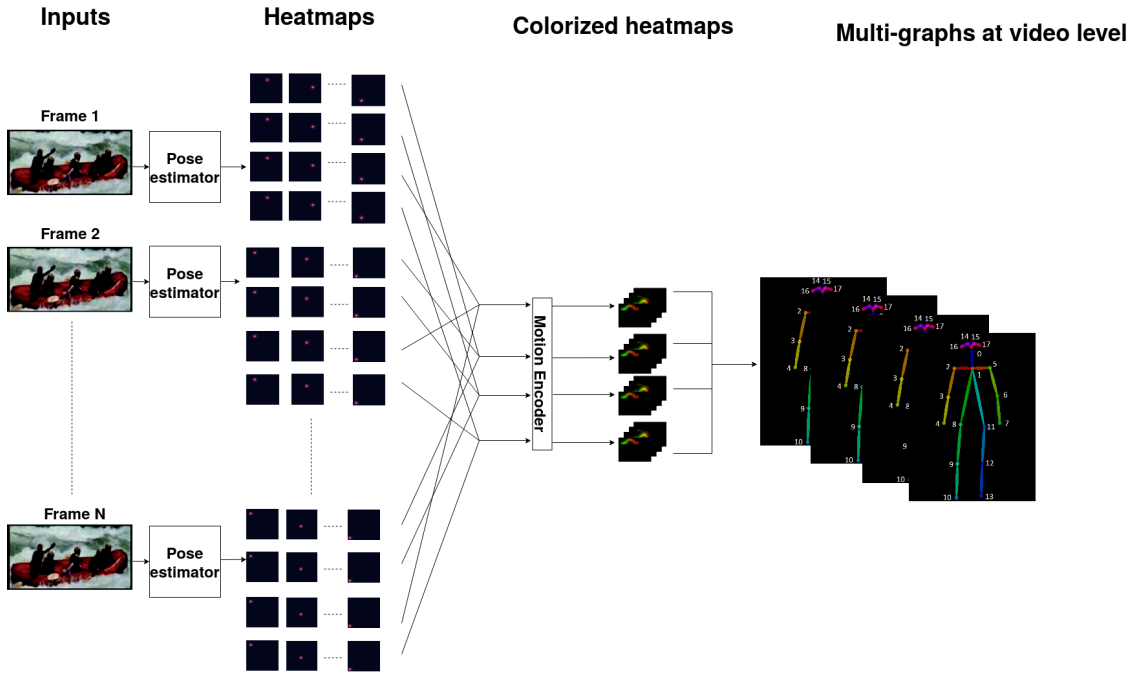


FIGURE 4.3 – Graph based motion features at video level. Each frame is fed to the human pose estimator [194] to extract the joints and their respective heatmaps. Each joint is associated with a heatmap describing its probability distribution through the spatial extent of its frame. The resulted heatmaps are then colorized and averaged to build a video level representation.

4.3 Multi-Laplacian Convolutional Networks

Given a collection of videos, we describe each one using a graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ as explained in Section 4.2. For each node $v \in \mathcal{V}_i$, we extract two feature vectors, denoted $\psi_m(v)$, $\psi_a(v)$, respectively corresponding to motion and appearance streams of v.

We also define a similarity between nodes in $\mathcal{V}_i$ as $\kappa_m(v, v') = \exp(-\|\psi_m(v) - \psi_m(v')\|_2^2/\sigma_m)$, here σ_m is the scale of the gaussian similarity and $\|.\|_2$ is the ℓ_2 norm. Similarly, we define $\kappa_a(v, v')$ using appearance features.

In the remainder of this chapter, unless explicitly mentioned, we denote a given graph $\mathcal{G}_i$ simply as $\mathcal{G}$. We also denote motion and appearance features $\psi_m(v)$, $\psi_a(v)$ as $\psi(v)$, scales σ_m, σ_a as σ, and similarities $\kappa_m(v, v')$, $\kappa_a(v, v')$ as $\kappa(v, v')$.

The goal is to design a GCN that returns the representation and the classification of a given graph. This includes *a novel design of Laplacian convolution and pooling* on graphs as shown subsequently.

4.3.1 Spectral Graph Convolution at a Glance

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n$, $|\mathcal{E}|$ being respectively the number of its vertices and edges and L the Laplacian of $\mathcal{G}$; for instance, L could be the normalized, unormalized or random walk Laplacians respectively defined as $L = I_n - D^{-1/2}\ A\ D^{-1/2}$, $L = D - A$ and $L = D^{-1}A$ where I_n is an $n \times n$ identity matrix, A is the affinity matrix built as $[A]_{vv'} = \mathbb{1}_{\{(v,v')\in\mathcal{E}\}}$ or by using the Gaussian similarity $\kappa(.,.)$ as $[A]_{vv'} = \mathbb{1}_{\{(v,v')\in\mathcal{E}\}}.\kappa(\psi(v), \psi(v'))$ and D a diagonal degree matrix with each diagonal entry $[D]_{vv} = \sum_{v'}[A]_{vv'}$.

Considering the eigen-decomposition of L as $U\Lambda U'$ with U, Λ being respectively the matrix of its eigenvectors (graph Fourier modes) and the diagonal matrix of its non-negative eigenvalues, spectral graph convolution is a well defined operator (see for instance [99]) which is achieved by first projecting a given graph signal $\psi(.)$ using the eigen-decomposition of L, and then multiplying the resulting projection by a convolutional filter prior to back-project the result in the original signal space.

Formally, the convolutional operator $\star_{\mathcal{G}}$ (rewritten for short as $\star$) on the graph signal $\psi(\mathcal{V}) \in \mathbb{R}^{n\times m}$ is

$$(\psi \star g_\theta)(\mathcal{V}) = U\ g_\theta(\Lambda)\ U'\ \psi(\mathcal{V}) \tag{4.1}$$

here g_θ denotes a non-parametric convolutional filter defined as $g_\theta(\Lambda) = diag(\theta)$ with $\theta \in \mathbb{R}^n$. As this filter is not localized, we consider instead [99]

$$(\psi \star g_\theta)(\mathcal{V}) := \sum_{k=0}^{K-1} \theta_k\ T_k(L)\ \psi(\mathcal{V}), \tag{4.2}$$

with K fixed and $\theta = (\theta_1 \dots \theta_K)' \in \mathbb{R}^K$ being its learned convolutional filter parameters; in practice, we consider a rescaled version of the Laplacian (i.e., $2L/\lambda_{max} - I_n$ instead of L with λ_{max} being its largest eigenvalue).

In the above equation, T_k is the k-th order Chebyshev polynomial recursively defined as $T_k(L) = 2L\ T_{k-1}(L) - T_{k-2}(L)$, with $T_k(L) \in \mathbb{R}^{n\times n}$ and $T_0 = I$, $T_1 = L$ (for more details see again [99]).

4.3.2 Multi-Laplacian Design

The success of the aforementioned convolutional process is highly dependent on the *relevance* of the used Laplacian, which in turn depends on the appropriate choice of the affinity matrix of the graph and its hyper-parameters.
Hence, knowing a priori which parameter to choose could be challenging and usually relies on the tedious cross-validation.

Our alternative contribution in this work aims at designing convolutional Laplacian operators while learning the topological structure of the input graphs (characterized by their Laplacians).
Starting from different *elementary* Laplacians [3] associated to multiple settings (for instance, by varying the scale σ of the gaussian similarity $\kappa(.,.)$ and the Laplacians), we train a *multiple Laplacian* as a deep nonlinear combination of multiple elementary Laplacians.

Figure 4.4 shows our learning framework with d-layers in the multi-Laplacian; for each layer $\ell+1$ ($\ell \in \{0,\dots,d-1\}$) and its associated unit $p \in \{1,\dots,n_{\ell+1}\}$, a Laplacian (denoted $L_p^{\ell+1}$) is recursively defined as

$$L_p^{\ell+1} = g\left(\sum_{q=1}^{n_\ell} \mathbf{w}_{q,p}^{\ell}\, L_q^{\ell}\right), \tag{4.3}$$

where g is a nonlinear activation function (see details in Section 4.4), n_ℓ is the number of units in layer ℓ and $\{\mathbf{w}_{q,p}^{\ell}\}_q$ are the (learned) weights associated to $L_p^{\ell+1}$.

For any given graph $\mathcal{G}$, a tensor of multiple elementary Laplacians $\{L_q^1\}_q$ (associated to different combinations of $\{\sigma\}$ and standard Laplacians namely unormalized, normalized, random walk, etc.) on $\mathcal{G}$ is considered as an input to our deep network. These elementary Laplacians are then forwarded to the subsequent intermediate layer resulting into n_2 multiple Laplacians through the nonlinear combination of the previous layer, etc. The final Laplacian L_1^d is a highly nonlinear combination of elementary Laplacians.

We notice that the deep Laplacian network in essence is a multi-layer perceptron (MLP), with nonlinear activation functions which is fed (together with the graph signal $\psi(\mathcal{V})$) as input in order to achieve convolution (see Figure 4.4). Hence, we can use standard backpropagation in order to optimize the parameters of both

3. Also referred to as single or individual Laplacians.

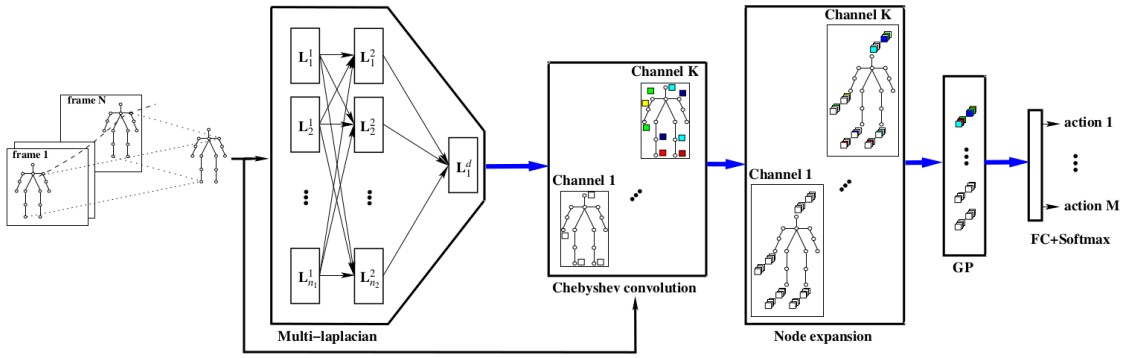


FIGURE 4.4 – This figure shows the architecture of our multi-Laplacian graph convolutional network (MLGCN). First, multiple elementary Laplacians (associated to $\mathcal{G} = (\mathcal{V}, \mathcal{E})$) and graph signal $\psi(\mathcal{V})$ are fed as input to an MLP in order to learn the best combination of Laplacians. Then, Chebyshev decomposition is achieved using the learned multi-Laplacian in order to perform graph convolution, followed by node expansion and global average pooling prior to softmax classification.

the MLP and the GCN networks. Let J denotes the loss function associated to our classification problem (namely cross-entropy); starting from the gradients of this loss J w.r.t the final softmax output, we use the chain rule in order to backpropagate the gradients w.r.t different layers and parameters (fully connected and convolutional layers as well as the MLP of the multi-Laplacians), and to update these parameters accordingly using gradient descent.

4.3.3 Temporal Pyramid MLGCN

As an extension, we define **TP-MLGCN** as a combination of multiple Laplacians each one dedicated to a particular node in a temporal pyramid. Similarly to *chapter 3*, nodes in this temporal pyramid capture coarse as well as fine temporal granularities in actions. Finding the most relevant Laplacian, that captures the best combination of granularities is not straightforward and requires a careful design. This is achieved by instantiating our **MLGCN** to hierarchical Laplacian learning where each node in the temporal pyramid is associated to a Laplacian that captures its underlying granularity. With this hierarchy of Laplacians, **TP-MLGCN** finds the best tradeoff between (i) top levels providing timely invariant video representations and (ii) bottom levels yielding timely resolute and well localized representations. The corresponding tree-structured network of Laplacians is illustrated in Figure 4.5.

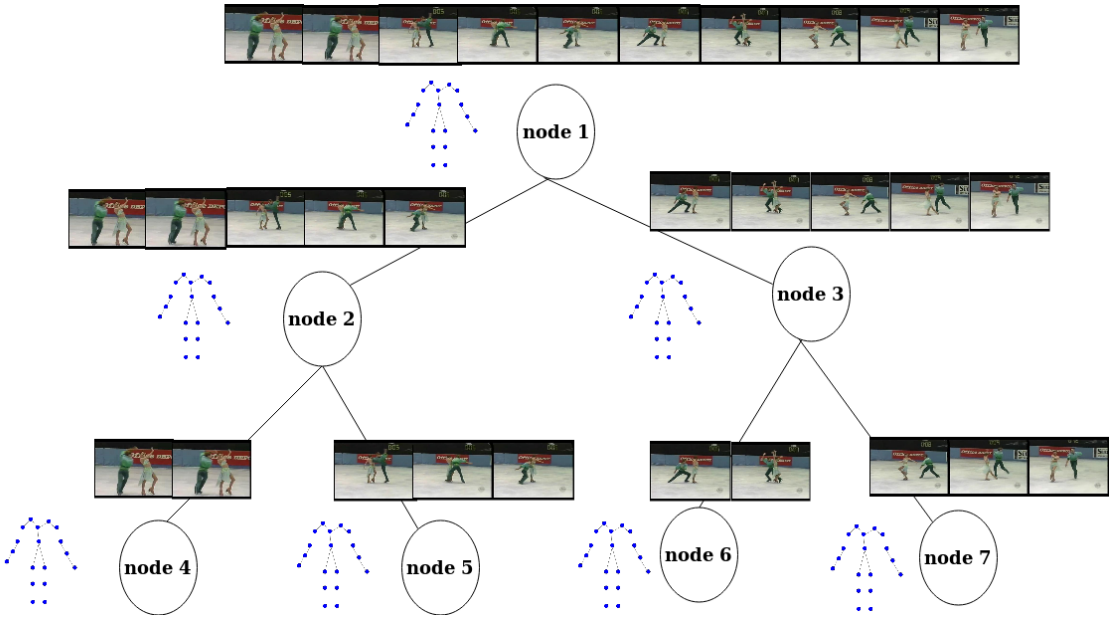


FIGURE 4.5 – Temporal pyramid on spatio-temporal graph data based on the work described in the previous Chapter 3. Each node of the temporal pyramid is represented by a graph describing a part of video, except the root node which takes the whole video sequence. This temporal pyramid is applied to build appearance and motion graph representation encoding different levels of granularity. The process of describing graphs at each node is illustrated in Figure 4.1.

4.4 Activation Functions and Optimization

We consider two activation functions g in Equation 4.3 : ReLU and leaky ReLU [225, 265, 266]. Note that only leaky ReLU provides negative entries in the learned Laplacians and both of these activations allow learning *conditionally* positive definite (c.p.d) Laplacian matrices.

In what follows, we discuss the sufficient conditions about the choices of the elementary input Laplacians, the parameters $\{\mathbf{w}_{q,p}^{\ell}\}$ and the activation functions that guarantee this c.p.d property.

Definition 4.1 (conditionally positive definite Laplacians). A Laplacian matrix L is conditionally positive definite, iff $\forall c_1, \ldots, c_n \in \mathbb{R}$ (with $\sum_{i=1}^{n} c_i = 0$), $\sum_{i,j} c_i c_j L_{ij} \geq 0$.

From the above definition, it is clear that any positive definite Laplacian is also c.p.d. The converse is not true, however c.p.d is a weaker (but sufficient) condition

in order to derive positive definite Laplacians (see following propositions).

Proposition 4.2 ([267]). *Consider $L_{i,j}$ as an entry of a matrix L and define $\hat{L}$ with*

$$\hat{L}_{i,j} \quad = \quad L_{i,j} - L_{i,n+1} - L_{n+1,j} + L_{n+1,n+1} \tag{4.4}$$

Then, $\hat{L}$ is positive definite if and only if L is c.p.d.

[**Proof of Proposition**4.2]

"If" part : assuming L c.p.d, $\forall c_1, \ldots, c_n, c_{n+1}$ (with $c_{n+1} = -\sum_{i=1}^{n} c_i$)

$$\begin{aligned}
0 \quad &\leq \quad \sum_{i,j=1}^{n+1} c_i c_j L_{i,j} \\
&= \quad \sum_{i,j=1}^{n} c_i c_j \; L_{i,j} + c_{n+1} \sum_{i=1}^{n} c_i \; L_{i,n+1} + c_{n+1} \sum_{j=1}^{n} c_j \; L_{n+1,j} + c_{n+1}^2 \; L_{n+1,n+1} \\
&= \quad \sum_{i,j=1}^{n} c_i c_j \; \left(L_{i,j} - L_{i,n+1} - L_{n+1,j} + L_{n+1,n+1} \right) \\
&= \quad \sum_{i,j=1}^{n} c_i c_j \hat{L}_{i,j}.
\end{aligned}$$

"Only If" part : assuming $\hat{L}$ positive definite $\forall c_1, \ldots, c_n$ (including when $\sum_i c_i = 0$) and for any u (for instance $u \in \{1, \ldots, n\}$)

$$\begin{aligned}
0 \quad &\leq \quad \sum_{i,j=1}^{n} c_i c_j \hat{L}_{i,j} \\
&= \quad \sum_{i,j=1}^{n} c_i c_j (L_{i,j} - L_{i,u} - L_{u,j} + L_{u,u}) \\
&= \quad \sum_{i,j=1}^{n} c_i c_j L_{i,j} - \sum_{i}^{n} c_i L_{i,u} \sum_{j=1}^{n} c_j - \sum_{j=1}^{n} c_j L_{u,j} \sum_{i=1}^{n} c_i + L_{u,u} \sum_{i=1}^{n} c_i \sum_{j=1}^{n} c_j \\
&= \quad \sum_{i,j=1}^{n} c_i c_j L_{i,j}
\end{aligned}$$

■

Now we derive our main result :

Proposition 4.3. *Provided that the input elementary Laplacians* $\{L_q^1\}_q$ *are c.p.d, and* $\{\mathbf{w}_{q,p}^\ell\}_{p,q,\ell}$ *belong to the positive orthant of the parameter space, any combination* $g(\sum_q \mathbf{w}_{q,p}^\ell\, L_q^\ell)$*, with* g *equal to ReLU or leaky ReLU, is also c.p.d.*

[**Proof of Proposition**4.3]

Details of the first part of the proof, based on recursion, are omitted and result from the application of definition 4.1 to $L = \sum_q \mathbf{w}_{q,p}^\ell\, L_q^\ell$ (for different values of ℓ) while considering $\{L_q^1\}_q$ c.p.d. Now we show the second part of the proof (i.e., if L is c.p.d, then $g(L)$ is also c.p.d for ReLU and leaky ReLU).

i) For $g(L) = \log(1 + \exp(L))$ [ReLU] : considering L c.p.d, and following the proposition 4.2, one may define a positive definite $\hat{L}$ and obtain $\forall\{c_i\}$

$$\sum_{i,j=1}^{n} c_i c_j \exp(L_{i,j}) = \exp(L_{n+1,n+1}) \sum_{i,j=1}^{n} (c_i \exp(L_{i,n+1})).(c_j \exp(L_{n+1,j})).\exp(\hat{L}_{i,j}) \geq 0$$

so $\exp(L)$ is also positive definite. Besides, for any arbitrary $\alpha > 0$, $(1 + \exp(L))^{\circ\alpha}$ is also positive definite with $\circ\alpha$ being the entry-wise matrix power.
By simply rewriting $(1 + \exp(L))^{\circ\alpha} = \exp(\alpha\, g(L))$, it follows (from [268]) that $g(L)$ is c.p.d since $\exp(\alpha\, g(L))$ is positive definite for all $\alpha > 0$.

ii) For $g(L) = \log(\exp(aL) + \exp(L))$ with $0 < a \ll 1$ [leaky-ReLU] : one may write g as

$$g(L) = a\, L + \log(1 + \exp((1 - a)\, L)). \tag{4.5}$$

Since $\exp(L)$ is positive definite, it follows that $(1+\exp((1-a)\, L))^{\circ\alpha}$ is also positive definite for any arbitray $\alpha > 0$ and $0 < a \ll 1$ so from [268], $\log(1+\exp((1-a)\, L))$ is c.p.d and so is $g(L)$; the latter results from the closure of the c.p.d with respect to the sum.

■

From proposition 4.3, provided that i) the elementary Laplacians are c.p.d, ii) the activation function g preserves the c.p.d (as ReLU and leaky-ReLU) and iii) weights $\{\mathbf{w}_{q,p}^\ell\}$ are positive, all the resulting multiple Laplacians in Equation 4.3 will also be c.p.d and admit equivalent positive definite Laplacians (following proposition 4.2), and thereby spectral graph convolution can be achieved.

Note that conditions (i) and (ii) are satisfied by construction while condition (iii) requires adding equality and inequality constraints to Equation 4.3, i.e., $\mathbf{w}_{q,p}^{\ell} \in [0,1]$ and $\sum_q \mathbf{w}_{q,p}^{\ell} = 1$.

In order to implement these constraints, we consider a reparametrization in Equation 4.3 as $\mathbf{w}_{q,p}^{\ell} = f(\hat{\mathbf{w}}_{q,p}^{\ell}) / \sum_{q'} f(\hat{\mathbf{w}}_{q',p}^{\ell})$ for some $\{\hat{\mathbf{w}}_{q,p}^{\ell}\}$ with f being strictly monotonic real-valued (positive) function and this allows free settings of the parameters $\{\hat{\mathbf{w}}_{q,p}^{\ell}\}$ during optimization while guaranteeing $\mathbf{w}_{q,p}^{\ell} \in [0,1]$ and $\sum_q \mathbf{w}_{q,p}^{\ell} = 1$. During backpropagation, the gradient of the loss J (now w.r.t $\hat{\mathbf{w}}$'s) is updated similarly to Section 3.3.2.

4.5 Pooling

If pooling on regular grids (or vectorial data in general) is well defined, it is not the case for graphs [269]. As a consequence, most of GCN architectures do not include pooling layers in their architectures [259, 270] excepting a few attempts which try to incorporate pooling in a non explicit way using multi-level graph coarsening (i.e., by reducing graphs by a factor of two at each level and describing each node by the average or the max of its descendants [99, 271] or by using clustering [272, 273] and reordering [242, 274-276]).

For highly irregular graphs (e.g., with heterogeneous degrees), this graph coarsening process usually results into imbalanced hierarchical representations and this substantially affects the accuracy of the learned graph representations.
In practice, existing methods (for instance [99]) add fake nodes in the input graphs in order to rebalance the coarsening process. However, fake nodes are spurious and this may lead to contaminated graph representations after coarsening. Besides, this pooling process is not invariant to node permutations and node reordering (based on automorphisms) cannot guarantee permutation invariance for general and irregular graphs.

In this section, we consider an alternative solution in order to achieve pooling. Our method relies on two steps : an expansion-step is first achieved at the node level followed by a global average pooling in order to achieve permutation invariance.

Note that the first step (expansion) is necessary in order to generate high dimensional (and sparse) node representations and hence preserve the discrimination power of nodes before applying the second step of global average pooling. Put

differently, without expansion, average pooling achieves permutation invariance but dilutes node information and this results into less discriminant graph representations as shown in experiments.

Considering $\mathcal{N}_r(v)$ as the set of r-hop neighbors of a given node $v \in \mathcal{V}$ and $\mathcal{N}_r(v) = \cup_{l=1}^{L} \mathcal{N}_r^l(v)$ as the union of L subsets [4], the expansion of v is defined as

$$\phi(v) \leftarrow \left((\psi \star g_\theta)(v), \frac{1}{|\mathcal{N}_r^1(v)|} \sum_{v' \in \mathcal{N}_r^1(v)} (\psi \star g_\theta)(v'), \ldots, \frac{1}{|\mathcal{N}_r^L(v)|} \sum_{v' \in \mathcal{N}_r^L(v)} (\psi \star g_\theta)(v') \right). \tag{4.6}$$

For a large and fine-grained neighborhood system $\mathcal{N}_r(v) = \cup_{l=1}^{L} \mathcal{N}_r^l(v)$ (i.e., $r \geq 1$ and $L \gg 1$), the expansion $\phi(v)$ takes into account not only the immediate neighbors of v but also a large extent and this results into high dimensional, sparse and discriminating representations.

Finally, a global average pooling is performed (as $\sum_{v \in \mathcal{V}} \phi(v)$) to achieve permutation invariance prior to the softmax fully connected classification layer (see again Figure 4.4).

4.6 Experiments

We evaluate the performance of our multi-Laplacian graph convolutional networks (**MLGCN**) and our **TP-MLGCN** on the challenging task of action recognition, including 2D/3D skeleton and video (rgb) frames data, using three standard datasets : 2D JHMDB [199] (see skeleton data for JHMDB in Section 2.5.2), 3D SBU kinect [264] (see Section 2.5.3) and UCF-101 [197] (see Section 2.5.1).

Datasets Description. JHMDB is a subset of HMDB composed of 928 video sequences belonging to 21 action categories obtained in difficult conditions (but less challenging than HMDB. see again Section 2.5.2). It is quickly exposed to over-fitting due to its small size, especially when trained with deep neural networks.

SBU is an interaction dataset acquired (under relatively well controlled conditions) using the Microsoft Kinect sensor ; it includes in total 282 video sequences belonging to 8 categories : "approaching", "departing", "pushing", "kicking", "punching", "exchanging objects", "hugging", and "hand shaking".

In contrast to JHMDB and SBU which are skeleton datasets, UCF-101 is a video dataset. It is larger and more challenging ; it includes 13,320 video shots belonging to 101 categories with variable duration, poor frame resolution, viewpoint and

4. In practice, each subset $\mathcal{N}_r^l(v)$ includes only nodes with labels equal to l (see again node labels in Figure 4.1).

illumination changes, occlusion, cluttered background and eclectic content ranging from multiple and highly interacting individuals to single and completely passive ones.
In all these experiments, we use the same evaluation protocols as the ones suggested in [197, 199, 264] (i.e.,three splits for JHMDB-21, split2 for UCF-101 and train-test split for SBU) and we report the average accuracy over all the classes of actions.

4.6.1 Settings and Performances

Performances of MLGCN against elementary Laplacians without and with expansion. We trained our MLGCN for 150 epochs on UCF-101, 40 epochs on SBU and 50 epochs on JHMDB using the PyTorch SGD optimizer and we set the learning rate to 0.0006 (decayed by a factor 0.1 after 100 epochs) for UCF-101, 0.17 for JHMDB and 0.7 for SBU. We set the batch size to 30 and the Chebyshev order K to $4,\ 3,\ 4$ respectively for UCF, JHMDB and SBU using grid search and cross validation. All these experiments are run on GPUs; Tesla P100 (with 16 Gb) for UCF-101 and Titan X Pascal (with 12 Gb) for JHMDB and SBU.

Table 4.2, Table 4.4 and Table 4.6 show a comparison of action recognition performances, using MLGCN against different baselines involving individual Laplacians (normalized, unormalized, random walk built on top of different affinity matrices and scale parameters). These tables show the results using expansion (as described in Section 4.5) and global average pooling (GP) while in Table 4.1, Table 4.3 and Table 4.5, we show the results without expansion.

2D skeletons JHMDB		Binary	Binary × Gaussian							Multi-lap
			$10^0\sigma$	$10^1\sigma$	$10^2\sigma$	$10^3\sigma$	$10^4\sigma$	$10^5\sigma$	$10^6\sigma$	
Unormalized	$k=1$	54.33	52.33	49.67	54.33	54.33	53.67	53.67	55.03	55.03
	$k=4$	54.33	50.66	49.67	54.67	54.00	54.33	53.67	53.00	55.05
	$k=15$	54.33	52.33	50.33	54.67	54.00	53.67	53.67	53.33	55.10
Normalized	$k=1$	54.33	52.33	54.00	54.33	54.33	54.33	54.67	56.38	56.38
	$k=4$	55.00	54.33	52.00	54.00	54.33	54.33	54.67	54.67	56.41
	$k=15$	55.00	54.33	52.33	54.00	54.00	54.33	54.67	54.67	58.71
Random walk	$k=1$	54.00	53.00	54.00	53.67	54.33	55.00	55.00	53.66	55.64
	$k=4$	54.00	53.00	54.00	53.67	54.33	55.00	55.00	53.00	55.64
	$k=15$	54.00	52.33	54.00	53.67	54.33	55.00	55.00	53.67	55.65
Multi-lap (MLGCN)		55.93	54.93	54.22	54.90	55.85	55.77	55.58	54.98	58.08

TABLE 4.1 – Performances on JHMDB (over the three splits) *without expansion* for different elementary Laplacians (normalized, unormalized and random walk) and their marginal and total combinations using MLGCN (note that our expansion is not used). In this table, "binary" means that A^k is used to build the elementary Laplacian while "binary × Gaussian" means that "A^k× Gaussian similarity" is used instead ; for each graph $\mathcal{G}$, the scale σ of the Gaussian similarity is taken as the average distance between node features in $\mathcal{G}$. Table 4.2 shows results *with expansion* as described in Section 4.5.

2D skeletons JHMDB		Binary	Binary × Gaussian							Multi-lap
			$10^0\sigma$	$10^1\sigma$	$10^2\sigma$	$10^3\sigma$	$10^4\sigma$	$10^5\sigma$	$10^6\sigma$	
Unormalized	$k=1$	55.33	52.67	57.33	57.33	56.67	56.67	56.33	57.93	57.93
	$k=4$	57.33	53.67	52.67	57.67	57	56.67	56.67	56.00	57.85
	$k=15$	57.33	55.33	52.67	57.67	57.00	56.67	56.67	56.33	57.94
Normalized	$k=1$	58.00	57.33	55.33	57.00	57.33	57.33	57.33	57.67	59.23
	$k=4$	58.00	57.33	55.00	57.00	57.33	57.33	57.67	57.67	58.85
	$k=15$	58.00	57.33	55.33	57.00	57.00	57.33	57.67	57.67	59.22
Random walk	$k=1$	57.00	56.00	57.00	57.00	57.33	58.00	58.00	56.67	58.58
	$k=4$	57.00	56.00	57.00	56.67	57.33	58.00	58.00	56.67	58.48
	$k=15$	57.00	56.00	57.00	56.67	57.33	58.00	58.00	56.67	58.57
Multi-lap (MLGCN)		58.61	58.01	57.16	57.85	59.14	58.53	58.52	58.00	61.21

TABLE 4.2 – Performances on JHMDB (over the three splits) *with expansion*. See Table 4.1 for results *without expansion* and for the settings.

3D skeletons SBU		Binary	Binary × Gaussian												Multi-lap
			$10^{-6}\sigma$	$10^{-5}\sigma$	$10^{-4}\sigma$	$10^{-3}\sigma$	$10^{-2}\sigma$	$10^{-1}\sigma$	σ	10σ	$10^{2}\sigma$	$10^{3}\sigma$	$10^{4}\sigma$	$10^{5}\sigma$	$10^{6}\sigma$	
Unormalized	$k=1$	92.22	91.73	91.73	91.73	91.73	91.73	91.73	91.73	91.73	91.71	91.71	91.71	91.71	91.71	92.69
	$k=4$	88.90	87.95	87.95	87.95	87.95	87.95	87.95	87.95	87.95	87.95	87.92	87.92	87.92	87.92	89.61
	$k=32$	85.78	84.48	84.48	84.48	84.48	84.48	84.48	84.48	84.50	84.50	84.50	84.50	84.50	84.50	86.28
Normalized	$k=1$	92.34	91.78	91.78	91.78	91.78	91.78	91.78	91.75	91.75	91.75	91.75	91.75	91.77	91.77	92.78
	$k=4$	89.67	88.56	88.56	88.56	88.56	88.56	88.56	88.56	88.59	88.59	88.59	88.56	88.56	88.56	90.13
	$k=32$	87.60	86.48	86.48	86.48	86.48	86.48	86.48	86.48	86.50	86.50	86.50	86.50	86.50	86.50	88.17
Random w	$k=1$	92.57	91.17	91.17	91.17	91.16	91.17	91.17	91.17	91.20	91.20	91.20	91.17	91.17	91.17	92.88
	$k=4$	95.81	93.88	93.88	93.88	93.81	93.81	93.80	93.80	93.79	93.83	93.81	93.80	93.80	93.81	96.12
	$k=32$	95.77	93.85	93.85	93.85	93.85	93.85	93.86	93.86	93.86	93.84	93.84	93.84	93.84	93.84	96.10
Multi-lap		96.96	94.26	94.26	94.26	94.28	94.28	94.28	94.28	94.27	94.26	94.26	94.26	94.26	94.26	98.14

TABLE 4.3 – Performances on SBU *without expansion*. See Table 4.1 for the settings and Table 4.4 for results *with expansion*.

3D skeletons SBU		Binary	Binary × Gaussian												Multi-lap
			$10^{-6}\sigma$	$10^{-5}\sigma$	$10^{-4}\sigma$	$10^{-3}\sigma$	$10^{-2}\sigma$	$10^{-1}\sigma$	σ	10σ	$10^{2}\sigma$	$10^{3}\sigma$	$10^{4}\sigma$	$10^{5}\sigma$	$10^{6}\sigma$	
Unormalized	$k=1$	93.00	92.32	92.32	92.32	92.32	92.32	92.32	92.32	92.32	92.30	92.30	92.30	92.30	92.30	93.41
	$k=4$	89.25	88.87	88.87	88.87	88.87	88.87	88.87	88.87	88.87	88.87	88.86	88.86	88.86	88.86	90.07
	$k=32$	86.00	86.31	86.31	86.31	86.31	84.31	86.31	86.31	86.32	86.32	86.32	86.32	86.32	86.32	86.91
Normalized	$k=1$	93.00	92.28	92.28	92.28	92.28	92.28	92.28	92.26	92.26	92.26	92.26	92.26	92.28	92.28	93.49
	$k=4$	90.00	89.36	89.36	89.36	89.36	89.36	89.36	89.36	89.38	89.38	89.39	89.37	89.37	89.37	91.49
	$k=32$	88.00	88.31	88.31	88.31	88.31	88.31	88.31	88.31	88.32	88.32	88.32	88.32	88.32	88.32	89.21
Random w	$k=1$	93.00	92.05	92.05	92.06	92.05	92.05	92.05	92.05	92.09	92.09	92.09	92.06	92.06	92.06	93.46
	$k=4$	96.00	94.06	94.06	94.06	94.00	94.00	94.00	94.01	94.00	94.01	94.00	94.00	94.00	94.00	96.31
	$k=32$	96.00	94.03	94.03	94.03	94.03	94.03	94.03	94.03	94.03	94.02	94.02	94.02	94.02	94.02	96.29
Multi-lap		97.15	94.61	94.58	94.61	94.63	94.63	94.63	94.62	94.63	94.63	94.63	94.63	94.63	94.63	**98.6**

TABLE 4.4 – Performances on SBU *with expansion*. See Table 4.1 for the settings and Table 4.3 for results *without expansion*.

Video frames UCF-101		Binary	Binary × Gaussian												Multi-lap
			$10^{-6}\sigma$	$10^{-5}\sigma$	$10^{-4}\sigma$	$10^{-3}\sigma$	$10^{-2}\sigma$	$10^{-1}\sigma$	σ	10σ	$10^{2}\sigma$	$10^{3}\sigma$	$10^{4}\sigma$	$10^{5}\sigma$	$10^{6}\sigma$	
Unormalized	$k=1$	54.78	49.08	49.08	48.08	48.08	48.10	48.10	48.10	48.13	48.13	48.13	48.13	48.09	48.10	55.38
	$k=4$	59.05	54.69	54.69	54.69	54.69	54.62	54.60	54.61	54.15	54.15	54.15	54.18	54.22	54.15	59.80
	$k=32$	54.66	51.37	51.37	51.37	51.37	51.52	51.52	51.50	51.51	51.51	51.78	51.78	51.78	51.75	55.31
Normalized	$k=1$	55.10	49.23	49.23	49.05	49.11	49.11	49.12	49.12	49.12	49.12	49.12	49.11	49.11	49.11	55.95
	$k=4$	59.2	54.89	54.89	54.89	54.60	54.62	53.95	53.95	53.95	53.95	53.95	53.96	53.96	53.96	59.98
	$k=32$	54.90	50.46	50.46	50.45	50.10	50.10	50.11	50.10	50.10	50.12	50.12	50.10	50.10	50.10	55.70
Random w	$k=1$	59.78	56.71	56.71	56.71	56.77	56.71	56.71	56.71	56.74	56.66	56.66	56.66	56.66	56.68	60.10
	$k=4$	61.25	56.80	56.80	56.80	56.80	56.80	56.75	56.76	56.75	56.70	56.70	56.70	56.70	56.72	61.35
	$k=32$	59.95	56.74	56.74	56.74	56.74	56.74	56.74	56.74	56.76	56.68	56.68	56.68	56.65	56.65	61.16
Multi-lap		61.50	57.00	56.95	56.93	56.93	56.93	56.90	56.96	56.91	56.91	56.94	56.94	56.95	56.97	62.70

TABLE 4.5 – Performances on UCF *without expansion*. See Table 4.1 for the settings and Table 4.6 for results *with expansion*.

Video frames UCF-101		Binary	Binary × Gaussian												Multi-lap
			$10^{-6}\sigma$	$10^{-5}\sigma$	$10^{-4}\sigma$	$10^{-3}\sigma$	$10^{-2}\sigma$	$10^{-1}\sigma$	σ	10σ	$10^{2}\sigma$	$10^{3}\sigma$	$10^{4}\sigma$	$10^{5}\sigma$	$10^{6}\sigma$	
Unormalized	$k=1$	55.32	50.67	50.67	50.67	50.68	50.70	50.70	50.70	50.71	50.72	50.72	50.72	50.70	50.70	56.55
	$k=4$	59.23	55.22	55.22	55.22	55.22	55.20	55.20	55.20	54.95	54.96	54.95	54.98	55.00	54.98	60.05
	$k=32$	55.10	52.05	52.05	52.05	52.05	52.11	52.11	52.11	52.11	52.11	52.06	52.06	52.06	52.08	56.48
Normalized	$k=1$	55.6	50.78	50.77	50.27	50.42	50.40	50.42	50.42	50.42	50.42	50.42	50.42	50.42	50.42	56.80
	$k=4$	59.45	55.32	55.35	55.35	55.00	55.00	54.60	54.60	54.60	54.60	54.60	54.60	54.60	54.60	60.35
	$k=32$	55.25	51.19	51.19	51.19	49.78	49.79	49.79	49.79	49.79	49.78	49.78	49.77	49.77	49.77	56.52
Random w	$k=1$	60.09	58.00	58.00	57.98	58.00	58.00	58.00	58.00	58.01	57.95	57.95	57.95	57.92	57.94	60.85
	$k=4$	61.63	58.05	58.05	58.05	58.05	58.05	58.02	58.02	58.02	57.98	57.98	57.98	57.98	58.02	61.90
	$k=32$	60.23	58.02	58.02	58.02	58.02	58.01	58.02	58.02	58.01	57.95	57.95	57.95	57.92	57.92	60.9
Multi-lap		62.00	58.24	58.16	58.14	58.14	58.14	58.15	58.15	58.13	58.14	58.16	58.18	58.15	58.17	**63.27**

TABLE 4.6 – Performance on UCF *with expansion*. See Table 4.1 for the settings and Table 4.5 for results *without expansion*.

Methods	Video frames	Skeletons	
	UCF-101	2D JHMDB	3D SBU
MLGCN (level 1)	62.70	58.08	98.14
TP-MLGCN (level 2)	63.11	58.63	**98.47**
TP-MLGCN (level 3)	**63.34**	**58.96**	98.35
TP-MLGCN (level 4)	63.19	58.77	98.26

TABLE 4.7 – Performances on UCF, JHMDB and SBU w.r.t the choice of granularity level in TP-MLGCN. We show results *without expansion* here and *with expansion* in Table 4.8.

Methods	Video frames	Skeletons	
	UCF-101	2D JHMDB	3D SBU
MLGCN (level 1)	63.27	61.21	98.6
TP-MLGCN (level 2)	63.91	61.72	**98.90**
TP-MLGCN (level 3)	**64.17**	**61.96**	98.83
TP-MLGCN (level 4)	64.00	61.79	98.72

TABLE 4.8 – Performances on UCF, JHMDB and SBU w.r.t the choice of granularity level in TP-MLGCN. We show results *with expansion* here and *without expansion* in Table 4.7.

From all these tables, we observe a clear and consistent gain of **MLGCN** w.r.t all the individual Laplacian settings. This gain is further amplified when using our *expansion* method (described in Section 4.5) followed by global pooling.

TP-MLGCN vs **MLGCN**. We follow the parameter settings of **MLGCN** described in previous paragraphs to train our **TP-MLGCN** in order to compare the models in the same conditions. The batch size, the Chebyshev order K, and the learning rate for the different datasets are identical to those of **MLGCN** while the number of epochs in **TP-MLGCN** is increased since **TP-MLGCN** model is more complex than **MLGCN**, including more parameters to learn. Hence, we set 210 epochs (instead of 150 epochs) on UCF-101, 70 epochs (instead of 40) on SBU and 65 epochs (instead of 50 epochs) on JHMDB. We measure the performance of our **TP-MLGCN** [5] without and with expansion (see Table 4.7 for results without expansion and Table 4.8 for results with expansion) on the three datasets.

The first row of Table 4.7 and Table 4.8 show results of the first level of **TP-MLGCN** which corresponds to **MLGCN**; the coarse-grained level of video representations and their associated graph Laplacians as shown in Figure 4.5 (node 1). The three remaining rows report respectively results with an increasing granularity level.
From the experimental results, we again observe the outperformance of the settings with expansion over settings without expansion. This confirms the motivation of our expansion pooling design (see again Section 4.5) and its robustness in different GCN architectures.

Different pooling strategies and skeleton representations. We also show in Table 4.9, Table 4.10 and Table 4.11 the results for (i) different pooling strategies (no-pooling, only GP, feature propagation [251] and feature propagation+GP), (ii) various multi-Laplacian depths and activation functions [6] and (iii) different input graph descriptions for skeleton data (JHMDB and SBU).

From all these results, we observe a clear gain of our **TP-MLGCN** and **MLGCN** w.r.t single Laplacian settings regarding different pooling strategies and skeleton representations. This gain results from the *complementary aspects of the used ele-*

5. Up to four levels
6. As shown in Table 4.10, performances improve/stabilize very quickly, as the depth increases, since the size of the training set is limited compared to the large number of training parameters in the MLP of the multi-Laplacian. These performances are consistently better when using leaky ReLU (compared to ReLU) and this is explained by the modeling capacity of the former. Indeed, leaky ReLU reflects better the (positive and negative) values of our Laplacians while ReLU cuts off all the negative values.

Pooling	Single-lapl			Multi-lap			TP-MLGCN		
	JHMDB	SBU	UCF	JHMDB	SBU	UCF	JHMDB	SBU	UCF
No pooling	54.52	93.94	59.16	58.72	95.70	61.20	59.19	95.75	61.26
Global Pooling (GP)	54.00	93.90	59.10	58.08	95.62	61.17	58.67	95.68	61.21
Features prop [251]	55.19	94.27	59.30	58.97	96.36	61.31	58.82	96.38	61.37
Features prop [251] + GP	55.10	94.30	59.26	58.87	96.43	61.25	58.74	96.46	61.32
Exp ($r = 1$, $L = 1$)+GP	55.23	94.15	59.20	58.81	96.35	61.25	58.68	96.40	61.32
Exp ($r = 2$, $L = 1$)+GP	55.31	94.32	59.33	58.95	96.42	61.30	58.95	96.56	61.37
Exp ($r = 1$, $L = n$)+GP	**58.00**	**96.00**	**61.63**	**61.21**	**98.60**	**63.27**	**61.96**	**98.90**	**64.17**

TABLE 4.9 – Behavior of our MLGCN with and without expansion, i.e., after its ablation and replacement with other pooling methods. Note that results with the best single Laplacians taken from Table 4.2, Table 4.4 and Table 4.6 are also shown.

Depth	Leaky ReLU			ReLU		
	JHMDB	SBU	UCF	JHMDB	SBU	UCF
1	**61.21**	**98.60**	63.10	61.19	98.57	63.07
2	61.20	98.56	**63.27**	61.16	98.52	63.25
3	61.11	98.30	63.27	61.07	98.23	63.23

TABLE 4.10 – Behavior of our *MLGCN* w.r.t different depths and activation functions.

mentary Laplacians and also the *match* between the topological properties of the learned multiple Laplacians and the actual topology of the manifolds enclosing the input graphs. These Performances are further amplified when using "expansion+GP" with a large spatial extent and a fine-grained neighborhood system $\mathcal{N}_r(v) = \cup_{l=1}^{L} \mathcal{N}_r^l(v)$ (i.e., $r \geq 1$ and $L \gg 1$). *expansion+GP* aggregates the representations of the learned GCN filters in a way that maintains their high discrimination power (at the node level) while achieving permutation invariance. The latter is clearly necessary especially when handing videos with multiple interacting persons that frequently appear in interchangeable orders (as in SBU and UCF).

4.6.2 Augmentation

In this section, we propose a graph augmentation method in order to increase the size of action recognition datasets based on video (rgb) frames (UCF-101 in our case) and to mitigate the effect of over-fitting, and also to enhance the performance of classification.

Data augmentation plays an important role in ML and in DL. For instance, in the case of image classification with ConvNets, augmentation consists in random

Skeleton representation	Single-Lap		Multi-Lap		TP-MLGCN	
	JHMDB	SBU	JHMDB	SBU	JHMDB	SBU
Cloud of joints	37.87	31.65	40.97	34.25	41.97	34.36
Spatio-temporel skeletons	40.9	36.10	44.5	38.00	44.96	38.09
Orthocentred joints	✗	43.25	✗	45.80	✗	45.91
Cylindrical features [277, 278]	43.02	38.42	46.15	40.10	46.69	40.15
3D coord +velocity features [279]	43.03	38.50	46.22	40.20	46.73	40.36
Joint joint orientation [280]	50.68	74.95	53.37	76.20	54	76.29
Joint line distance [280]	54.78	85.60	57.03	87.50	57.77	87.65
Our temporal chunking (Section 4.2)	**58**	**96.00**	**61.21**	**98.60**	**61.96**	**98.90**

TABLE 4.11 – Performance of MLGCN on SBU and JHMDB for different state of the art skeleton graph/node representations; again results are also shown for the best underlying single Laplacians (taken from Table 4.2, Table 4.4 and Table 4.6). In this table, "Cloud of joints" stand for graphs based on the similarity between all the keypoints of different frames; "Spatio-temporel skeleton" graphs are obtained by computing intra-frame joint similarity and by connecting them to their predecessors and successors through frames; "Orthocentered joints" are obtained by centering the keypoint coordinates of each skeleton in each frame. Details about the other used node features (namely "Cylindrical features", "3D coord + velocity features", " Joint joint orientation" and "Joint line distance") can be found in [277-282]. ✗means that orthocentred joints representation doesn't apply for JHMDB because there is one person per frame.

UCF-101 (Appearance features)		Accuracy		
		Single-Laplacian	MLGCN	TP-MLGCN
Global pooling	Without augmentation	61.25	62.70	63.34
	With augmentation	61.79	63.34	63.89
Expansion + Global pooling	Without augmentation	61.63	63.27	64.17
	With augmentation	62.12	64.00	**64.74**

TABLE 4.12 – Performance of *Single Laplacian*, **MLGCN** and **TP-MLGCN** on UCF-101 (appearance features) without and with graph augmentation, including *global pooling* and *expansion + global pooling*. The best underlying *single Laplacian*, **MLGCN** and **TP-MLGCN** without augmentation, with *global pooling* and with *expansion + global pooling* are taken from Table 4.5, Table 4.6, Table 4.7 and Table 4.8.

flipping, rotating, and translating the input data.

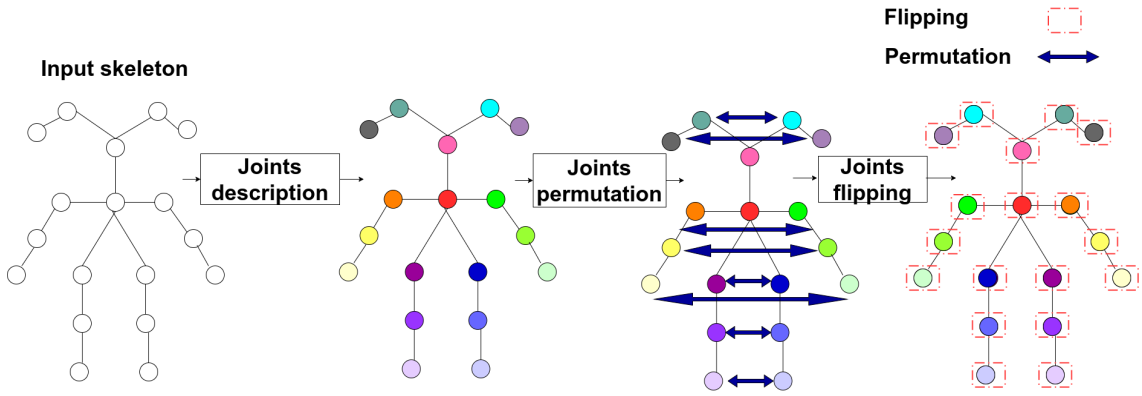


FIGURE 4.6 – The process of graph augmentation for joint features based on convolutional features. See Section 4.2.2 for joints description (appearance features for UCF-101).

Here, we propose an augmentation for the specific task of human action recognition (graph classification) based on joint features extracted from pretrained ConvNets on *ImageNet* (see Section 4.2.2). Our skeleton graphs are characterized by a central *symmetry* w.r.t to their vertical axis. Following this property, one way to do augmentation is to permute the joints of the left side with those of the right side. For instance, permuting the joints of right/left hands and shoulders. The whole process of graph augmentation is depicted in Figure 4.6.

Table 4.12 shows the effectiveness of our graph augmentation method and the extra gain in performance that it brings in different settings, including *Single Laplacian*, **MLGCN** and **TP-MLGCN** along with *global pooling* and *expansion + global pooling*.

4.6.3 Comparison Against State-Of-The-Art

In this section, we compare the classification performances of our **MLGCN** and **TP-MLGCN** against related methods ranging from standard machine learning ones (SVMs [227, 264], sequence based such as LSTM and GRU [283-285], 2D/3D CNNs [15, 28, 194, 227] including appearance and motion streams) to deep graph (no-vectorial) methods based on spatial and spectral convolution [99, 251, 259]. From the results in Table 4.13 and Table 4.14, **MLGCN** and **TP-MLGCN** bring a substantial gain w.r.t state of the art graph-based methods on the three sets, and provide comparable results with the best vectorial methods on JHMDB and SBU. On UCF, while vectorial methods are highly effective, their combination with our **MLGCN** and **TP-MLGCN** (through a late fusion) bring an extra gain despite

the fact that bridging the last few percentage gap is challenging, and this clearly shows its complementary aspect.

GCNConv [259]	90.00	Graph methods	3D skeletons SBU
ArmaConv [260]	96.00		
SGCConv [251]	94.00		
ChebyNet [99]	96.00		
Our best MLGCN	98.60		
Our best TP-MLGCN	**98.90**		
Raw coordinates [264]	49.7	Vectorial methods	
Joint features [264]	80.3		
Interact Pose [286]	86.9		
CHARM [287]	83.9		
HBRNN-L [288]	80.35		
Co-occurence LSTM [289]	90.41		
ST-LSTM [290]	93.3		
Joint line distance[280]	**99.02**		
Topological pose ordering[282]	90.5		
STA-LSTM [285]	91.51		
GCA-LSTM [284]	94.9		
VA-LSTM [291]	97.2		
DeepGRU [283]	95.7		
Riemannian manifold traj [292]	93.7		
GCNConv [259]	58.3	Graph methods	2D skeletons JHMDB
ArmaConv [260]	59.46		
SGCConv [251]	60.08		
ChebyNet [99]	59.33		
Our best MLGCN	61.21		
Our best TP-MLGCN	**61.96**		
Colorized heatmaps [195]	57.00	Vectorial methods	
P-CNN [293]	61.1		
Action Tubes [294]	**62.5**		
HLPF [199]	54.1		
JointAP [295]	61.2		
PA-AP [296]	61.5		

TABLE 4.13 – Comparison against state of the art methods for 2D and 3D skeletons.

GCNConv [259]	59.39	Video frames UCF-101 / Graph methods
ArmaConv [260]	61.11	
SGCConv [251]	62.81	
ChebyNet [99]	60.54	
Our best MLGCN	64.00	
Our best TP-MLGCN	**64.74**	
Temporal pyramid [227]	68.58	Video frames UCF-101 / Vectorial methods
Colorized heatmaps [195]	64.38	
Temporal pyramid [227] + our best MLGCN / TP-MLGCN	70.46/70.58	
colorized heatmaps [195] + our best MLGCN /TP-MLGCN	68.21/ 68.30	
Temporal pyramid [227] + colorized heatmaps [195]	77.34	
Temporal pyramid [227] + colorized heatmaps [195]+ our best MLGCN /TP-MLGCN	79.18 / 79.27	
2D two stream [15]	91.12	
2D two stream [15]+ our best MLGCN /TP-MLGCN	93.23/ 93.29	
3D appearance [28]	95.60	
3D appearance [28]+ our best MLGCN / TP-MLGCN	95.92/ 95.94	
3D motion [28]	96.41	
3D motion [28]+ our best MLGCN /TP-MLGCN	96.63/ 96.65	
3D two stream [28]	**97.94**	

TABLE 4.14 – Comparison against state of the art methods for on UCF-101 (video frames based dataset).

4.7 Conclusion

We introduced in this chapter a novel spectral GCN method referred to as **MLGCN** for action recognition. It consists in learning a graph Laplacian as a convex combination of elementary Laplacians. This **MLGCN** is then extended to tree-structured temporal pyramid referred to as **TP-MLGCN** to learn a hierarchical Laplacian. It is achieved by combining the learned Laplacians at different nodes of the temporal pyramid. The strength of our method resides in its effectiveness in learning combined Laplacian convolutional operators ; each one dedicated to a particular setting of the manifold enclosing the input graph data. Particularly, the **TP-MLGCN** variant has the ability to capture different levels of granularity yielding discriminating representations for classification. We also introduced a novel pooling process which is invariant to node permutation. This pooling process first expands nodes with their context prior to achieve global averaging. This results into representations which are more discriminating than those averaged without expansion because expansion helps to better preserve the responses of convolutional layers. Moreover, we proposed a method to build graph inputs to train GCNs from raw 2D/3D skeletons and video (rgb) frames. Finally, we introduced an augmentation for the specific task of human action recognition based on the permutation of joints and the flipping of their features.
Extensive experiments conducted on the JHMDB and SBU, as well as the challenging UCF-101 datasets, show the outperformance and also the complementary aspect of our **MLGCN/TP-MLGCN** w.r.t different baselines and the related works including graph methods.

CHAPITRE 5

CONCLUSION AND PERSPECTIVES

Contents

5.1 Summary of Contributions

In this thesis, we tackled the challenging problem of Action Recognition in videos. We investigated an approach based on the interplay between Deep Learning, Geometric Deep Learning and Kernel Methods, where we identified miscellaneous limitations, for which we have proposed some solutions. Our contribution is organized in two main points detailed below.

Hierarchical aggregation design for action recognition. Global pooling in Deep Convolutional Neural Networks plays the role of dimensionality reduction operator. It helps keeping the most informative representations and reducing the computational complexity of trained networks. However, this operator is not well suited for the specific task of action recognition in videos. In Chapter 3, we proposed a tree-structured hierarchical pooling operator that exhibits the multiple levels of temporal granularity of action categories. It consists in providing a representation that jointly captures the global description of videos and also their details. This hierarchical pooling operator has also the ability to handle the misalignments of action categories while being well localized and agnostic to video duration. We proposed two variants for learning the parameters of our hierarchical pooling function : *shallow* relying on linear/quadratic programming, and an end-to-end deep framework based on *ResNet*. Moreover, we designed a procedure that allows to efficiently train deep networks without downsampling videos and hence benefiting from the whole frames of videos. We also extended this framework by considering multiple instances of temporal pyramids to capture the complex dynamic of action categories in a fully end-to-end and differentiable manner. As a natural extension of this work, one may consider a differentiable

formulation of the constrained quadratic programming problem to learn its parameters along with those of ResNet and also replacing the activation functions and pooling operations of the temporal pyramids layer with explicit kernels.

Generalization of Deep Learning on graphs. Following the success of vectorial DL in wide spectrum of applications, its generalization to irregular domains such as graphs and manifolds has drawn a lot of attention. However, this generalization is not straightforward as it requires a careful design of pooling and convolutional operators that satisfy the properties of locality, translation invariance and equivariance, compositionality, as well as a linear computational complexity in learning as it is the case in standard ConvNets. Recently, in the geometric deep learning framework, spatial and spectral methods have been explored to achieve classification on regular grid graphs such as hand-written digits. In this thesis, we were interested in achieving the particular task of action recognition with geometric deep learning where few works have emerged. The latter are based on 2D/3D skeletons features while those operating on sequences of video frames have been less investigated in the literature especially with spectral methods. To do so, in Chapter 4, we proposed spectral convolutional and pooling operators on graphs. This spectral convolutional operator is based on convex combination of several Laplacians ; each one dedicated to a particular (possible) topology of graphs ; in order to learn a highly non-linear graph Laplacian. Then, we generalized it to a tree-structured temporal pyramid to learn a hierarchical Laplacian. It is achieved by combining the learned Laplacians at each node of the temporal pyramid. We also introduced a pooling operator that proceeds in two steps : context-dependent node expansion is achieved, followed by a global averaging. Moreover, we introduced a method to build input graphs for skeletons and video frames relying on the recent advances in human pose estimation and extraction. Finally, we designed a graph augmentation technique in order to increase the size of skeleton based action recognition datasets.

5.2 Perspectives for Future Works

At the end of this work, multiple directions seem to be worth exploring to improve action recognition models, to generalize to other tasks, and to open the blackbox of deep learning in order to build principled and interpretable models. We argue that *Graphs/Manifolds Learning* along with *non-Euclidean Geometry*, *Causality* and *Self-Supervised Learning* are one of the most important topics to be focused on during this decade as next step in AI to design commonsense knowledge mo-

dels and hence to graze *Super Intelligent Machine*.

Deep Multiple Kernel Learning. One of the extension of the work presented in Chapter 3 is the design of *Deep Multiple Non-Linear Kernels Model* to achieve action recognition on small datasets. This allows to avoid tuning the large amounts of parameters in DL (especially 3D models) and to add extra context by learning a set of different similarity functions. The advantages of incorporating kernels in DL models are multiple : these methods are theoretically well grounded and guarantee global minimum in supervised settings and since their properties are well understood, it helps to control the learning capacity of models by designing specific regularization functions for targeted tasks. Therefore, this can be a step toward DL theory.
Despite the aforementioned advantages, one of the bottlenecks of deep kernels is the difficulty to scale them up and to design hierarchical kernels that capture compositional structures as it is the case for convolutional operators satisfying some invariance properties and stability to small deformations.

Deep Graph Random Walk Kernels. Since kernel methods can be applied to different types of data including sequences, vectors and graphs, we argue that the work presented in Chapter 4 can be formulated as *Graph Kernel Learning* problem. Particularly, to design a powerful pooling operator on graphs. In Chapter 4, we modeled this operation on skeletons as the expansion of their dimensionality w.r.t their context. Similarly, we think that this pooling operation can be expressed as *graph random walk kernel* ; a finite symmetric Markov chain on undirected graphs. The rational is that the expansion step could be implicitly achieved by random walk kernels while benefiting from the interesting properties of kernels. Moreover, this design may help to add extra geometric interpretation to GCNs.

GCNs and hypergraph representations. In Chapter 4, we showed how powerful spectral methods are to achieve convolution on graphs. Benefiting from the well studied Fourier transform and the eigen-decomposition of graph Laplacians. However, the latter can not be applied to large scale graphs due to its cubic computational complexity. One way to achieve spectral convolution on these graphs is to consider hypergraph representations or graphons [297]. The latter are particularly interesting since they allow to define consistency between graphs of variable sizes thanks to their regularity property which allow to capture the structure of arbitrary large and variable size graphs.

From skeletons to generic graphs (supervoxels). In Chapter 4, we achieved action recognition with GCNs relying on skeleton graphs. The latter are estimated, and may result into inaccurate and missing joints/skeletons especially on videos

with low frame resolution and multiple interacting individuals, which makes their detection and tracking challenging. In order to circumvent that, one way to construct input graphs from video sequences is to estimate them in an unsupervised manner relying on supervoxel graphs. One of the advantages of supervoxel representations is their invariance to the number of persons and to their reordering. However, these representations have the drawbacks of being large-scale graphs which require a careful design of pooling (graph coarsening) and convolutional operators. Furthermore, it is difficult to track them across frames and this requires an appropriate design of multiple object tracking and re-identification. To do so, one may generalize optical flow algorithm on supervoxels, initially designed for pixel level displacements. In addition, following the recent advances of DL for physical processes, incorporating a prior knowledge can help to better estimate motion, including Kalman filters and density estimation processes.

Video Activity Recognition. From a practical standpoint, our models can be naturally extended to activity recognition. The latter is more general than action recognition since it is characterized by sequences of very long duration. Moreover, in activity recognition, several actions are occurring at the same time involving several persons which may result into over-crowded scenes. As a consequence, it requires deep and multiple temporal pyramids capable to discriminate the different actions that occur simultaneously. From a graph method standpoint, graph representation of activities could be an attempting way to model the complex geometric structure of over-crowed scenes.

BIBLIOGRAPHIE

[1] Spyros Makridakis. "The forthcoming Artificial Intelligence (AI) revolution: Its impact on society and firms". In : *Futures, Volume 90, Pages 46-60*. 2017.

[2] Bo-hu Li et al. "Applications of artificial intelligence in intelligent manufacturing: a review". In : *Frontiers Inf Technol Electronic Eng 18, 86–96 (2017). https://doi.org/10.1631/FITEE.1601885*. 2017.

[3] Aaron Sloman. "The computer revolution in philosophy : philosophy science and models of minds". In : *Book. Reader in philosophy and artificial intelligence, cognitive studies program, the Universtiy of Sussex*. 1978.

[4] Vincent C. Müller et Nick Bostrom. "Future progress in artificial intelligence: A survey of expert opinion". In : *in Vincent C. Müller (ed.), Fundamental Issues of ArtificialIntelligence (Synthese Library; Berlin: Springer), 553-571.* 2016.

[5] Adriana Braga et Robert K. Logan. "The Emperor of Strong AI Has No Clothes: Limits to Artificial Intelligence". In : *Information, 8, 15*. 2017.

[6] Elena Spitzer. "Tacit Representations and Artificial Intelligence: Hidden Lessons from an Embodied Perspective on Cognition". In : *In: Müller V. (eds) Fundamental Issues of Artificial Intelligence. Synthese Library (Studies in Epistemology, Logic, Methodology, and Philosophy of Science), vol 376. Springer, Cham*. 2016.

[7] Stuart Russell, Daniel Dewey et Max Tegmark. "Research Priorities for Robust and Beneficial Artificial Intelligence". In : *AI Magazine, 36(4), 105-114.*
https://doi.org/10.1609/aimag.v36i4.2577. 2015.

[8] James Hendler. "Avoiding Another AI Winter". In : *IEEE Intelligent Systems*. T. 23. 02. Los Alamitos, CA, USA : IEEE Computer Society, mar. 2008, p. 2-4. doi : 10.1109/MIS.2008.20.

[9] Leonid N. Yasnitsky. "Whether Be New "Winter" of Artificial Intelligence?" In : *In: Antipova T. (eds) Integrated Science in Digital Age. ICIS 2019. Lecture Notes in Networks and Systems, vol 78. Springer, Cham*. 2020.

[10] Chris Olah, Alexander Mordvintsev et Ludwig Schubert. "Feature Visualization". In : (2017).

[11] Kunihiko FUKUSHIMA. "Neocognitron: A self-organizing neural network for a mechanism of pattern recognition unaffected by shift in position. In Biological Cybernetics". In : (1980).

[12] Rumelhart David E, Hinton GEOFFREY, Williams Ronald J et al. "Learning representations by back-propagating errors. In Nature 323, 533–536". In : (1986).

[13] Yann LECUN et al. "Gradient based learning applied to document recognition. In Proceedings of the IEEE". In : (1998).

[14] Alex KRIZHEVSKY, Ilya SUTSKEVER et Geoffrey HINTON. "Imagenet classification with deep convolutional neural networks". In : *Advances in Neural Information Processing Systems (NIPS)*. 2012.

[15] Karen SIMONYAN et Andrew ZISSERMAN. "Two-Stream Convolutional Networks for Action Recognition in Videos". In : *Advances in Neural Information Processing Systems 27*. Sous la dir. de Z. GHAHRAMANI et al. Curran Associates, Inc., 2014, p. 568-576. URL : http://papers.nips.cc/paper/5353-two-stream-convolutional-networks-for-action-recognition-in-videos.pdf.

[16] Kaiming HE et al. "Deep Residual Learning for Image Recognition". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[17] Hamed PIRSIAVASH et Deva RAMANAN. "Detecting activities of daily living in first person camera views". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[18] Heng WANG et Cordelia SCHMID. "Action Recognition with Improved Trajectories". In : *The IEEE International Conference on Computer Vision (ICCV)*. 2013.

[19] Ivan Laptev CHRISTIAN SCHULDT et Barbara CAPUTO. "Recognizing human actions: a local svm approach". In : *The IEEE International Conference on Pattern Recognition (ICPR)*. 2004.

[20] Cordelia Schmid SVETLANA LAZEBNIK et Jean PONCE. "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2006.

[21] Dong XU et Shih-Fu CHANG. "Visual Event Recognition in News Video using Kernel Methods with Multi-Level Temporal Alignment". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2007.

[22] Alexander Klasera HENG WANG et Cordelia SCHMID. "Action Recognition by Dense Trajectories". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2011.

[23] Ying Wu JIANG WANG Zicheng Liu et Junsong YUAN. "Mining actionlet ensemble for action recognition with depth cameras". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[24] Alireza FATHI et Greg MORI. "Action recognition by learning mid-level motion features". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2008.

[25] Saad ALI et Mubarak SHAH. "Human action recognition in videos using kinematic features and multiple instance learning". In : *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. 2010.

[26] Chen LIN, Duan LIXIN et Xu DONG. "Event Recognition in Videos by Learning from Heterogeneous Web Sources". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013.

[27] Kristen GRAUMAN et Trevor DARRELL. "The pyramid match kernel: Efficient learning with sets of features". In : *Journal of Machine Learning Research (JMLR)*. 2007.

[28] Joao CARREIRA et Andrew ZISSERMAN. "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[29] Du TRAN et al. "ConvNet Architecture Search for Spatiotemporal Feature Learning". In : *arxiv*. 2017.

[30] Timur BAGAUTDINOV et al. "Social Scene Understanding: End-To-End Multi-Person Action Localization and Collective Activity Recognition". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[31] Nayyer AAFAQ et al. "Spatio-Temporal Dynamics and Semantic Attribute Enriched Visual Encoding for Video Captioning". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[32] J. ARUNNEHRU, G. CHAMUNDEESWARI et S. Prasanna BHARATHI. "Human Action Recognition using 3D Convolutional Neural Networks with 3D Motion Cuboids in Surveillance Video". In : *Procedia Computer Science, ELSEIVER*. 2018.

[33] A. CIPTADI, M. S. GOODWIN et J. M. REHG. "Movement pattern histogram for action recognition and retrieval". In : *European Conference on Computer Vision (ECCV)*. 2014.

[34] Tianhong LI et al. "Making the Invisible Visible: Action Recognition Through Walls and Occlusions". In : *IEEE International Conference on Computer Vision (ICCV)*. Oct. 2019.

[35] Daniel Weinland, Mustafa Özuysal et Pascal Fua. "Making Action Recognition Robust to Occlusions and Viewpoint Change". In : *In: Daniilidis K., Maragos P., Paragios N. (eds) Computer Vision – ECCV 2010. ECCV 2010. Lecture Notes in Computer Science, vol 6313. Springer, Berlin, Heidelberg*. 2010.

[36] Yang Wang et al. "Occlusion Aware Unsupervised Learning of Optical Flow". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2018.

[37] Sanchit Singh, Sergio A Velastin et Hossein Ragheb. "MuHAVi: A Multicamera Human Action Video Dataset for the Evaluation of Action Recognition Methods". In : *In IEEE International Conference on Advanced Video and Signal Based Surveillance*. 2007.

[38] Behrooz Mahasseni et Sinisa Todorovic. "Latent Multitask Learning for View-Invariant Action Recognition". In : *The IEEE International Conference on Computer Vision (ICCV)*. Déc. 2013.

[39] Junnan Li et al. "Unsupervised Learning of View-invariant Action Representations". In : *Advances in Neural Information Processing Systems 31*. Sous la dir. de S. Bengio et al. Curran Associates, Inc., 2018, p. 1254-1264. url : http://papers.nips.cc/paper/7401-unsupervised-learning-of-view-invariant-action-representations.pdf.

[40] Moritz Menze et Andreas Geiger. "Object Scene Flow for Autonomous Vehicles". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2015.

[41] Wenqi Ren et al. "Video Deblurring via Semantic Segmentation and Pixel-Wise Non-Linear Kernel". In : *The IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017.

[42] Seungjun Nah, Sanghyun Son et Kyoung Mu Lee. "Recurrent Neural Networks With Intra-Frame Iterations for Video Deblurring". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2019.

[43] Jonas Wulff et Michael J. Black. "Modeling Blurred Video with Layers". In : *European Conference on Computer Vision (ECCV)*. 2014.

[44] Yen-Fu.Ou, Yan Zhou et Yao Wang. "Perceptual quality of video with frame rate variation: A subjective study". In : *In IEEE International Conference on Acoustics, Speech and Signal Processing*. 2010.

[45] Ashok Veeraraghavan ; Anuj Srivastava ; Amit K. Roy-Chowdhury ; Rama Chellappa. "Rate-Invariant Recognition of Humans and Their Activities". In : *In IEEE Transactions on Image Processing, Volume: 18 , Issue: 6*. 2009.

[46] Fabian CABA HEILBRON, Juan CARLOS NIEBLES et Bernard GHANEM. "Fast Temporal Activity Proposals for Efficient Detection of Human Actions in Untrimmed Videos". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2016.

[47] Limin WANG et al. "UntrimmedNets for Weakly Supervised Action Recognition and Detection". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juil. 2017.

[48] Horaud R SANCHEZ-RIERA J. Čech J. "Action Recognition Robust to Background Clutter by Using Stereo Vision". In : *(eds) Computer Vision – ECCV 2012. Workshops and Demonstrations. ECCV 2012. Lecture Notes in Computer Science, vol 7583. Springer, Berlin, Heidelberg*. 2012.

[49] Quanzeng YOU et Hao JIANG. "Action4D: Online Action Recognition in the Crowd and Clutter". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2019.

[50] Vivek Kumar SINGH et Ram NEVATIA. "Action recognition in cluttered dynamic scenes using Pose-Specific Part Models". In : *IEEE International Conference on Computer Vision (ICCV)*. 2011.

[51] Shiyang YAN et al. "Hierarchical Multi-scale Attention Networks for action recognition". In : *In Signal Processing: Image Communication Volume 61, Pages 73-84*. 2018.

[52] Tingzhao YU et al. "Joint spatial-temporal attention for action recognition". In : *Pattern Recognition Letters Volume 112, Pages 226-233*. 2018.

[53] Razvan PASCANU, Tomas MIKOLOV et Yoshua BENGIO. "On the difficulty of training recurrent neural networks". In : *International Conference on Machine Learning (ICML)*. 2013.

[54] Olga RUSSAKOVSKY et al. "ImageNet Large Scale Visual Recognition Challenge". In : *International Journal of Computer Vision (IJCV)* 115.3 (2015), p. 211-252. DOI : 10.1007/s11263-015-0816-y.

[55] Shuiwang JI et al. "3D Convolutional Neural Networks for Human Action Recognition." In : *IEEE Trans. Pattern Anal. Mach. Intell.* T. 35. 1. 2013, p. 221-231. URL : http://dblp.uni-trier.de/db/journals/pami/pami35.html#JiXYY13.

[56] Andrej KARPATHY et al. "Large-scale Video Classification with Convolutional Neural Networks". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[57] Qing GUO et al. "Learning Dynamic Siamese Network for Visual Object Tracking". In : *The IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017.

[58] Yuan-Ting Hu, Jia-Bin Huang et Alexander G. Schwing. "Unsupervised Video Object Segmentation using Motion Saliency-Guided Spatio-Temporal Propagation". In : *The European Conference on Computer Vision (ECCV)*. Sept. 2018.

[59] Yingwei Li et al. "VLAD3: Encoding Dynamics of Deep Features for Action Recognition". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2016.

[60] David I Shuman et al. "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains". In : *in IEEE Signal Processing Magazine*. 2013.

[61] Joan Bruna et al. "Spectral networks and locally connected networks on graphs". In : *International Conference on Learning Representations (ICLR)*. 2014.

[62] Mikael Henaff, Joan Bruna et Yann LeCun. "Deep Convolutional Networks on Graph-Structured Data". In : *arXiv:1506.05163*. 2015.

[63] Michael M. Bronstein et al. "Geometric Deep Learning: Going beyond Euclidean data". In : *in IEEE Signal Processing Magazine, vol. 34, no. 4, pp. 18-42*. 2017.

[64] Palash Goyal et Emilio Ferrara. "Graph Embedding Techniques, Applications, and Performance: A Survey." In : abs/1705.02801 (2017). url : http://dblp.uni-trier.de/db/journals/corr/corr1705.html#GoyalF17.

[65] Ohn-Bar et al. "Joint Angles Similarities and HOG2 for Action Recognition". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. Juin 2013.

[66] Zhu Yu, Chen Wenbin et Guo Guodong. "Fusing Spatiotemporal Features and Joints for 3D Action Recognition". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. Juin 2013.

[67] Vemulapalli Raviteja, Arrate Felipe et Chellappa Rama. "Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2014.

[68] Y-Lan Boureau, Jean Ponce et Yann LeCun. "A Theoretical Analysis of Feature Pooling in Visual Recognition". In : *International Conference on Machine Learning (ICML)*. 2010, p. 111-118.

[69] Yang Gao et al. "Compact Bilinear Pooling". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2016.

[70] Shu KONG et Charless FOWLKES. "Low-Rank Bilinear Pooling for Fine-Grained Classification". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juil. 2017.

[71] Naila MURRAY et Florent PERRONNIN. "Generalized Max Pooling". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2014.

[72] Mateusz MALINOWSKI et Mario FRITZ. "Learning Smooth Pooling Regions for Visual Recognition". In : *British Machine Vision Conference (BMVC)*. 2013.

[73] Krikamol MUANDET, David BALDUZZI et Bernhard SCHÖLKOPF. "Domain Generalization via Invariant Feature Representation". In : *Proceedings of the 30th International Conference on Machine Learning*. Sous la dir. de Sanjoy DASGUPTA et David MCALLESTER. T. 28. Proceedings of Machine Learning Research 1. Atlanta, Georgia, USA : PMLR, 17–19 Jun 2013, p. 10-18. URL : http://proceedings.mlr.press/v28/muandet13.html.

[74] Ben S. WEBB, Timothy LEDGEWAY et Francesca ROCCHI. "Neural Computations Governing Spatiotemporal Pooling of Visual Motion Signals in Humans". In : *Journal of Neuroscience, 31 (13) 4917-4925; DOI: https://doi.org/10.1523/JNEUROSCI.6185-10.2011*. 2011.

[75] Hakan BILEN et al. "Dynamic Image Networks for Action Recognition". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2016.

[76] Tsung-Yu LIN, Aruni ROYCHOWDHURY et Subhransu MAJI. "Bilinear CNN Models for Fine-Grained Visual Recognition". In : *The IEEE International Conference on Computer Vision (ICCV)*. Déc. 2015.

[77] Bharat SINGH et al. "A Multi-Stream Bi-Directional Recurrent Neural Network for Fine-Grained Action Detection". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2016.

[78] Bingbing NI, Vignesh R. PARAMATHAYALAN et Pierre MOULIN. "Multiple Granularity Analysis for Fine-grained Action Detection". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2014.

[79] Bingbing NI, Xiaokang YANG et Shenghua GAO. "Progressively Parsing Interactional Objects for Fine Grained Action Detection". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2016.

[80] Yang ZHOU et al. "Interaction Part Mining: A Mid-Level Approach for Fine-Grained Action Recognition". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2015.

[81] Basura FERNANDO et al. "Discriminative Hierarchical Rank Pooling for Activity Recognition". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2016.

[82] Amlan KAR et al. "AdaScan: Adaptive Scan Pooling in Deep Convolutional Neural Networks for Human Action Recognition in Videos". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juil. 2017.

[83] Rohit GIRDHAR et Deva RAMANAN. "Attentional Pooling for Action Recognition". In : *Advances in Neural Information Processing Systems (NIPS)*. 2017.

[84] Michael BRONSTEIN et al. "Tutorial : Geometric deep learning on graphs and manifolds". In : *Society for Industrial and Applied Mathematics (SIAM)*. 2018.

[85] Justin GILMER et al. "Neural Message Passing for Quantum Chemistry". In : *Proceedings of the 34th International Conference on Machine Learning*. Sous la dir. de Doina PRECUP et Yee Whye TEH. T. 70. Proceedings of Machine Learning Research. International Convention Centre, Sydney, Australia : PMLR, juin 2017, p. 1263-1272. URL : http://proceedings.mlr.press/v70/gilmer17a.html.

[86] Alexander N. GORBAN et Andrei ZINOVYEV. "Principal manifolds and graphs in practice: from molecular biology to dynamical systems". In : *International Journal of Neural Systems, Vol. 20, No. 03, pp. 219-232*. 2010.

[87] Connor W. COLEY et al. "A graph-convolutional neural network model for the prediction of chemical reactivity". In : *In the Royal Society of Chemistry, DOI: 10.1039/C8SC04228D (Edge Article) Chem. Sci., 10, 370-377*. 2019.

[88] Geoffrey HINTON et al. "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups". In : *IEEE Signal Processing Magazine, Volume: 29 , Issue: 6*. 2012.

[89] Ruhi SARIKAYA, Geoffrey E. HINTON et Anoop DEORAS. "Application of Deep Belief Networks for Natural Language Understanding". In : *IEEE/ACM Transactions on Audio, Speech, and Language Processing, Volume: 22 , Issue: 4*. 2014.

[90] Bruna JOAN et Mallat STEPHANE. "Invariant Scattering Convolution Networks. In IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Vol. 35, No. 8". In : (2013).

[91] Stéphane MALLAT. "Understanding deep convolutional networks". In : *The royal society. Philosophical transactions A*. 2016.

[92] Taco COHEN et Max WELLING. "Group equivariant convolutional networks". In : *International Conference on Machine Learning (ICML)*. 2016.

[93] Mallat STEPHANE. "Group Invariant Scattering. Communications on Pure and Applied Mathematics (CPAM)". In : (2012).

[94] Austin STONE et al. "Teaching Compositionality to CNNs". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juil. 2017.

[95] Fan R. K. CHUNG. "Spectral Graph Theory". In : *Conference Board of the Mathematical Sciences (CBMS), number 92*. 1997.

[96] Petar DJURIC et Cédric RICHARD. "Cooperative and Graph Signal Processing: Principles and Applications. Book". In : 2018.

[97] David K HAMMOND, Pierre VANDERGHEYNST et Rémi GRIBONVAL. "Wavelets on graphs via spectral graph theory". In : *In Applied and Computational Harmonic Analysis, volume 30, number 2, pages 129-150*. 2011.

[98] Stephane MALLAT et Gabriel PEYRÉ. *A wavelet tour of signal processing : the sparse way*. Academic Press, 2009. ISBN : 9780123743701.

[99] Michaël DEFFERRARD, Xavier BRESSON et Pierre VANDERGHEYNST. "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering". In : *Advances in Neural Information Processing Systems 29*. Sous la dir. de D. D. LEE et al. Curran Associates, Inc., 2016, p. 3844-3852. URL : http://papers.nips.cc/paper/6081-convolutional-neural-networks-on-graphs-with-fast-localized-spectral-filtering.pdf.

[100] Antonio ORTEGA et al. "Graph signal processing: Overview, challenges, and applications". In : *In Proceedings of the IEEE, Volume 106, Number 5, Pages 808-828*. 2018.

[101] Risi KONDOR et al. "Covariant Compositional Networks For Learning Graphs". In : *International Conference on Learning Representations Workshop (ICLR-W)*. 2018.

[102] Dongmian ZOU et Gilad LERMAN. "Graph convolutional neural networks via scattering". In : *Applied and Computational Harmonic Analysis*. 2019.

[103] Sijie YAN, Yuanjun XIONG et Dahua LIN. "Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition". In : *Conference on Artificial Intelligence (AAAI)*. 2018.

[104] Christian SZEGEDY et al. "Rethinking the inception architecture for computer vision". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[105] Ivan LAPTEV. "Human action recognition". In : *INRIA Computer Vision and Machine Learning Summer School*. 2010.

[106] S. S. Beauchemin et J. L. Barron. "The computation of optical flow". In : *ACM Computing Surveys (CSUR) Surveys, Volume 27, Issue 3, Pages 433-466*. 1995.

[107] Donghao Gu et al. "Continuous Bidirectional Optical Flow for Video Frame Sequence Interpolation". In : *In IEEE International Conference on Multimedia and Expo (ICME)*. 2019.

[108] Deqing Sun, Stefan Roth et Michael J. Black. "Secrets of optical flow estimation and their principles". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2010.

[109] Li Xu, Jiaya Jia et Yasuyuki Matsushita. "Motion Detail Preserving Optical Flow Estimation". In : *In IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Volume : 34, Issue : 9*. 2012.

[110] Ivan Laptev et al. "Learning realistic human actions from movies". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2008.

[111] Edward Rosten et Tom Drummond. "Machine learning for high-speed corner detection". In : *European Conference on Computer Vision (ECCV)*. 2006.

[112] Krystian Mikolajczyk et Cordelia Schmid. "An affine invariant interest point detector". In : *European Conference on Computer Vision (ECCV)*. 2002.

[113] P. Dollár et al. "Behavior Recognition via Sparse Spatio-Temporal Features". In : *2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance. pp. 65–72.* 2005.

[114] Konstantinos G. Derpanis. "The harris corner detector." In : *York University*. 2004.

[115] A. Oikonomopoulos, I. Patras et M. Pantic. "An implicit spatiotemporal shape model for human activity localization and recognition". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009.

[116] Ahmed J. Obaid Tawfiq A. Al-asadi. "Object detection and recognition by using enhanced Speeded Up Robust Feature". In : *IJCSNS International Journal of Computer Science and Network Security, Volume 16, Number 4*. 2016.

[117] T. Lindeberg. "Feature detection with automatic scale selection". In : *International Journal of Computer Vision (IJCV)*. 1998.

[118] Navneet Dalal et Bill Triggs. "Histograms of oriented gradients for human detection". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2005.

[119] Navneet DALAL, Bill TRIGGS et Cordelia SCHMID. "Human Detection Using Oriented Histograms of Flow and Appearance". In : *European Conference on Computer Vision (ECCV)*. 2006.

[120] Salton GERARD et Michael MCGILL. "Introduction to Modern Information Retrieval". In : 1986.

[121] Sivic JOSEF et Andrew ZISSERMAN. "Video Google: A Text Retrieval Approach to Object Matching in Videos". In : *IEEE International Conference on Computer Vision (ICCV)*. 2003.

[122] Tommi JAAKKOLA et David HAUSSLER. "Exploiting Generative Models in Discriminative Classifiers". In : *Advances in Neural Information Processing Systems (NIPS)*. 1998.

[123] Lazebnik SVETLANA, Cordelia SCHMID et Jean PONCE. "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2006.

[124] Perronnin FLORENT et Christopher DANCE. "Fisher Kernels on Visual Vocabularies for Image Categorization". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2007.

[125] Florent PERRONNIN, Jorge SÀNCHEZ et Thomas MENSINK. "Improving the fisher kernel for large-scale image classification". In : *European Conference on Computer Vision (ECCV)*. 2010.

[126] Heng WANG et Cordelia SCHMID. "Action Recognition with Improved Trajectories". In : *IEEE International Conference on Computer Vision*. Sydney, Australia, 2013. URL : http://hal.inria.fr/hal-00873267.

[127] Vivek Kumar SINGH et Ram NEVATIA. "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories". In : *ICVGIP '10: Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing*. 2010.

[128] Aaron F. BOBICK et James W. DAVIS. "The recognition of human movement using temporal templates". In : *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. 2001.

[129] Yaser SHEIKH, Mumtaz SHEIKH et Mubarak SHAH. "Exploring the space of a human action". In : *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. 2005.

[130] Herbert BAY, Tinne TUYTELAARS et Luc Van GOOL. "SURF : Speeded Up Robust Features". In : *European Conference on Computer Vision (ECCV)*. 2006.

[131] Kadir TIMOR et Michael BRADY. "Saliency, Scale and Image Description". In : *International Journal of Computer Vision (IJCV)*. 2001.

[132] Samy SADEK et al. "An action recognition scheme using fuzzy log-polar histogram and temporal self-similarity". In : *EURASIP Journal on Advances in Signal Processing volume*. 2011.

[133] B. YAO, Alhaddad M.J. et Alghazzawi D. "A fuzzy logic-based system for the automation of human behavior recognition using machine vision in intelligent environments". In : *Soft Computing*. 2015.

[134] Chern Hong LIM et Chee Seng CHAN. "Fuzzy qualitative human model for viewpoint identification". In : *Neural Computation Applications, 27, 845–856*. 2016.

[135] Bardia YOUSEFI et Chu Kiong LOO. "Bio-Inspired Human Action Recognition using Hybrid Max-Product Neuro-Fuzzy Classifier and Quantum-Behaved PSO". In : *arXiv:1509.03789*. 2015.

[136] Peng HUANG, Adrian HILTON et Jonathan STARCK. "Shape similarity for 3D video sequences of people". In : *International Journal of Computer Vision (IJCV)*. 2010.

[137] Ross GIRSHICK et al. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2014.

[138] Shah Atiqur RAHMAN, David CHO et Karhang LEUNG. "Recognising human actions by analysing negative spaces". In : *IET Computer Vision, Volume: 6 , Issue: 3*. 2012.

[139] Shah Atiqur RAHMAN et al. "Fast action recognition using negative space features". In : *Expert Systems with Applications, Volume 41, Issue 2, Pages 574-587*. 2014.

[140] Shahzad CHEEMA et al. "Action recognition by learning discriminative key poses". In : *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. 2011.

[141] Jia-xin CAI, Xin TANG et Guo-can FENG. "Learning pose dictionary for human action recognition". In : *The IEEE International Conference on Pattern Recognition (ICPR)*. 2014.

[142] Alireza FATHI et Greg MORI. "Action recognition by learning mid-level motion features". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2008.

[143] SungYong CHUN et Chan-Su LEE. "Human action recognition using histogram of motion intensity and direction from multiple views". In : *IET Computer Vision, Volume 10, Number 4, Pages 250-256*. 2016.

[144] Daniel WEINLAND, Rémi RONFARD et Edmond BOYER. "Free viewpoint action recognition using motion history volumes". In : *Computer Vision and Image Understanding*. 2006.

[145] Fiza MURTAZA, Muhammad Haroon YOUSAF et Sergio A. VELASTIN. "Multi-view human action recognition using 2D motion templates based on MHIs and their HOG description". In : *IET Computer Vision, Volume 10, Number 7, Pages 758-767*. 2016.

[146] Selen PEHLIVAN et David A. FORSYTH. "Recognizing activities in multiple views with fusion of frame judgments". In : *Image and Vision Computing, Volume 32, Issue 4, Pages 237-249*. 2014.

[147] Mohiuddin AHMAD et Seong-Whan LEE. "HMM-based Human Action Recognition Using Multiview Image Sequences". In : *The IEEE International Conference on Pattern Recognition (ICPR)*. 2006.

[148] Zhuolin JIANG, Zhe LIN et Larry DAVIS. "Recognizing Human Actions by Learning and Matching Shape-Motion Prototype Trees". In : *IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 34, Number 3, Pages 533-547*. 2012.

[149] Donald R. JONES, Matthias SCHONLAU et William J. WELCH. "Efficient Global Optimization of Expensive Black-Box Functions". In : *Journal of Global Optimization 13: 455–492.* 1998.

[150] Shiwei ZHANG et al. "Group Sparse-Based Mid-Level Representation for Action Recognition". In : *Systems Man and Cybernetics : Systems IEEE Transactions on, vol. 47, no. 4, pp. 660-672*. 2017.

[151] Tanaya GUHA et Rabab K WARD. "Learning Sparse Representations for Human Action Recognition". In : *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. 2012.

[152] Haoran WANG et al. "Supervised class-specific dictionary learning for sparse modeling in action recognition". In : *Pattern Recognition. Volume 45, pages 3902–3911.* 2012.

[153] Jingjing ZHENG et al. "Cross-View Action Recognition via a Transferable Dictionary". In : *British Machine Vision Conference (BMVC)*. 2012.

[154] Fan ZHU et Ling SHAO. "Correspondence-Free Dictionary Learning for Cross-View Action Recognition". In : *In International Conference on Pattern Recognition (ICPR)*. 2014.

[155] Jinjun WANG et al. "Locality-constrained Linear Coding for Image Classification". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2010.

[156] Warren S McCulloch et Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". In : (1943).

[157] F Rosenblatt. "The perceptron: A probabilistic model for information storage and organization in the brain". In : *Psychological Review, 65(6), 386–408. https://doi.org/10.1037/h0042519*. 1958.

[158] Yann LeCun et al. "Backpropagation applied to handwritten zip code recognition". In : (1989).

[159] Kurt Hornik. "Approximation capabilities of multilayer feedforward networks". In : *Neural networks, 4(2):251–257*. 1991.

[160] Corochann. "MNIST training with Multi Layer Perceptron. https://corochann.com/mnist-training-with-multi-layer-perceptron-1149.html". In : 2017.

[161] Karen Simonyan et Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In : *International Conference on Learning Representations (ICLR)*. 2015.

[162] Christian Szegedy et al. "Going deeper with convolutions". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[163] Matthew D Zeiler et Rob Fergus. "Visualizing and understanding convolutional networks". In : *European Conference on Computer Vision (ECCV)*. 2014.

[164] Ken Chatfield et al. "Return of the Devil in the Details: Delving Deep into Convolutional Nets." In : abs/1405.3531 (2014). url : http://dblp.uni-trier.de/db/journals/corr/corr1405.html#ChatfieldSVZ14.

[165] Huang Gao et al. "Densely Connected Convolutional Networks". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[166] Zhou Lu et al. "The Expressive Power of Neural Networks : A View from the Width". In : *Advances in Neural Information Processing Systems (NIPS)*. 2017.

[167] Sergey Zagoruyko et Nikos Komodakis. "Wide Residual Networks". In : *arXiv preprint libary*. 2016.

[168] Han Dongyoon, Kim Jiwhan et Kim Junmo. "Deep Pyramidal Residual Networks". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[169] Chollet Francois. "Xception : Deep Learning With Depthwise Separable Convolutions". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[170] Xie SAINING et al. "Aggregated Residual Transformations for Deep Neural Networks". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[171] Hu JIE, Shen LI et Sun GANG. "Squeeze-and-Excitation Networks". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[172] Howard Andrew G. et al. *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. 2017.

[173] Olga RUSSAKOVSKY et al. "ImageNet Large Scale Visual Recognition Challenge". In : *International Journal of Computer Vision (IJCV)* 115.3 (2015), p. 211-252. DOI : 10.1007/s11263-015-0816-y.

[174] Ali Sharif RAZAVIAN et al. "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition." In : *CVPR Workshops*. IEEE Computer Society, 2014, p. 512-519. ISBN : 978-1-4799-4308-1. URL : http://dblp.uni-trier.de/db/conf/cvpr/cvprw2014.html#RazavianASC14.

[175] Christoph FEICHTENHOFER et al. "What Have We Learned From Deep Representations for Action Recognition?" In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2018.

[176] Christoph FEICHTENHOFER, Axel PINZ et Andrew ZISSERMAN. "Convolutional Two-Stream Network Fusion for Video Action Recognition". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2016.

[177] Joe YUE-HEI NG et al. "Beyond Short Snippets: Deep Networks for Video Classification". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2015.

[178] Limin WANG et al. "Temporal Segment Networks: Towards Good Practices for Deep Action Recognition". In : *ECCV*. 2016.

[179] Christoph FEICHTENHOFER, Axel PINZ et Richard WILDES. "Spatiotemporal Residual Networks for Video Action Recognition". In : *Advances in Neural Information Processing Systems 29*. Sous la dir. de D. D. LEE et al. Curran Associates, Inc., 2016, p. 3468-3476. URL : http://papers.nips.cc/paper/6433-spatiotemporal-residual-networks-for-video-action-recognition.pdf.

[180] Christoph FEICHTENHOFER, Axel PINZ et Richard P. WILDES. "Spatiotemporal Multiplier Networks for Video Action Recognition". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juil. 2017.

[181] Yi ZHU et al. "Hidden Two-Stream Convolutional Networks for Action Recognition". In : 2018.

[182] Azizpour Hossein et al. "Factors of Transferability for a Generic ConvNet Representation." In : *In IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 38.9, pp. 1790–1802*. 2016.

[183] Razavian Ali Sharif, Hossein Azizpour Josephine Sullivan et Stefan Carlsson. "CNN Features off-the-shelf: an Astounding Baseline for Recognition". In : *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop*. 2014.

[184] Du Tran et al. "Learning Spatiotemporal Features With 3D Convolutional Networks". In : *The IEEE International Conference on Computer Vision (ICCV)*. Déc. 2015.

[185] Laurens van der Maaten et Geoffrey Hinton. "Visualizing Data using t-SNE". In : *Journal of Machine Learning Research*. T. 9. 2008, p. 2579-2605. URL : http://www.jmlr.org/papers/v9/vandermaaten08a.html.

[186] Ali Diba et al. "Temporal 3d convnets: New architecture and transfer learning for video classification". In : *arXiv preprint arXiv:1711.08200*. 2017.

[187] Limin Wang, Yu Qiao et Xiaoou Tang. "Action Recognition With Trajectory-Pooled Deep-Convolutional Descriptors". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2015.

[188] Baccouche Moez et al. "Action Classification in Soccer Videos with Long Short-Term Memory Recurrent Neural Networks". In : *In: Diamantaras K., Duch W., Iliadis L.S. (eds) Artificial Neural Networks – ICANN 2010. ICANN 2010. Lecture Notes in Computer Science, vol 6353. Springer, Berlin, Heidelberg*. 2010.

[189] Joe Yue-Hei Ng et al. "Beyond Short Snippets: Deep Networks for Video Classification". In : *Computer Vision and Pattern Recognition*. 2015.

[190] Li Yao et al. "Describing Videos by Exploiting Temporal Structure". In : *The IEEE International Conference on Computer Vision (ICCV)*. Déc. 2015.

[191] Jeff Donahue et al. "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description". In : *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. Avr. 2017.

[192] D. Gong, G. Medioni et X. Zhao. "Structured time series analysis for human action segmentation and recognition". In : *IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 36, NO. 7*. 2014.

[193] Du Yong, Wang Wei et Wang Liang. "Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2015.

[194] Zhe CAO et al. "Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields". In : *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[195] Vasileios CHOUTAS et al. "PoTion: Pose MoTion Representation for Action Recognition". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2018.

[196] Sijie YAN, Yuanjun XIONG et Dahua LIN. "Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition". In : *Conference on Artificial Intelligence (AAAI)*. 2018.

[197] Amir Roshan Zamir KHURRAM SOOMRO et Mubarak SHAH. "UCF101: A Dataset of 101 Human Action Classes From Videos in The Wild". In : *CRCV-TR-12-01*. 2012.

[198] H. KUEHNE et al. "HMDB: a large video database for human motion recognition". In : *Proceedings of the International Conference on Computer Vision (ICCV)*. 2011.

[199] H. JHUANG et al. "Towards understanding action recognition". In : *International Conf. on Computer Vision (ICCV)*. 2013, p. 3192-3199.

[200] Kiwon YUN et al. "Two-person Interaction Detection Using Body-Pose Features and Multiple Instance Learning". In : *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE. 2012.

[201] J. SHAO et al. "Deeply learned attributes for crowded scene understanding". In : *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, p. 4657-4666.

[202] Pantic M. et al. "Human Computing and Machine Understanding of Human Behavior: A Survey". In : *In: Huang T.S., Nijholt A., Pantic M., Pentland A. (eds) Artifical Intelligence for Human Computing. Lecture Notes in Computer Science, vol 4451. Springer, Berlin, Heidelberg*. 2007.

[203] Amira Ben MABROUK et Ezzeddine ZAGROUBA. "Abnormal behavior recognition for intelligent video surveillance systems: A review". In : *Expert Systems with Applications, Volume 91, January 2018, Pages 480-491*. 2018.

[204] Yamin HAN et al. "Going deeper with two-stream ConvNets for action recognition in video surveillance". In : *In Pattern Recognition Letters Volume 107, Pages 83-90*. 2018.

[205] Bairui WANG et al. "Reconstruction Network for Video Captioning". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2018.

[206] Jingwen WANG et al. "Bidirectional Attentive Fusion With Context Gating for Dense Video Captioning". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2018.

[207] Minlong LU et al. "Deep Attention Network for Egocentric Action Recognition". In : *IEEE Transactions on Image Processing*. T. 28. 8. 2019, p. 3703-3713.

[208] Tahmida MAHMUD et al. "Prediction and Description of Near-Future Activities in Video". In : *arXiv:1908.00943*. 2019.

[209] Ivan LAPTEV et Patrick PEREZ. "Retrieving actions in movies". In : *2007 IEEE 11th International Conference on Computer Vision*. 2007, p. 1-8.

[210] Lamberto BALLAN et al. "Event detection and recognition for semantic annotation of video". In : *Multimedia Tools and Applications*. T. 51. 2010, p. 279-302.

[211] A. PATRON-PEREZ et al. "High Five: Recognising human interactions in TV shows". In : *British Machine Vision Conference (BMVC)*. 2010.

[212] Alejandro JAIMES et al. "Memory Cues for Meeting Video Retrieval". In : *Proceedings of the the 1st ACM Workshop on Continuous Archival and Retrieval of Personal Experiences*. CARPE'04. New York, New York, USA : Association for Computing Machinery, 2004, p. 74-85. ISBN : 1581139322. DOI : 10.1145/1026653.1026665. URL : https://doi.org/10.1145/1026653.1026665.

[213] Olivier DUCHENNE et al. "Automatic annotation of human actions in video". In : *2009 IEEE 12th International Conference on Computer Vision*. 2009, p. 1491-1498.

[214] Hongying MENG, Nick PEARS et Chris BAILEY. "A Human Action Recognition System for Embedded Computer Vision Application". In : *2007 IEEE Conference on Computer Vision and Pattern Recognition*. 2007, p. 1-6.

[215] Theodoros THEODORIDIS et al. "Ubiquitous robotics in physical human action recognition: A comparison between dynamic ANNs and GP". In : *2008 IEEE International Conference on Robotics and Automation*. 2008, p. 3064-3069.

[216] Yiannis DEMIRIS. "Prediction of intent in robotics and multi-agent systems". In : *Cogn Process (2007) 8: 151. https://doi.org/10.1007/s10339-007-0168-9*. 2007.

[217] Mihai NAN et al. "Human Action Recognition for Social Robots". In : *2019 22nd International Conference on Control Systems and Computer Science (CSCS)*. 2019, p. 675-681.

[218] Wei-Lwun Lu et J. J. Little. "Simultaneous Tracking and Action Recognition using the PCA-HOG Descriptor". In : *The 3rd Canadian Conference on Computer and Robot Vision (CRV'06)*. 2006, p. 6-6.

[219] B. G.Schunck. B. K.P.Horn. "Determining optical flow". In : *Artificial Intelligence, Volume 17, Issues 13, Pages 185-203*. 1981.

[220] Gabriella Csurka et al. "Visual Categorization with Bags of Keypoints". In : *European Conference on Computer Vision (ECCV)*. 2004.

[221] Csurka G. et Perronnin F. "Fisher Vectors: Beyond Bag-of-Visual-Words Image Representations". In : *In: Richard P., Braz J. (eds) Computer Vision, Imaging and Computer Graphics. Theory and Applications. VISIGRAPP 2010. Communications in Computer and Information Science, vol 229. Springer, Berlin, Heidelberg*. 211.

[222] Kristen Grauman, Trevor Darrell et Pietro Perona. "The pyramid match kernel: Efficient learning with sets of features". In : 8 (2007), p. 2007.

[223] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In : *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, p. 248-255.

[224] Alex Graves, Abdel-rahman Mohamed et Geoffrey Hinton. "Speech recognition with deep recurrent neural networks". In : *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013, p. 6645-6649.

[225] Kaiming He et al. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In : *The IEEE International Conference on Computer Vision (ICCV)*. 2015.

[226] Tianjun Xiao et al. "The application of two-level attention models in deep convolutional neural network for fine-grained image classification". In : *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, p. 842-850.

[227] Ahmed Mazari et Hichem Sahbi. "Deep Temporal Pyramid Design for Action Recognition". In : *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, p. 2077-2081.

[228] Amin Ullah et al. "Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features". In : *IEEE Access*. T. 6. 2018, p. 1155-1166.

[229] K. He et al. "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition". In : *IEEE Transactions on Pattern Analysis and Machine Intelligence*. T. 37. 9. 2015, p. 1904-1916.

[230] Ahmed Mazari et Hichem Sahbi. "Coarse-To-Fine Aggregation For Cross-Granularity Action Recognition". In : *In the 27th IEEE International Conference on Image Processing (ICIP)*. 2020.

[231] Yunbo Wang et al. "Spatiotemporal Pyramid Network for Video Action Recognition". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juil. 2017.

[232] Jiagang Zhu, Wei Zou et Zheng Zhu. "End-to-end Video-level Representation Learning for Action Recognition". In : *2018 24th International Conference on Pattern Recognition (ICPR)*. 2018, p. 645-650.

[233] Zhenxing Zheng et al. "Spatial-temporal pyramid based Convolutional Neural Network for action recognition". In : *Neurocomputing, Volume 358, 17 September 2019, Pages 446-455*. 2019.

[234] Da Zhang, Xiyang Dai et Yuan-Fang Wang. "Dynamic Temporal Pyramid Network: A Closer Look at Multi-scale Modeling for Activity Detection". In : *In: Jawahar C., Li H., Mori G., Schindler K. (eds) Computer Vision – ACCV 2018. ACCV 2018. Lecture Notes in Computer Science, vol 11364. Springer, Cham*. 2018.

[235] Ke Yang et al. "Temporal Pyramid Relation Network for Video-Based Gesture Recognition". In : *2018 25th IEEE International Conference on Image Processing (ICIP)*. 2018, p. 3104-3108.

[236] AJ Piergiovanni et Michael S. Ryoo. "Fine-Grained Activity Recognition in Baseball Videos". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. Juin 2018.

[237] Amir Shahroudy et al. "NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2016.

[238] J. Yihuang. "Pretrained 2D two streams network for action recognition on UCF-101 based on temporal segment network". In : 2017. URL : https://github.com/jeffreyyihuang/two-stream-action-recognition.

[239] Barret Zoph et al. "Learning Transferable Architectures for Scalable Image Recognition". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2018.

[240] Shawe-Taylor J. et N Cristianini. "Kernel Methods for Pattern Analysis". In : *Cambridge: Cambridge University Press. doi:10.1017/CBO9780511809682*. 2004.

[241] Mehmet Gönen et Ethem Alpaydin. "Multiple Kernel Learning Algorithms". In : *Journal of Machine Learning Research*. T. 12. 64. 2011, p. 2211-2268. URL : http://jmlr.org/papers/v12/gonen11a.html.

[242] Martin SIMONOVSKY et Nikos KOMODAKIS. "Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[243] Zonghan WU et al. "A Comprehensive Survey on Graph Neural Networks". In : *IEEE Transactions on Neural Networks and Learning Systems, Pages 1-21*. 2020.

[244] Lei SHI et al. "Skeleton-Based Action Recognition With Directed Graph Neural Networks". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2019.

[245] Kalpit THAKKAR et P J NARAYANANS. "Part-based Graph Convolutional Network for Action Recognition". In : *British Machine Vision Conference (BMVC)*. 2018.

[246] Petar VELICKOVIC et al. "Graph Attention Networks". In : *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL : https://openreview.net/forum?id=rJXMpikCZ.

[247] Federico MONTI et al. "Geometric Deep Learning on Graphs and Manifolds Using Mixture Model CNNs". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juil. 2017.

[248] Davide BOSCAINI et al. "Learning shape correspondence with anisotropic convolutional neural networks". In : *Advances in Neural Information Processing Systems 29*. Sous la dir. de D. D. LEE et al. Curran Associates, Inc., 2016, p. 3189-3197. URL : http://papers.nips.cc/paper/6045-learning-shape-correspondence-with-anisotropic-convolutional-neural-networks.pdf.

[249] Will HAMILTON, Zhitao YING et Jure LESKOVEC. "Inductive Representation Learning on Large Graphs". In : *Advances in Neural Information Processing Systems 30*. Sous la dir. d'I. GUYON et al. Curran Associates, Inc., 2017, p. 1024-1034. URL : http://papers.nips.cc/paper/6703-inductive-representation-learning-on-large-graphs.pdf.

[250] Kiran K. THEKUMPARAMPIL et al. "Attention-based Graph Neural Network for Semi-supervised Learning". In : *In arXiv:1803.03735*. 2018.

[251] Felix WU et al. "Simplifying Graph Convolutional Networks". In : *arXiv:1902.07153*. 2019.

[252] Johannes KLICPERA, Aleksandar BOJCHEVSKI et Stephan GÜNNEMANN. "Combining Neural Networks with Personalized PageRank for Classification on Graphs". In : *International Conference on Learning Representations*. 2019. URL : https://openreview.net/forum?id=H1gL-2A9Ym.

[253] Pedro Hermosilla et al. “Monte Carlo Convolution for Learning on Non-Uniformly Sampled Point Clouds”. In : *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2018)*. T. 37. 6. 2018.

[254] Michael Schlichtkrull et al. “Modeling Relational Data with Graph Convolutional Networks”. In : *arXiv preprint arXiv:1703.06103*. 2017.

[255] C Morris et al. “Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks”. In : *Conference on Artificial Intelligence (AAAI)*. 2019.

[256] Matthias Fey et al. “SplineCNN: Fast Geometric Deep Learning With Continuous B-Spline Kernels”. In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2018.

[257] Benjamin Girault, Antonio Ortega et Shrikanth Narayanan. “Irregularity-Aware Graph Fourier Transforms”. In : *IEEE Transactions on Signal Processing, Volume 66, Number 21, Pages 5746-5761*. 2018.

[258] David I Shuman, Benjamin Ricaud et Pierre Vandergheynst. “Vertex-frequency analysis on graphs”. In : *Applied and Computational Harmonic Analysis 40 (2), 260-291*. 2016.

[259] Thomas N. Kipf et Max Welling. “Semi-Supervised Classification with Graph Convolutional Networks”. In : *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. url : https://openreview.net/forum?id=SJU4ayYgl.

[260] Filippo Maria Bianchi et al. “Graph Neural Networks with Convolutional ARMA Filters”. In : *arXiv:1901.01343*. 2019.

[261] Bingbing Xu et al. “Graph Wavelet Neural Network”. In : *International Conference on Learning Representations (ICLR)*. 2019.

[262] Fernando Gama, Alejandro Ribeiro et Joan Bruna. “Diffusion Scattering Transforms on Graphs”. In : *International Conference on Learning Representations*. 2019. url : https://openreview.net/forum?id=BygqBiRcFQ.

[263] Xiaowen Dong et al. “Learning Laplacian Matrix in Smooth Graph Signal Representations”. In : *IEEE Transactions on Signal Processing, Volume 64, Number 23, Pages 6160-6173*. 2016.

[264] Kiwon Yun et al. “Two-person interaction detection using body-pose features and multiple instance learning”. In : *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. 2012.

[265] Xavier GLOROT, Antoine BORDES et Yoshua BENGIO. "Deep Sparse Rectifier Neural Networks". In : *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Sous la dir. de Geoffrey GORDON, David DUNSON et Miroslav DUDÍK. T. 15. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA : PMLR, nov. 2011, p. 315-323.

[266] Charles DUGAS et al. "Incorporating second-order functional knowledge for better option pricing". In : *Advances in Neural Information Processing Systems (NIPS)*. 2000.

[267] Christian BERG, Jens Peter Reus CHRISTENSEN et Paul RESSEL. "Harmonic analysis on semigroups: theory of positive definite and related functions". In : *Vol. 100. New York: Springer*. 1984.

[268] Isaac J SCHOENBERG. "Metric spaces and positive definite functions. In Transactions of the American Mathematical Society 44.3 (1938): 522-536". In : (1938).

[269] Andreas LOUKAS et Pierre VANDERGHEYNST. "Spectrally Approximating Large Graphs with Smaller Graphs". In : *Proceedings of the 35th International Conference on Machine Learning*. Sous la dir. de Jennifer DY et Andreas KRAUSE. T. 80. Proceedings of Machine Learning Research. 2018, p. 3237-3246.

[270] Rianne van den BERG, Thomas N. KIPF et Max WELLING. "Graph Convolutional Matrix Completion". In : *arXiv:1706.02263*. 2017.

[271] Inderjit S. DHILLON, Yuqiang GUAN et Brian Kulis AND. "Weighted Graph Cuts without Eigenvectors A Multilevel Approach". In : *IEEE Transactions on Pattern Analysis and Machine Intelligence*. T. 29. 11. 2007, p. 1944-1957.

[272] Cătălina CANGEA et al. "Towards sparse hierarchical graph classifiers". In : *arXiv:1811.01287*. 2018.

[273] Zhitao YING et al. "Hierarchical Graph Representation Learning with Differentiable Pooling". In : *Advances in Neural Information Processing Systems 31*. Sous la dir. de S. BENGIO et al. Curran Associates, Inc., 2018, p. 4800-4810. URL : http://papers.nips.cc/paper/7729-hierarchical-graph-representation-learning-with-differentiable-pooling.pdf.

[274] Oriol VINYALS, Samy BENGIO et Manjunath KUDLUR. "Order Matters: Sequence to sequence for sets". In : *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. Sous la dir. d'Yoshua BENGIO et Yann LECUN. 2016. URL : http://arxiv.org/abs/1511.06391.

[275] Muhan ZHANG et al. "An End-to-End Deep Learning Architecture for Graph Classification". In : *AAAI*. 2018, p. 4438-4445.

[276] Charles Ruizhongtai QI et al. "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space". In : *Advances in Neural Information Processing Systems 30*. Sous la dir. d'I. GUYON et al. Curran Associates, Inc., 2017, p. 5099-5108. URL : http://papers.nips.cc/paper/7095-pointnet-deep-hierarchical-feature-learning-on-point-sets-in-a-metric-space.pdf.

[277] Qiuhong KE et al. "A New Representation of Skeleton Sequences for 3D Action Recognition". In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juil. 2017.

[278] W. H. BEYER. In : *CRC Standard Mathematical Tables, 28th ed. Boca Raton, FL: CRC Press*. 1987.

[279] Mihai ZANFIR, Marius LEORDEANU et Cristian SMINCHISESCU. "The Moving Pose: An Efficient 3D Kinematics Descriptor for Low-Latency Action Recognition and Detection". In : *2013 IEEE International Conference on Computer Vision*. 2013, p. 2752-2759.

[280] Songyang ZHANG, Xiaoming LIU et Jun XIAO. "On Geometric Features for Skeleton-Based Action Recognition Using Multilayer LSTM Networks". In : *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2017, p. 148-157.

[281] D WEINLAND, R RONFARD et E BOYER. "Free viewpoint action recognition using motion history volumes". In : *In Computer vision and image understanding 104 (2-3), 249-257*. 2006.

[282] Fabien BARADEL, Christian WOLF et Julien MILLE. "Pose-conditioned Spatio-Temporal Attention for Human Action Recognition". In : *arXiv:1703.10106*. 2017.

[283] LaViola J.J. MAGHOUMI M. "DeepGRU: Deep Gesture Recognition Utility". In : *In: Bebis G. et al. (eds) Advances in Visual Computing. ISVC 2019. Lecture Notes in Computer Science, vol 11844. Springer, Cham*. 2019.

[284] Jun LIU et al. "Skeleton-Based Human Action Recognition With Global Context-Aware Attention LSTM Networks". In : *IEEE Transactions on Image Processing*. T. 27. 4. 2018, p. 1586-1599.

[285] Sijie SONG et al. "An end-to end spatio-temporal attention model for human action recognition from skeleton data". In : *Conference on Artificial Intelligence (AAAI)*. 2017.

[286] Yanli Ji, Guo Ye et Hong Cheng. "Interactive body part contrast mining for human interaction recognition". In : *2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*. 2014, p. 1-6.

[287] Wenbo Li et al. "Category-Blind Human Action Recognition: A Practical Recognition System". In : *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, p. 4444-4452.

[288] Yong Du, W. Wang et L. Wang. "Hierarchical recurrent neural network for skeleton based action recognition". In : *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, p. 1110-1118.

[289] Wentao Zhu et al. "Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks". In : *Conference on Artificial Intelligence (AAAI)*. 2016.

[290] Jun Liu et al. "Spatio-temporal LSTM with trust gates for 3D human action recognition". In : *European Conference on Computer Vision (ECCV)*. 2016.

[291] Pengfei Zhang et al. "View Adaptive Recurrent Neural Networks for High Performance Human Action Recognition From Skeleton Data". In : *The IEEE International Conference on Computer Vision (ICCV)*. 2017.

[292] Anis Kacem et al. "A Novel Geometric Framework on Gram Matrix Trajectories for Human Behavior Understanding". In : *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2018.

[293] Guilhem Chéron, Ivan Laptev et Cordelia Schmid. "P-CNN: Pose-Based CNN Features for Action Recognition". In : *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, p. 3218-3226.

[294] Georgia Gkioxari et Jitendra Malik. "Finding action tubes". In : *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, p. 759-768.

[295] Bruce Xiaohan Nie, Caiming Xiong et Song-Chun Zhu. "Joint action recognition and pose estimation from video". In : *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, p. 1293-1301.

[296] Umar Iqbal, Martin Garbade et Juergen Gall. "Pose for Action - Action for Pose". In : *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*. 2017, p. 438-445.

[297] Luana Ruiz, Luiz F. O. Chamon et Alejandro Ribeiro. "The Graphon Fourier Transform". In : *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020, p. 5660-5664.