


Numéro d'ordre : 2013-ISAL-0071 Année 2013

École Doctorale Informatique et Mathématiques de Lyon

Thèse de l'Institut National des Sciences Appliquées de Lyon

Présentée en vue d'obtenir le grade de Docteur,
spécialité : Informatique

par

Moez BACCOUCHE

Apprentissage neuronal de caractéristiques spatio-temporelles pour la classification automatique de séquences vidéo

---

Préparée à Orange Labs - France Télécom R&D, Rennes
et au laboratoire LIRIS - UMR 5205, INSA de Lyon

Soutenue publiquement le 15 juillet 2013 devant le jury composé de :

M.	Denis PELLERIN	PRU, Polytech Grenoble	Président
Mme.	Bernadette DORIZZI	PRU, Télécom SudParis	Rapporteur
M.	Jean-Marc ODOBEZ	MER, EPF Lausanne	Rapporteur
M.	Nicolas THOME	MC, UPMC Paris VI	Examinateur
M.	Franck MAMALET	Chercheur, Orange Labs Rennes	Co-encadrant
M.	Christian WOLF	MC/HDR, INSA de Lyon	Co-encadrant
M.	Christophe GARCIA	PRU, INSA de Lyon	Co-encadrant
M.	Atilla BASKURT	PRU, INSA de Lyon	Directeur

À Khaoula, à Elyes…

Remerciements

Cette thèse est le fruit d'un travail nécessitant le concours de nombreuses personnes, que je tiens à remercier.

En premier lieu, mes remerciements s'adressent à mes responsables de thèse, messieurs Franck Mamalet, Christian Wolf, Christophe Garcia et Atilla Baskurt. J'ai pris un très grand plaisir à travailler avec eux et je les remercie sincèrement pour leurs précieux conseils, la qualité de leur encadrement, leur investissement, leur écoute, et leur disponibilité. Je leur exprime ici ma gratitude pour toute l'aide qu'ils m'ont apportée au cours des différentes étapes de cette thèse. Je remercie tout particulièrement Franck, avec qui j'ai eu le plaisir de partager le même bureau pendant ces trois années.

Je tiens ensuite à remercier les membres du Jury pour avoir accepté la charge d'évaluer mon travail. Je remercie tout d'abord M. Denis Pellerin, Professeur à Polytech Grenoble, d'avoir accepté de présider mon jury de soutenance. Je remercie également Mme. Bernadette Dorizzi, Professeur à Télécom SudParis, et M. Jean-Marc Odobez, Maître d'Enseignement et de Recherche à l'École Polytechnique Fédérale de Lausanne, qui ont accepté de rapporter cette thèse, et dont les remarques constructives ont beaucoup participé à la finalisation du manuscrit. Je tiens enfin à exprimer ma gratitude à M. Nicolas Thome, Maître de Conférences à l'Université Pierre et Marie Curie, pour avoir examiné ce travail, ainsi que pour les discussions enrichissantes pendant et après la soutenance.

Je remercie également toutes les personnes que j'ai eu le plaisir de côtoyer pendant ces années. Côté Orange Labs R&D, mes remerciements s'adressent tout d'abord à messieurs Alexandre Nolle et Sid-Ahmed Berrani, qui m'ont permis d'intégrer l'unité de recherche et de développement MAS, au sein de laquelle ce travail a été effectué. Je remercie également tous les membres de l'équipe, Jean-Bernard, Patrice, Benoît, Olivier et les autres, et plus particulièrement les Doctorants et stagiaires, Khaoula, Ali, Alina, Haykel, Sonia, Gaël, Valentin et Qinglin. Côté LIRIS, mes courts séjours au sein du laboratoire ont été à chaque fois très plaisants et enrichissants, et m'ont permis parfois de prendre du recul sur ma thèse. Je tiens à remercier tous les membres des équipes IMAGINE et M2DISCO, et plus particulièrement Phuong, Jérôme, Vincent et Mingyuan.

Enfin, mes remerciements vont à mes parents, ma famille, mes amis, à tous ceux qui m'ont supporté pendant ces trois années, et tout particulièrement à mes deux anges, Khaoula et Elyes.

Résumé

Cette thèse s'intéresse à la problématique de la classification automatique des séquences vidéo. L'idée est de se démarquer de la méthodologie dominante qui se base sur l'utilisation de caractéristiques conçues manuellement, et de proposer des modèles qui soient les plus génériques possibles et indépendants du domaine. Ceci est fait en automatisant la phase d'extraction des caractéristiques, qui sont dans notre cas générées par apprentissage à partir d'exemples, sans aucune connaissance a priori.

Nous nous appuyons pour ce faire sur des travaux existants sur les modèles neuronaux pour la reconnaissance d'objets dans les images fixes, et nous étudions leur extension au cas de la vidéo.

Plus concrètement, nous proposons deux modèles d'apprentissage des caractéristiques spatio-temporelles pour la classification vidéo :

- Un modèle d'apprentissage supervisé profond, qui peut être vu comme une extension des modèles *ConvNets* au cas de la vidéo.

- Un modèle d'apprentissage non supervisé, qui se base sur un schéma d'auto-encodage, et sur une représentation parcimonieuse sur-complète des données.

Outre les originalités liées à chacune de ces deux approches, une contribution supplémentaire de cette thèse est une étude comparative entre plusieurs modèles de classification de séquences parmi les plus populaires de l'état de l'art. Cette étude a été réalisée en se basant sur des caractéristiques manuelles adaptées à la problématique de la reconnaissance d'actions dans les vidéos de football. Ceci a permis d'identifier le modèle de classification le plus performant (un réseau de neurone récurrent bidirectionnel à longue mémoire à court-terme -BLSTM-), et de justifier son utilisation pour le reste des expérimentations.

Enfin, afin de valider la généricité des deux modèles proposés, ceux-ci ont été évalués sur deux problématiques différentes, à savoir la reconnaissance d'actions humaines (sur la base KTH), et la reconnaissance d'expressions faciales (sur la base GEMEP-FERA). L'étude des résultats a permis de valider les approches, et de montrer qu'elles obtiennent des performances parmi les meilleures de l'état de l'art (avec 95, 83% de bonne reconnaissance pour la base KTH, et 87, 57% pour la base GEMEP-FERA).

Mots clés : Apprentissage de caractéristiques, modèle *ConvNet*, apprentissage profond, auto-encodage parcimonieux, classification LSTM, reconnaissance d'actions humaines, reconnaissance d'expressions faciales, reconnaissance d'actions de football.

Abstract

This thesis focuses on the issue of automatic classification of video sequences. We aim, through this work, at standing out from the dominant methodology, which relies on so-called hand-crafted features, by proposing generic and problem-independent models. This can be done by automating the feature extraction process, which is performed in our case through a learning scheme from training examples, without any prior knowledge.

To do so, we rely on existing neural-based methods, which are dedicated to object recognition in still images, and investigate their extension to the video case.

More concretely, we introduce two learning-based models to extract spatio-temporal features for video classification :

- A deep learning model, which is trained in a supervised way, and which can be considered as an extension of the popular *ConvNets* model to the video case.

- An unsupervised learning model that relies on an auto-encoder scheme, and a sparse over-complete representation.

Moreover, an additional contribution of this work lies in a comparative study between several sequence classification models. This study was performed using hand-crafted features especially designed to be optimal for the soccer action recognition problem. Obtained results have permitted to select the best classifier (a bidirectional long short-term memory recurrent neural network -BLSTM-) to be used for all experiments.

In order to validate the genericity of the two proposed models, experiments were carried out on two different problems, namely human action recognition (using the KTH dataset) and facial expression recognition (using the GEMEP-FERA dataset). Obtained results show that our approaches achieve outstanding performances, among the best of the related works (with a recognition rate of $95,83\%$ for the KTH dataset, and $87,57\%$ for the GEMEP-FERA dataset).

Keywords : Feature learning, *ConvNet* model, deep learning, sparse auto-encoder, LSTM classification, human action recognition, facial expression recognition, soccer action recognition.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction générale

Sommaire

1.1 Contexte et motivations

Avec l'avènement du numérique et le développement des technologies d'acquisition, d'archivage et de partage de photos et de vidéos, les volumes des contenus audio-visuels mis à disposition ne cessent de croître. Naviguer simplement et rechercher précisément un document Multimédia au sein de grandes collections devient un enjeu de première importance. L'utilisation de bases de données traditionnelles, nécessitant la saisie manuelle de descriptions ou de mot-clés, s'avère impossible pour décrire et surtout maintenir la description d'un tel volume de données. L'un des enjeux majeurs dans les systèmes d'information s'impose donc comme étant l'indexation et la recherche des

FIGURE 1.1 – Quelques exemples de classes, avec des niveaux sémantiques plus ou moins élevés, dans des problématiques de classification vidéo : (a) - Classification des plans d'un journal télévisé en "plateau" et "reportage" (b) - Reconnaissance des actions "touche" et "mêlée" dans des vidéos de rugby.

documents audio-visuels, et plus particulièrement des vidéos, par analyse automatique de leur contenu.

Dans ce contexte, de plus en plus d'intérêt est porté sur les méthodes pouvant extraire de manière automatique des informations sémantiques décrivant le contenu d'une vidéo. La classification vidéo s'inscrit dans ce cadre, et a pour objectif de catégoriser les séquences vidéo en leur associant des labels de niveau sémantique plus ou moins élevé. Cela peut aller d'une classification bas-niveau (comme par exemple la catégorisation des plans d'un journal télévisé en "plateau" et "reportage") à des classifications beaucoup plus fines et de haut niveau sémantique (comme par exemple les actions dans des vidéos de sport). La Figure 1.1 illustre quelques exemples de classes, avec des niveaux sémantiques plus ou moins élevés, dans des problématiques de classification vidéo.

A noter qu'au delà de l'indexation et de l'accès aux contenus, la classification vidéo trouve des applications dans beaucoup d'autres domaines, parmi lesquels nous pouvons citer :

- La vidéo-surveillance : Plusieurs applications de vidéo-surveillance liées à la classification de séquences vidéo existent, comme par exemple la détection d'évènements particuliers (comme l'abandon d'un bagage dans un aéroport ou une gare)

ou encore la détection d'un comportement anormal dans une foule.

- La vision robotique : De plus en plus de travaux en vision robotique s'intéressent, en plus des problématiques classiques de reconnaissance d'objets, à des problématiques de classification vidéo comme la reconnaissance d'actions. Un exemple récent est la plateforme VOIR[1] (*Vision and Observation In Robotics*).

- L'interaction homme-machine : La plupart des approches existantes dans ce domaine se basaient jusque là sur des instructions explicites, qui sont soient communiquées via un périphérique dédié (par exemple un *joystick* pour les jeux vidéos), ou dans le cas le plus courant explicitement par écrit. De plus en plus de nouvelles approches s'intéressent cependant à la communication homme-machine implicite, soit par la parole, soit à travers les gestes. Ce deuxième cas est directement lié à des problématiques de classification vidéo. Un exemple concret est le périphérique *Kinect* de *Microsoft*.

- Les systèmes de visio-conférence intelligents : La dernière génération de systèmes de visio-conférence offre des nouvelles fonctionnalités telles que le suivi des personnes, ou encore l'authentification automatique, qui sont toutes directement reliées à des problématiques de classification vidéo.

- La domotique : C'est l'ensemble des techniques permettant de centraliser le contrôle des différents systèmes d'une maison. La classification vidéo intervient aujourd'hui dans plusieurs applications liées à la domotique, et plus précisément dans l'adaptation des fonctionnalités du domicile aux comportements des habitants.

- L'assistance aux personnes âgées à domicile : De plus en plus de travaux s'intéressent à cette problématique pour laquelle la vision par ordinateur en général et la classification vidéo en particulier peuvent apporter beaucoup de solutions telles que la détection automatique de chutes, la vérification de la prise des médicaments, ou encore l'étude de certains symptômes comme les troubles du sommeil.

Nous allons nous intéresser dans la section suivante à la problématique et aux objectifs fixés dans le cadre de cette thèse.

[1] Plus de détails sur : http://liris.cnrs.fr/voir/

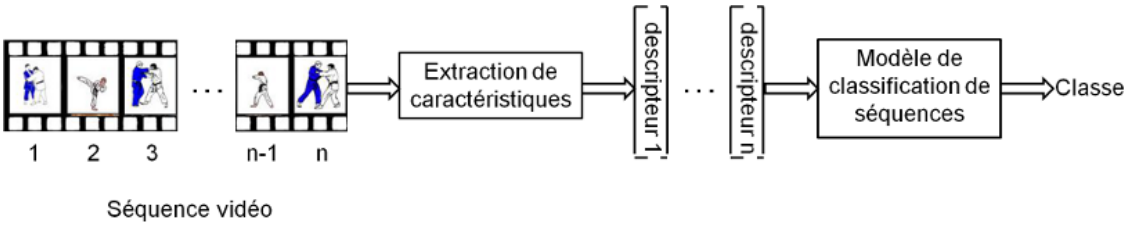

FIGURE 1.2 – Schéma général d'un modèle de classification de séquences vidéo basé sur une représentation séquentielle des données.

1.2 Problématique et Objectifs

Un modèle de classification vidéo est souvent composé de deux parties : (i) Une phase d'extraction de caractéristiques, qui consiste à extraire à partir de la vidéo des informations saillantes qui décrivent son contenu, et les encoder dans des vecteurs appelés descripteurs, et (ii) une phase de classification, qui attribue un (ou plusieurs) label(s) à la vidéo en se basant sur les descripteurs. Dans le cadre de cette thèse, nous allons nous intéresser aux modèles faisant intervenir une représentation séquentielle des données, c'est à dire que la vidéo en entrée est représentée par une séquence de vecteurs de description. Ces derniers alimentent un modèle de classification adapté aux entrées séquentielles. Le schéma général d'un tel modèle est illustré sur la Figure 1.2. A noter cependant que ce schéma n'est pas le seul possible, vu qu'un certain nombre d'approches modélisent les vidéos globalement par un traitement volumique spatio-temporel. D'autres approches proposent aussi une phase de localisation avant celle de la classification. Nous nous focalisons néanmoins dans le cadre de cette thèse sur les approches qui se basent sur le schéma de la Figure 1.2.

La phase d'extraction des caractéristiques est primordiale vu que la performance de la classification est directement liée à la pertinence de la représentation. Plusieurs approches ont été proposées dans la littérature pour l'extraction des caractéristiques pour la classification de séquences vidéo. La plupart des caractéristiques proposées sont adaptées manuellement à la problématique étudiée, réduisant ainsi considérablement la généricité de l'approche. Or, il est important, surtout dans un contexte industriel, de disposer d'un modèle qui soit indépendant du type de vidéos étudiées, et qui peut s'appliquer à des problématiques différentes.

Concernant la phase de classification, plusieurs classifieurs adaptés à la nature séquentielle des descripteurs existent, et ils sont souvent issus d'un autre domaine scientifique faisant intervenir des données séquentielles (comme par exemple le traitement audio). Même si la plupart de ces classifieurs ont déjà été utilisés pour des problèma-

tiques de classification vidéo, rares sont les travaux qui se sont intéressés à l'évaluation et la comparaison de leurs performances respectives dans le cadre d'une problématique donnée. Or, il est évident que le choix d'un classifieur qui soit le plus performant possible est une étape cruciale dans la mise en place d'un modèle complet de classification.

L'objectif de cette thèse est de proposer un ou plusieurs modèles complets de classification de séquences vidéo, qui reprennent le schéma général de la Figure 1.2. Le modèle proposé devra être générique et applicable à des problématiques différentes. Pour ce faire, la phase d'extraction de caractéristiques sera effectuée par apprentissage automatique à partir d'exemples, afin de surmonter les limites de non généricité des caractéristiques manuelles. La phase de classification reposera quant à elle sur une étude comparative préliminaire, qui devra sélectionner le classifieur de séquences le plus performant.

1.3 Contributions

La première contribution de cette thèse porte sur la problématique de la classification des vidéos de sport, pour laquelle nous introduirons une approche originale basée sur des caractéristiques spécifiques décrivant le mouvement de la caméra. Nous allons aussi mener une étude comparative qui permettra d'évaluer les performances de différents modèles de classification de séquences. Pour cette étude comparative, nous allons nous baser sur les caractéristiques visuelles introduites par Ballan et al. [BBBS09] pour les actions de football pour entraîner, dans les mêmes conditions, plusieurs modèles de classification de séquences parmi les plus populaires de l'état de l'art. Cette étude servira à sélectionner le modèle de classification le plus performant, qui sera utilisé pour le reste des expérimentations. Nous allons aussi montrer que l'introduction des caractéristiques de mouvement, spécifiquement pour ce type de séquences vidéo, a permis d'améliorer considérablement les résultats de l'état de l'art sur les données de l'étude (avec plus de 20 points d'amélioration).

Les contributions suivantes portent sur deux modèles d'apprentissage automatique de caractéristiques spatio-temporelles, qui seront combinés au modèle de classification qui aura été sélectionné. Nous avons exploré deux pistes pour l'apprentissage de ces caractéristiques spatio-temporelles :

L'apprentissage supervisé : Nous allons nous inspirer des modèles neuronaux à convolutions $2D$, qui ont été largement appliqués avec succès pour des problématiques de reconnaissance d'objets dans les images, et proposer un modèle $3D$ adapté au cas de la vidéo. Contrairement à d'autres travaux qui se sont intéressés à

cette extension, notre modèle opèrera sur les données brutes, sans faire intervenir d'autres informations ni des pré-traitements complexes. Vu qu'il est entraîné de manière supervisée, le modèle proposé prendra en compte la classe visée lors de l'apprentissage des caractéristiques, et permettra donc d'évaluer séparément son propre pouvoir discriminant, et celui du classifieur de séquences.

L'apprentissage non supervisé : Nous allons ensuite proposer un modèle d'apprentissage non supervisé et parcimonieux de caractéristiques, basé sur un schéma d'auto-encodage des données. Cette approche permet de projeter les données d'entrée dans un espace de représentation qui a la même dimension que les entrées, tout en rajoutant des contraintes de parcimonie sur les coordonnées dans ce nouvel espace de représentation. Ces coordonnées parcimonieuses seront utilisées par la suite comme caractéristiques pour entraîner le modèle de classification de séquences sélectionné.

Afin de vérifier la généricité de chacune de ces caractéristiques apprises, elles seront évaluées et comparées à l'état de l'art sur deux problématiques de classification différentes, à savoir la reconnaissance d'actions humaines et la reconnaissance d'expressions faciales. Nous allons montrer que les performances des deux modèles proposés sont parmi les meilleures de l'état de l'art sur les deux applications étudiées, même quand elles sont comparées à des approches basées sur des caractéristiques spécifiquement adaptées à la problématique.

1.4 Organisation du manuscrit

Le reste du présent manuscrit s'organise en sept chapitres, qui s'articulent autour de deux parties :

Partie I - État de l'art : Les chapitres 2 et 3 constituent la présentation de l'état de l'art des différents domaines de recherche afférents à cette thèse :

- Le chapitre 2 s'intéressera aux caractéristiques visuelles utilisées dans l'état de l'art pour entraîner les modèles de classification de séquences. La distinction sera faite entre les caractéristiques conçues manuellement en faisant intervenir beaucoup d'informations a priori sur les séquences à classer, et celles qui sont apprises automatiquement à partir des données d'entrée. Pour la première catégorie, nous présenterons les caractéristiques les plus populaires de chacun des sous-domaines

de la classification de séquences vidéo, en insistant sur le fait qu'elles sont différentes d'un domaine à un autre. Pour la seconde catégorie, nous présenterons plusieurs modèles d'apprentissage de caractéristiques, en mettant l'accent sur les modèles neuronaux à convolutions, vu qu'ils sont l'objet des principales contributions de cette thèse.

- Le chapitre 3 présentera quant à lui quelques modèles de classification de séquences parmi les plus populaires de l'état de l'art. Nous allons rappeler quelques définitions et fondements théoriques de chacun de ces modèles, ainsi que les principaux travaux qui les ont utilisés dans différentes applications liés à la classification de séquences. Certains de ces modèles ne sont pas directement adaptés aux données séquentielles, nous allons dans ce cas présenter les stratégies d'adaptation employées dans l'état de l'art.

Partie II - Contibutions : Les chapitres 4 à 7 regrouperont les différentes contributions de cette thèse qui ont été introduites dans la section 1.3. Le chapitre 4 présentera d'abord la méthode de classification de vidéos de sport que nous avons proposée, ainsi que l'étude comparative sur les différents modèles de classification de séquences, qui permettra de sélectionner le modèle le plus performant. Ensuite, dans les chapitres 5 et 6, nous introduirons deux modèles génériques d'apprentissage automatique de caractéristiques spatio-temporelles, qui sont entraînés respectivement de manière supervisée et non supervisée. Enfin, le chapitre 7 détaillera les bases étudiées et les protocoles d'évaluation utilisés, ainsi que les résultats expérimentaux permettant de comparer entre-eux les différents modèles proposés dans cette thèse ainsi qu'à l'état de l'art.

Le dernier chapitre de ce manuscrit sera enfin consacré aux conclusions, en dressant un bilan critique des principales contributions de cette thèse, et en proposant quelques pistes de travaux futurs ainsi qu'une liste des publications associées aux travaux de cette thèse.

Première partie

Définitions et état de l'art

Chapitre 2

Caractéristiques visuelles pour la classification de séquences vidéo

Sommaire

2.1 Introduction

Comme pour un grand nombre de systèmes de classification en reconnaissance de formes et en vision par ordinateur, la classification vidéo repose généralement sur deux étapes : (i) Une extraction des caractéristiques qui consiste à représenter le contenu à classer par un vecteur (ou une séquence de vecteurs) de description, et (ii) Une classification qui associe à cette représentation un label. Dans ce chapitre, nous allons nous

intéresser à la première étape, en présentant les différentes caractéristiques utilisées dans la littérature pour la classification vidéo.

Plusieurs caractéristiques ont ainsi été présentées dans l'état de l'art, qui peuvent être regroupées en deux principales catégories : La première concerne les caractéristiques dites manuelles. En réalité, le processus d'extraction en lui-même n'est pas manuel mais automatisé, mais ces caractéristiques sont généralement conçues en tenant compte des spécificités de la tâche étudiée et des données utilisées, et en faisant intervenir des connaissances a priori du domaine (d'où l'emploi abusif du terme "manuel"). Ce type de caractéristiques représente la méthodologie dominante de la classification vidéo, et obtient généralement de très bon résultats. Cependant, vu qu'ils sont spécifiquement adaptés à la tâche de classification visée, les différents domaines de la classification vidéo font intervenir des caractéristiques très différentes.

D'autres travaux proposent des caractéristiques qui ne sont pas conçues manuellement mais construites automatiquement par apprentissage à partir d'exemples. Le processus d'apprentissage ne fait intervenir aucune connaissance a priori du domaine étudié, ce qui rend ces caractéristiques très génériques, et particulièrement adaptées au cadre de cette thèse. Cependant, bien que l'état de l'art sur les modèles d'apprentissage de caractéristiques pour le cas $2D$ soit très conséquent, avec plusieurs travaux notamment en reconnaissance d'objets, l'extension de ces modèles au cas spatio-temporel n'est pas trivial, et est toujours un enjeu d'actualité.

Le reste de ce chapitre s'organisera comme suit : Dans la section 2.2, nous allons présenter les caractéristiques manuelles les plus populaires de l'état de l'art pour trois domaines différents, à savoir la reconnaissance d'actions humaines, la reconnaissance d'expressions faciales, et la classification de vidéos de sport. Nous vérifierons ainsi que les caractéristiques les plus populaires diffèrent d'un domaine à un autre. Ensuite, nous allons nous intéresser dans la section 2.3 aux modèles d'apprentissage automatique des caractéristiques pour la classification vidéo. Enfin, une conclusion sera faite dans la section 2.4.

2.2 Caractéristiques conçues manuellement

Comme mentionné précédemment, les caractéristiques manuelles sont conçues pour une problématique donnée. Cette section s'organisera donc autour de différentes problématiques, en présentant pour chacune d'elles, les caractéristiques manuelles les plus populaires de l'état de l'art.

2.2.1 Reconnaissance d'actions humaines

Les caractéristiques manuelles utilisées pour la classification d'actions humaines peuvent être regroupées en deux catégories : Des caractéristiques locales (basés sur les points d'intérêt ou sur d'autres détecteurs locaux de primitives), et les caractéristiques globales. Nous allons détailler chacune de ces catégories dans ce qui suit.

2.2.1.1 Points d'intérêts spatio-temporels

Un point d'intérêt spatio-temporel est défini comme étant une localisation dans le temps et dans l'espace d'une séquence vidéo pour laquelle le signal spatio-temporel est considéré comme "saillant". Bien qu'il n'existe pas de définition exacte de la saillance (vu qu'elle dépend souvent de l'application étudiée), ce mot est associé généralement à la présence de changements brusques simultanément dans le temps et dans l'espace. Cette définition est une extension des points d'intérêts spatiaux (par exemple les points SIFT de D. G. Lowe [Low04], les coins de Harris [HS88], les points SURF [BTVG06], ...) au cas des signaux vidéo. Dans la pratique, un détecteur de points d'intérêts calcule les maximas d'une certaine *fonction réponse* qui caractérise le signal spatio-temporel. Plusieurs détecteurs de points d'intérêts ont ainsi été présentés dans la littérature. Nous allons nous intéresser dans ce qui suit aux détecteurs les plus utilisés dans l'état de l'art.

Les coins 3D : Ce détecteur a été introduit par Laptev et Lindeberg [LL03], et est une extension du détecteur de coins de Harris [HS88] au cas spatio-temporel. Les auteurs caractérisent les points d'intérêts spatio-temporels comme étant les maximas locaux de la *fonction réponse* R définie par :

$$R = det\left(\mu\right) - k \cdot trace^3\left(\mu\right) \tag{2.1}$$

où k est un paramètre défini empiriquement (la valeur retenue dans [LL03] est de 5×10^{-4}), et μ est une matrice 3×3 appelée "tenseur de structure", qui est définie pour chaque pixel (x, y, t) d'intensité $I\left(x, y, t\right)$ par :

$$\mu\left(x, y, t\right) = \begin{pmatrix} \frac{\partial I}{\partial x}^2 & \frac{\partial I}{\partial x} \cdot \frac{\partial I}{\partial y} & \frac{\partial I}{\partial x} \cdot \frac{\partial I}{\partial t} \\ \frac{\partial I}{\partial x} \cdot \frac{\partial I}{\partial y} & \frac{\partial I}{\partial y}^2 & \frac{\partial I}{\partial y} \cdot \frac{\partial I}{\partial t} \\ \frac{\partial I}{\partial x} \cdot \frac{\partial I}{\partial t} & \frac{\partial I}{\partial y} \cdot \frac{\partial I}{\partial t} & \frac{\partial I}{\partial t}^2 \end{pmatrix} \tag{2.2}$$

A noter que dans [LL03], la matrice μ est lissée par une Gaussienne spatio-temporelle afin, d'une part, de réduire le bruit induit par la dérivation, et d'autre part, pouvoir caractériser les échelles spatiales et temporelles des coins détectés.

FIGURE 2.1 – Exemples de détection de points d'intérêts par le détecteur de coins $3D$ de Laptev et Lindeberg [LL03]. Figure extraite de [LL03].

Une fois les points d'intérêts détectés, leurs voisinages spatio-temporels sont décrits en calculant les dérivées Gaussiennes normalisées appliquées aux échelles utilisées lors de la détection. Les vecteurs de description ainsi obtenus sont utilisés par la suite comme caractéristiques lors de classification.

La Figure 2.1 illustre un exemple de détection de points d'intérêts par le détecteur de coins $3D$ de Laptev et al. sur une vidéo de la base d'actions humaines KTH (qui est une base standard de l'état de l'art, et qui sera présentée plus en détails dans le chapitre 7). Sur la Figure 2.1, nous pouvons voir que le nombre de points détectés pour cette action est assez faible. Ceci représente d'ailleurs la principale limite de ce détecteur, et a motivé l'introduction du détecteur qui va être présenté dans ce qui suit.

Les "points périodiques" : Ce détecteur a été introduit par Dollár et al. [DRCB05] et se base sur des filtres Gaussiens (appliqués dans le domaine spatial) et des filtres de Gabor (appliqués dans le domaine temporel) pour sélectionner les patchs spatio-temporels (ou les *cuboïdes*) qui maximisent localement une certaine *fonction réponse* R, définie pour chaque *cuboïde* I par :

$$R = (I \otimes g \otimes h_1) + (I \otimes g \otimes h_2) \tag{2.3}$$

où g est une Gaussienne 2D, h_1 et h_2 sont des filtres de Gabor 1D, et $\otimes$ désigne l'opérateur de convolution.

Pour chaque *cuboïde* ainsi sélectionné, les auteurs associent un vecteur de description qui correspond à la concaténation des gradients des pixels du *cuboïde*, qui subit ensuite une réduction de sa dimension par une analyse en composantes principales. Ce vecteur de description est utilisé comme vecteur de caractéristiques pour la classification.

La Figure 2.2 montre un exemple de résultat obtenu en appliquant le détecteur

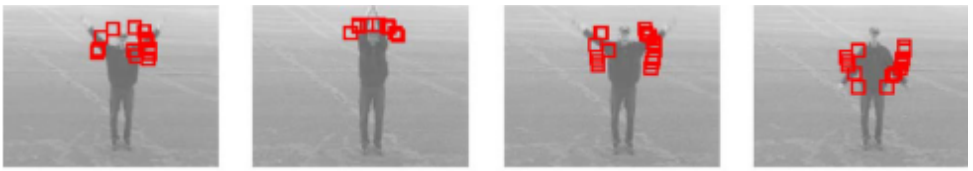

FIGURE 2.2 – Exemples de détection de points d'intérêts par le détecteur de points périodiques de Dollár et al. [DRCB05]. Figure extraite de [NWFF08].

de points périodiques sur une vidéo de la base d'actions humaines KTH. En comparaison avec la Figure 2.1, nous pouvons remarquer que le nombre de points détectés est plus important.

Le détecteur Hessien : Ce détecteur a été introduit par Willems et al. [WTVG08], et est une extension du détecteur de "*blobs*" de Lindeberg et al. [Lin98] au cas de la vidéo. La *fonction réponse* R utilisée par les auteurs est définie par :

$$R = |det\,(H)| \tag{2.4}$$

où H est la matrice Hessienne qui regroupe pour chaque pixel (x, y) à chaque instant t les dérivées partielles de second ordre dans le temps et dans l'espace de la fonction d'intensité $I\,(x, y, t)$:

$$H\,(x, y, t) = \begin{pmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial x \partial t} \\ \frac{\partial^2 I}{\partial y \partial x} & \frac{\partial^2 I}{\partial y^2} & \frac{\partial^2 I}{\partial y \partial t} \\ \frac{\partial^2 I}{\partial t \partial x} & \frac{\partial^2 I}{\partial y \partial t} & \frac{\partial^2 I}{\partial t^2} \end{pmatrix} \tag{2.5}$$

La *fonction réponse* exprimée par l'équation 2.4 est une "mesure de saillance", qui permet d'attribuer à chaque pixel un score d'autant plus fort que le contenu spatio-temporel autour de ce pixel est pertinent. Les auteurs proposent donc d'exploiter cette propriété afin de contrôler le nombre de points d'intérêts détectés. La Figure 2.3 illustre ce principe sur une séquence vidéo de la base d'actions humaines KTH.

Afin de décrire les points d'intérêt détectés, les auteurs proposent une extension du descripteur SURF de Bay et al. [BTVG06] : Pour chaque volume $3D$ autour d'un point détecté, le vecteur de description correspond à la somme pondérée des réponses de trois filtres de Haar correspondant chacun à une direction (x, y et t). Ces vecteurs de description sont utilisés comme caractéristiques pour la classification.

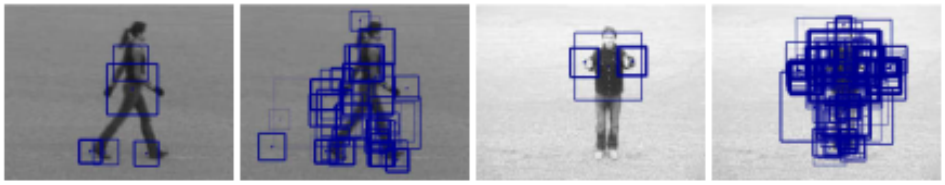

FIGURE 2.3 – Exemples de détection de points d'intérêts par le détecteur Hessien de Willems et al. [WTVG08], avec une illustration de l'influence du seuil de réponse sur le nombre de points d'intérêt détectés. Figure extraite de [WTVG08].

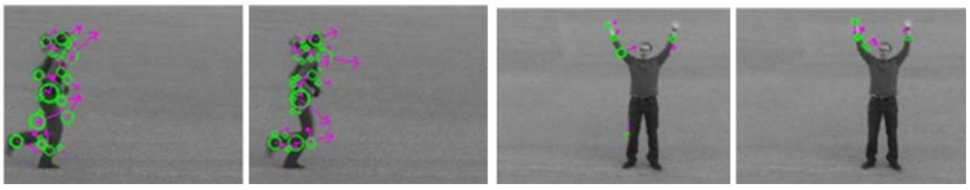

FIGURE 2.4 – Exemple de détection de points d'intérêts par le détecteur MoSIFT de Chen Et Hauptmann [CH09]. Figure extraite de [CH09].

Les MoSIFT : Les points d'intérêts MoSIFT (*Motion SIFT* en anglais) ont été introduits par Chen et Hauptmann [CH09]. Contrairement à ce que leur nom pourrait indiquer, les MoSIFT ne sont pas une extension des points d'intérêts 2*D* SIFT de D. Lowe [Low04] au cas de la vidéo, mais détectent les points SIFT 2*D* qui présentent des mouvements conséquents. De plus, contrairement aux trois détecteurs présentés précédemment, le processus de détection de points d'intérêt MoSIFT ne repose pas sur une maximisation d'une *fonction réponse*. L'algorithme présenté par les auteurs se déroule en deux étapes : (i) L'algorithme de détection de points d'intérêts SIFT est appliqué pour chaque image de la vidéo afin d'extraire l'information spatiale saillante, et (ii) un critère de mouvement (se basant sur le calcul du flot optique) est appliqué sur le voisinage spatio-temporel de chacun de ces points 2*D* détectés, et ceux présentant des mouvements "importants" sont retenus. En ce qui concerne la description des points détectés, les auteurs proposent un descripteur combinant les descripteurs SIFT et le flot optique. Pour plus de détails, se référer à l'article de Chen et Hauptmann [CH09]. La Figure 2.4 illustre un exemple de détection de points d'intérêts MoSIFT sur deux séquences de la base KTH.

Autres détecteurs : Hormis ces quatre détecteurs qui sont les plus utilisés dans l'état de l'art, d'autres approches (moins connues) ont été proposées. Oikonomopoulos et al. [OPP05] ont étudié l'extension des détecteurs 2*D* de Kadir et Brady [KB03]

au cas de la vidéo. Ils définissent ainsi un terme d'énergie caractérisant les changements locaux du signal à différentes échelles. Dans [WC07], Wong et Cipolla proposent un critère d'information globale pour extraire les zones de l'image qui présentent les mouvements les plus importants. Klaser et al. [KMS+08] se basent quand à eux sur les histogrammes de gradients $3D$ orientés. D'autres approches reposent sur des chaînes de traitement plus complexes. A titre d'exemple, nous pouvons citer les travaux de Bregonzio et al. [BGX09] qui appliquent des filtres de Gabor sur un nombre important d'images de différence (entre des instants choisis aléatoirement dans la séquence), afin de caractériser les instants et les localisations pour lesquels les changements sont importants.

Outre les approches proposées pour la détection des points d'intérêts, d'autres travaux se sont intéressés à la description du voisinage spatio-temporel des points détectés. Plusieurs descripteurs ont ainsi été proposés dans la littérature (hormis ceux qui ont été cités précédemment), parmi lesquels nous pouvons citer le descripteur HOG/HOF de Laptev et al. [LMSR08] (qui est le plus couramment utilisé dans l'état de l'art, et qui se base sur le calcul d'histogrammes de flots optiques et d'orientations du gradient sur des volumes $3D$ locaux), le descripteur HOG-3D de Klaser et al. [KMS+08], ou encore le descripteur SIFT-3D de Scovanner et al. [SAS07].

Après avoir passé en revue les points d'intérêts spatio-temporels les plus populaires de l'état de l'art, il est important de noter que ces caractéristiques dites "locales" (par opposition aux autres types de caractéristiques qui décrivent le contenu spatio-temporel global de la vidéo, et qui seront présentées au paragraphe 2.2.1.2) sont très rarement utilisées directement pour la classification. En effet, ceci est dû à trois raisons principales : (i) Le nombre de points d'intérêts détectés est généralement très élevé, et avec une forte redondance, (ii) ce nombre ainsi que la taille des images varient d'une séquence à l'autre, ce qui conduit à des vecteurs de description de tailles variables, alors que les classifieurs que nous décrirons dans le chapitre 3 ne gèrent que des vecteurs de caractéristiques de tailles fixes par instant de la séquence, et (iii) il n'existe pas d'ordre défini sur les points détectés permettant de les mettre directement dans un vecteur.

Afin de remédier à ces problèmes, plusieurs solutions ont été proposées dans l'état de l'art, dont la plus courante est celle des sacs de mots [SM86]. Concrètement, une fois les points d'intérêts extraits, un dictionnaire de "mots" spatio-temporels est généré en appliquant un algorithme de partitionnement (typiquement l'algorithme des k-moyennes) sur les vecteurs de description. Le choix du nombre de partitions (c'est à dire la taille du dictionnaire) est crucial vu qu'il est directement lié au niveau de quanti-

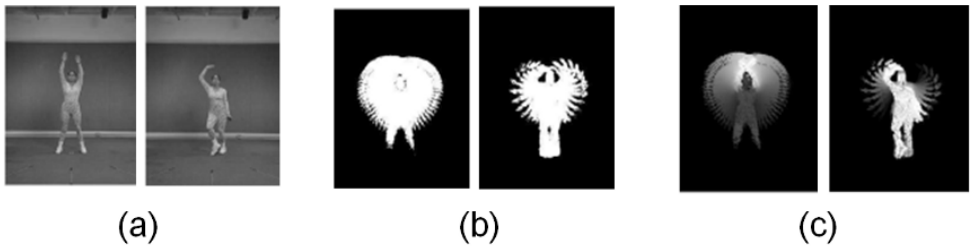

FIGURE 2.5 – Illustration des caractéristiques globales pour la classification vidéo : (a) - Images clés (b) - MEIs (c) - MHIs. Figure extraite de [WRB10].

fication de l'information, et donc aux performances de la classification. Ensuite, chaque séquence vidéo est représentée par un histogramme décrivant la fréquence d'apparition de chaque mot spatio-temporel dans la vidéo. Cette approche a été testée avec succès en combinaison avec chacun des points d'intérêts spatio-temporels décrits précédemment [SLC04, DRCB05, NWFF08, WTVG08, SDNFF08] pour des problématiques de reconnaissance d'actions humaines.

Dans le paragraphe suivant, nous allons nous intéresser à d'autres approches (dites "globales") qui, contrairement aux points d'intérêts, décrivent le contenu spatio-temporel global de la vidéo.

2.2.1.2 Caractéristiques globales

Hormis les approches locales basées sur les points d'intérêts, d'autres méthodes proposent des caractéristiques qui encodent les images dans leur globalité. Souvent, le principe utilisé pour ce dernier cas est de générer une image (appelée aussi "carte") qui représente le contenu de la séquence vidéo, et de se baser sur cette image pour la classification. Deux catégories d'approches basées sur ce principe ont été proposées dans la littérature :

Les images clés : Cette méthode a été introduite par Carlsson et Sullivan [CS01]. L'idée est de sélectionner l'une des images de la vidéo pour représenter toute la séquence, et de se comparer à cette image pour déterminer la classe d'une séquence vidéo donnée (cf. Figure 2.5-(a)). Ce principe a été étendu par la suite par Schindler et Van Gool [SVG08] pour sélectionner plusieurs images clés au lieu d'une seule. Néanmoins, ces approches basées sur la sélection d'une ou de plusieurs images clés n'exploitent pas l'information de mouvement, et ont des performances assez faibles en terme de classification.

Les images d'énergie et d'historique du mouvement : Respectivement MEI (pour *Mo-*

tion Energy Images) et MHI (pour *Motion History Images*) ont été introduites par Bobick et Davis [BD96, BD01]. Les MEIs sont des cartes binaires qui encodent les emplacements du mouvement (cf. Figure 2.5-(b)). Les MHIs sont quand à elles des images en niveau de gris qui représentent l'historique du mouvement durant l'action (cf. Figure 2.5-(c)). Les MEIs et les MHIs sont calculées en accumulant les images de différence sur un certain nombre d'instants consécutifs. Les auteurs associent à ces images un vecteur de description basé sur les moments de Hu et les moments de Zernike [BD96, BD01]. D'autres travaux ont utilisé des descripteurs différents pour les MEIs et les MHIs. A titre d'exemple, nous pouvons citer les travaux de Lv et Nevatia [LN07] qui utilisent des descripteurs de forme. A noter aussi que le principe des MHIs a été étendu par Weinland et al. dans [WRB06] au cas des multi-caméras, dans lequel une silhouette $3D$ qui représente l'historique volumétrique du mouvement remplace l'image MHI.

Toutes ces caractéristiques globales présentent plusieurs points positifs. Tout d'abord, elles sont moins coûteuses que les approches locales. Ensuite, elles permettent de décrire la forme globale de l'objet/de l'action, ce qui représente un réel avantage par rapport aux approches locales qui nécessitent de modéliser les relations spatiales/spatio-temporelles entre les points (ce qui s'avère compliqué et peu robuste aux différentes variations). Les approches globales permettent aussi de tirer profit de certaines informations géométriques comme les silhouettes des personnes, la forme...

Néanmoins, le principal point faible de ces approches est leur manque de robustesse vis-à-vis des occultations, des zooms et des bruits, vu qu'elles ne disposent pas d'informations sur les différentes parties de la scène qu'elles modélisent. De plus, elles nécessitent généralement certains pré-traitements (soustraction de fond, suivi d'objets...) qui sont souvent plus complexes que la classification.

Après avoir passé en revue les caractéristiques locales et globales les plus utilisées dans l'état de l'art de la reconnaissance d'actions humaines, nous allons nous intéresser dans la sous-section suivante à un autre sous-domaine de la classification vidéo, à savoir celui de la reconnaissance d'expressions faciales.

2.2.2 Reconnaissance d'expressions faciales

Plusieurs types de caractéristiques ont été présentés dans la littérature pour la reconnaissance d'expressions faciales. Elles peuvent être regroupées en deux catégories selon qu'elles décrivent l'apparence ou bien la géométrie du visage. Nous allons présenter les approches les plus populaires pour chacune de ces catégories dans ce qui suit.

2.2.2.1 Caractéristiques d'apparence

Les motifs binaires locaux : Un motif binaire local (LBP pour *Local Binary Pattern*) est un opérateur non-paramétrique qui décrit la structure spatiale locale d'une image. Cet opérateur a été initialement introduit par Ojala et al. et a démontré un fort pouvoir discriminant pour la classification d'images de textures [OPH96]. Depuis, les LBPs ont été appliqués dans plusieurs domaines notamment en indexation d'images [HS03], en détection d'objets [HP06], en imagerie médicale [OLFM07] ou encore en télédétection [LSF05]. Mais leur importante robustesse aux variations d'illumination les rendent particulièrement adaptés à l'analyse des visages.

Considérons un pixel donné (x, y) d'une iamge I, à qui est associé un voisinage circulaire $\Omega(x, y, R)$ centré en (x, y) et de rayon R. Le motif binaire local en (x, y) par rapport au voisinage $\Omega(x, y, R)$ est alors exprimé par :

$$LBP_R(x, y) = \sum_{(x', y')_i \in \Omega} 2^i \cdot \chi(x, y, x', y') \tag{2.6}$$

où χ est la fonction binaire de Heaviside, définie par :

$$\chi(x, y, x', y') = \begin{cases} 0 \; si \; I(x, y) \prec I(x', y') \\ 1 \; si \; I(x, y) \geq I(x', y') \end{cases} \tag{2.7}$$

Pour chaque pixel de l'image d'entrée, le code LBP (calculé par l'équation 2.6) encode des informations sur la distribution des motifs locaux (contours, zones planes/pics d'intensité, ...) sur le voisinage de ce pixel. Un histogramme de fréquence de ces codes LBP est généralement généré afin de caractériser l'image entière. Pour le cas des visages, vu que les relations spatiales entre les motifs locaux sont particulièrement importantes, les histogrammes ne sont pas calculés sur l'image entière mais sur des patchs, permettant ainsi d'encoder aussi bien le contenu des motifs que leurs emplacements. Le vecteur de description correspondant à l'image complète est obtenu simplement en concaténant les descripteurs des différents patchs (cf. Figure 2.6).

Cette représentation a été largement utilisée en analyse de visages, notamment pour la reconnaissance de personnes [AHP04, HPA04], mais aussi la reconnaissance d'expressions : Par exemple dans [FHP04], Feng et al. utilisent des caractéristiques LBP combinées à une optimisation linéaire pour classer 7 expressions faciales à partir d'images fixes. La même approche a été utilisée dans [SGM05]

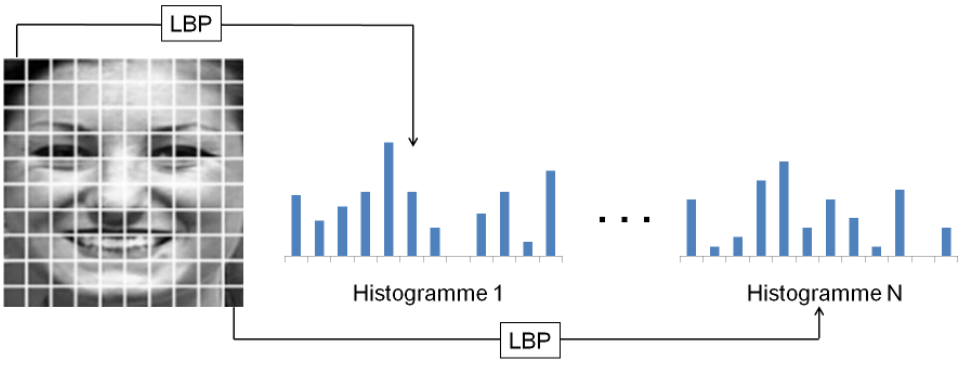

FIGURE 2.6 – Illustration de la génération du descripteur LBP global d'une image par concaténation des histogrammes LBP correspondant à chacun des patchs.

avec un classifieur SVM. D'autres travaux en reconnaissance d'expressions faciales ont proposé de ne pas calculer les caractéristiques LBPs directement sur les images en niveau de gris, mais sur des transformées telles que l'image du gradient dans [LFCY06] ou encore la transformée en ondelettes dans [HZZH06].

Plusieurs travaux plus récents se sont intéressés au cas de la vidéo. L'approche la plus simple a été proposée par Shan et al. [SGM09], dans laquelle les caractéristiques LBPs sont calculés sur une seule image de la vidéo. Valstar et al. [VJM+11] ont proposé de calculer ces caractéristiques sur toutes les images de la vidéo, de les classer une à une avec un classifieur SVM, et d'utiliser un système de vote pour attribuer le label final. D'autres travaux ont proposé d'adapter l'opérateur aux signaux $2D + t$. Par exemple dans [ZP07], Zhao et Pietikainen ont proposé les LBPs volumiques (VLBP), qui permettent de capturer les textures dynamiques dans un voisinage $3D$ sphérique autour d'un pixel donné. les séquences vidéos sont alors décrites de la même manière que pour les LBPs classiques, mais en remplaçant les patchs $2D$ par des blocs spatio-temporels $3D$.

Les filtres de Gabor : Un filtre de Gabor est un filtre linéaire dont la réponse impulsionnelle est une sinusoïde modulée par une gaussienne. Nous avons déjà évoqué les filtres de Gabor $1D$ au cours du paragraphe 2.2.1.1, puisque le détecteur de points d'intérêts spatio-temporel introduit par Dollar et al. [DRCB05] utilise ces filtres dans le domaine temporel. Le cas $2D$ quant à lui a été introduit par J. G. Daugman dans [Dau85], et a été depuis utilisé dans plusieurs applications telles que la biométrie [LW99], l'indexation d'images [WJKB00, ZWIL00] ou encore la reconnaissance de caractères [CSB01]. Mais ces filtres ont connu un essor particulier avec les travaux de Lyons et al. en analyse de visages, et plus particulièrement en

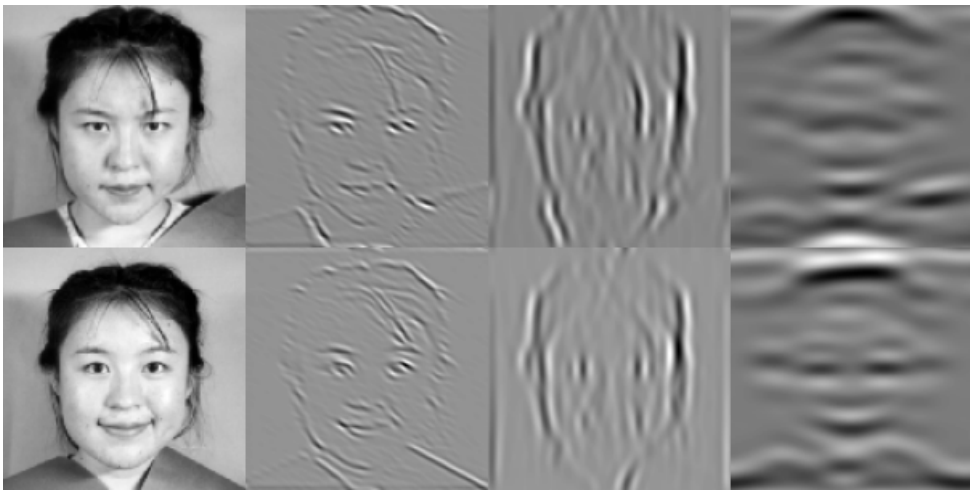

FIGURE 2.7 – Exemples de réponses d'un banc de filtres de Gabor $2D$ (avec trois valeurs d'orientations et longueurs d'ondes différentes), appliqué à deux images d'expressions faciales. Figure extraite de [LAKG98].

reconnaissance d'expressions faciales [LAKG98]. A noter que les travaux de Hubel et Wiesel [HW62] ont démontré l'existence d'une ressemblance entre les filtres de Gabor et ceux utilisés dans le cortex visuel des mammifères.

La réponse d'un banc de filtre de Gabor $2D$ s'exprime en chaque pixel (x, y) par :

$$G_j(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \cdot exp\left[-\frac{1}{2}\left(\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_x^2}\right)\right] \cdot exp\left[j\frac{2\pi x'}{\lambda}\right] \tag{2.8}$$

où σ_x et σ_y sont les écarts types le long des deux directions horizontale et verticale, λ est la longueur d'onde (en pixels) du filtre, et $[x'\ y']$ est un vecteur défini par :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.9}$$

où θ est l'orientation du filtre. Souvent, un banc de filtres "classique" comporte 8 orientations et 5 longueurs d'ondes [FLo3]. La Figure 2.7 montre quelques exemples de résultats obtenus en appliquant un banc de filtres de Gabor $2D$ à des images d'expressions faciales. Le vecteur de caractéristiques utilisé pour la classification est obtenu en deux étapes : (i) En représentant chacune des images de réponse par un vecteur en concaténant les lignes, puis (ii) en concaténant les différents vecteurs obtenus pour obtenir un vecteur de description qui est généralement de très grande dimension. A noter que d'autres travaux plus récents proposent de réduire la dimension de ce vecteur en calculant l'image de la réponse

moyenne [LL08]. Une fois ces vecteurs de caractéristiques calculés, un modèle de classification est entrainé à attribuer un label à l'image ou à la vidéo (cf. chapitre 3). Plusieurs modèles ont été utilisés dans la littérature pour la reconnaissance d'expressions faciales, parmi lesquels nous citons les K plus proches voisins dans [LAKG98, LL08], les SVMs dans [BLB+02, LBL07, WBR+11], ou encore l'optimisation linéaire dans [GD05].

Autres caractéristiques d'apparence : Les motifs locaux binaires et les filtres de Gabor (présentés précédemment) sont les caractéristiques d'apparence les plus utilisées de l'état de l'art pour la reconnaissance d'expressions faciales dans les images et les vidéos [PB07]. Cependant, d'autres travaux se basent sur d'autres caractéristiques. Par exemple, dans [WO06] Whitehill et Omlin proposent de remplacer les filtres de Gabor par des ondelettes de Haar. D'autres approches proposent d'utiliser des caractéristiques initialement introduites pour la reconnaissance d'actions humaines, et de les appliquer au cas des expressions faciales. Par exemple dans [VPP04], Valstar et al. calculent les images d'historique du mouvement (MHI) de Bobick et Davis [BD96, BD01] (présentées au paragraphe 2.2.1.2) combinés à un classifieur par K plus proches voisins. Une approche similaire a été présentée par Essa et Pentland dans [EP97], mais qui se base sur les MEIs. Les performances restent néanmoins inférieures à celles obtenues par les approches basées sur les LBPs et les filtres de Gabor.

2.2.2.2 Caractéristiques géométriques

Par opposition aux caractéristiques d'apparence, qui se basent sur l'analyse des textures des visages (ainsi que leurs mouvements) pour décrire les expressions faciales, une autre catégorie de caractéristiques, dites géométriques, localisent des points saillants du visage et décrivent les relations spatiales et/ou spatio-temporelles qui existent entre eux. Contrairement aux points saillants décrits dans le paragraphe 2.2.1.1, ceux-ci ne correspondent pas à des maximas d'une *fonction réponse* mais à des points caractéristiques du visage comme les coins des yeux, des sourcils, de la bouche et du nez (un exemple est illustré sur la Figure 2.8). L'intérêt des caractéristiques géométriques vient du fait que les positions relatives de ces parties du visage, ainsi que leurs mouvements, sont très caractéristiques des expressions faciales.

Plusieurs techniques ont ainsi été présentées pour détecter ces points d'intérêts sur les visages. Les plus connues (et les plus utilisées) sont les "modèles actifs d'apparence" (AAM pour *Active Appearence Models*), introduits par Cootes et al. dans [CET01]. Le

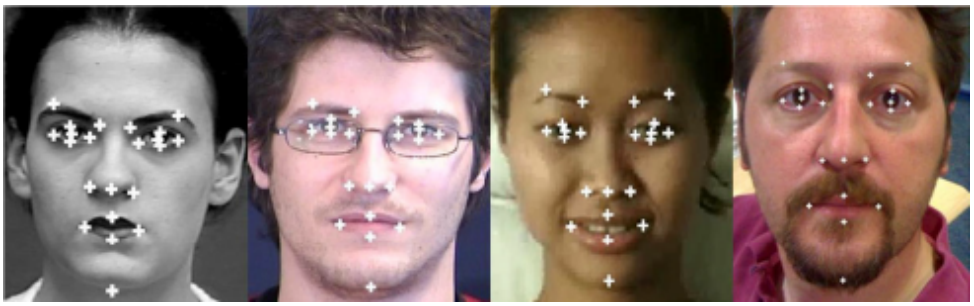

FIGURE 2.8 – Illustration des caractéristiques géométriques issues du détecteur de points saillants de Vukadinovic et Pantic [VP05]. Figure extraite de [VP05].

principe des AAMs est de générer, par apprentissage à partir d'exemples, un modèle statistique caractérisant l'apparence des visages, et d'effectuer une mise en correspondance entre ce modèle et l'image à traiter afin d'en extraire les points d'intérêt.

D'autres détecteurs ont été introduits par la suite parmi lesquels nous citons celui de Viola et Jones [VJ01] (qui se base sur des caractéristiques pseudo-Haar construites à partir des sorties des ondelettes de Haar), celui de Duffner et Garcia [DG05] (basé sur un modèle neuronal à convolutions), celui de Vukadinovic et Pantic [VP05] (illustré sur la Figure 2.8, et basé sur une analyse statistique des sorties des filtres de Gabor), celui de Chang et al. [CHFT06] (qui se base sur une analyse des contours), ou encore celui de Kotsia et Pitas [KP07] (basé sur un modèle dit "filaire", *wire-frame* en anglais).

Une fois ces points d'intérêts détectés, leurs localisations sont utilisées pour former les vecteurs de caractéristiques. La représentation la plus simple (et la moins utilisée) consiste à coder directement dans un vecteur les positions relatives de ces points par rapport à un repère donné, et de normaliser les valeurs ainsi obtenues par rapport à la taille de l'image. Cette approche a été notamment utilisée dans [VP05] et dans [Fas06], mais a démontré des performances limitées en terme de classification. La plupart des travaux représentent plutôt les expressions faciales par des vecteurs de déplacement de ces points d'intérêts détectés, afin de décrire les mouvements locaux des différentes parties du visages lors d'une expression donnée. L'estimation de ces vecteurs de déplacement est effectuée par des algorithmes de suivi plus ou moins complexes : DeCarlo et Metaxas [DM96] ont présenté un algorithme de suivi basé sur une modélisation de la forme du visage (les contours) et une mise en correspondance des points détectés. Des travaux plus récents [TKC01, GBTG02, CRA+04] ont ensuite proposé d'utiliser l'algorithme d'estimation du flot optique de Lucas et Kanade [LK81] afin de calculer les valeurs du déplacement correspondant aux points d'intérêts. Plus récemment, et afin de remédier à certaines limitations liées à l'estimation du flot optique (notamment la

sensibilité au bruit, aux changements d'illuminations ainsi qu'aux occultations), plusieurs travaux ont utilisé des algorithmes de suivi plus sophistiqués basés sur les filtres de Kalman [ZJ05, GJ05] ou encore les filtres particulaires [PP05, VP06], afin de suivre l'évolution dans le temps des points détectés.

Enfin, notons que certains travaux (par exemple ceux de Pantic et Patras [PP05], ou encore ceux de Lucey et al. [LAC07]) ont étudié la possibilité d'utiliser conjointement ces caractéristiques géométriques avec des caractéristiques d'apparence, et ont démontré que ces caractéristiques "hybrides" obtenaient des meilleures performances qu'en utilisant chacune d'entre elles séparément.

2.2.3 Classification de séquences vidéo de sport

Les vidéos de sport représentent un type de contenus particulièrement intéressant à traiter de part les enjeux commerciaux qui y sont liés. Plusieurs travaux se sont ainsi intéressés à la classification automatique de séquences vidéo de sport, dans le cadre d'applications très variées comme les résumés automatiques, l'indexation, ou encore la structuration de flux vidéo.

Nous allons nous intéresser dans cette partie uniquement aux approches qui se basent sur des caractéristiques visuelles. Il est important de noter toutefois qu'il existe des approches qui utilisent d'autres informations (surtout l'audio et les textes) pour la classification de vidéos de sport, souvent en complément des informations visuelles. Par exemple, pour le cas des textes, Babaguchi et al. [BKK02] proposent d'extraire des mots clés à partir des sous-titres présents sur des vidéos de football américain, et d'utiliser cette information afin d'affiner la classification basée sur des caractéristiques visuelles. Pour les approches utilisant le signal audio, nous pouvons citer les travaux de Tjondronegoro et al. [TCP04], qui proposent de détecter, à l'aide de caractéristiques audio bas-niveau, les instants de la vidéo où l'arbitre siffle. Les auteurs proposent de combiner cette information supplémentaire avec des caractéristiques visuelles afin d'affiner la classification des segments vidéo en "phases de jeu" et "pause". Cette approche a été validée sur plusieurs sports différents (football, basketball, rugby, natation, ...) et une amélioration considérable des performances de la classification a été constatée. Leonardi et al. [LMP04] proposent quant à eux d'exploiter des caractéristiques audio similaires afin de réduire le nombre de fausses alarmes dans une application de détection de buts dans des vidéos de football. Des approches similaires, mais se basant sur des caractéristiques haut-niveau (les coefficients cepstraux pour des vidéos de baseball dans [RGA00], ou encore la densité spectrale de puissance pour des vidéos de tennis dans [DKRD03]) ont

aussi été introduites.

Nous allons présenter dans ce qui suit les principaux travaux sur l'extraction des caractéristiques visuelles pour la classification de séquences de sport. Contrairement au cas des actions humaines et des expressions faciales, il n'existe pas pour ce type de contenus de méthodologie dominante ou de caractéristiques populaires sur lesquelles se sont basés plusieurs travaux (comme les points d'intérêts spatio-temporels pour la reconnaissance d'actions ou encore les LBP pour les expressions faciales). Nous avons néanmoins identifié deux types d'approches : (i) Celles qui visent à extraire des informations de bas-niveau sémantique (cf. paragraphe 2.2.3.1), et (ii) celles qui qui visent à extraire des informations plus haut-niveau (cf. paragraphe 2.2.3.2).

2.2.3.1 Caractéristiques pour la classification de bas-niveau sémantique

Cette première catégorie de travaux a pour but principal de structurer les vidéos et non d'en extraire des informations sémantiquement riches. La plupart de ces travaux se basent sur des primitives visuelles bas niveau. Par exemple, dans [ZC01], les auteurs utilisent seulement la couleur pour structurer des vidéos de tennis et de baseball en détectant des évènements de type "service" pour le tennis et le "lancer" pour le baseball. Dans [XXC+01], Xu et al. proposent une méthode de classification des vidéos de football qui se base sur une pré-classification de chaque image selon son angle de prise de vue (vue globale, zoom ou gros plan). Cette pré-classification est faite en analysant la couleur dominante de chaque image. Un système de vote est ensuite utilisé pour classer les séquences en deux catégories : "Phase de jeu" et "pause". Xie et al. [XXC+04] ont ensuite apporté une amélioration à cette méthode en utilisant les HMMs (cf. chapitre 3) pour modéliser les transitions entre les différents angles de vue, ce qui a permis d'améliorer les résultats par rapport à [XXC+01]. D'autres travaux ont présenté des approches similaires pour la classification des séquences en "phase de jeu" et "pause" en se basant sur l'angle de prise de vue. Par exemple dans [ETM03], Ekin et al. proposent de rajouter d'autres informations sur la durée minimale d'une phase de jeu ou encore l'écart moyen entre deux phases de jeu consécutives.

Quelques autres travaux ont essayé de proposer des classifications plus fines. Par exemple dans [ABCDB02], Assfalg et al. proposent de classer les séquences en trois catégories : "Vue du terrain", "vue du joueur" ou "vue du public". Les caractéristiques utilisées par les auteurs sont : (i) Des histogrammes d'intensité des contours, et (ii) des histogrammes de longueur et orientation des segments (cf. Figure 2.9). Ces caractéristiques sont extraites à partir de l'image clé de chaque séquence. Cette technique a été

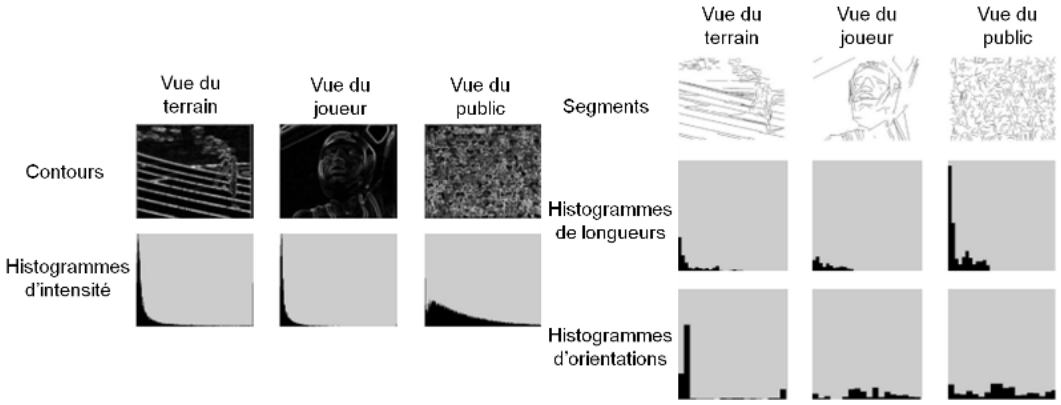

FIGURE 2.9 – Caractéristiques utilisées par Assfalg et al. dans [ABCDB02]. Figure extraite de [ABCDB02].

appliquée à 10 sports différents (individuels et collectifs) et donne des résultats satisfaisants malgré la simplicité des caractéristiques utilisées.

Notons que la plupart des travaux cités précédemment ne font pas intervenir le mouvement. Une autre famille de travaux, s'intéresse à l'analyse des trajectoires et le suivi d'objets pour extraire des informations statistiques sur le jeu. Par exemple dans [PJC98] pour le tennis et [Gueo2] pour le baseball, les auteurs se basent sur le suivi des mouvements des joueurs et de la balle pour extraire des informations sur les principales zones du terrain occupées par le jeu, ou encore le joueur le plus mobile.

2.2.3.2 Caractéristiques pour la classification de haut-niveau sémantique

Ce que nous pouvons conclure des travaux cités dans le paragraphe précédent c'est que l'utilisation de caractéristiques visuelles de "bas niveau" seules ne permet pas d'extraire des informations sémantiquement riches comme les actions ou les évènements. Des approches plus sophistiquées ont ainsi été introduites. La plupart d'entre-elles font intervenir des informations a priori sur le sport étudié, et des connaissances du domaine. Par exemple dans [GLC95], Gong et al. utilisent un modèle du terrain ainsi que la position des joueurs et celle de la balle pour classer les séquences vidéos de football (plus précisément les images clés de chaque séquence) en 15 catégories différentes sémantiquement très riches. Une approche similaire a été aussi introduite dans [MIoo] pour les vidéos de tennis et qui intègre, en plus d'un modèle du court, un modèle des lignes et du filet. Dans [TLD+05], Tong et al. proposent une méthode de classification dans laquelle des primitives visuelles bas-niveau (texture, couleur, mouvement, ...) sont combinés à des informations a priori, telles que les relations spatiales entre le terrain, les joueurs et le public. Les auteurs évaluent l'apport de ces informations a priori et

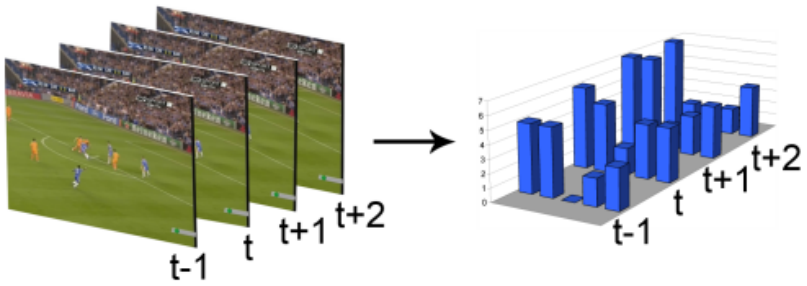

FIGURE 2.10 – Caractéristiques utilisées par Ballan et al. dans [BBBS09] pour la classification de vidéos de football. Chaque vidéo est représentée par une séquence d'histogramme de mots visuels. Figure extraite de [BBBS09].

montrent qu'ils améliorent considérablement les performances de la classification. Dans [ZVK00], Zhou et al. proposent d'utiliser les vecteurs de déplacement estimés par l'algorithme de compression MPEG-1, et de les combiner à des caractéristiques visuelles de contours et de couleur dominante. Ces caractéristiques sont calculées sur l'image clé de chaque séquence et sont ensuite combinées à un arbre de décision intégrant des règles définies par l'utilisateur (et donc des informations a priori). Ce modèle a été évalué sur une base de vidéos de basketball comprenant 9 classes sémantiquement riches. Une approche similaire a été présentée dans [TSKR00] avec un arbre de décision différent, et pour une autre base de basketball.

A noter toutefois que quelques approches essaient d'extraire des informations de haut niveau sémantique en se basant uniquement sur le contenu sans faire intervenir des informations a priori. Par exemple, dans [NPZ02], Ngo et al. utilisent des histogrammes de couleur et de mouvement calculés sur des courtes séquences (correspondant à des plans) pour regrouper ceux qui ont des descripteurs bas-niveau similaires. Cette méthode a été testée sur des vidéos de basketball et les résultats sont peu satisfaisant. Plus récemment, Ballan et al. [BBBS09] ont proposé une autre approche basée sur les sacs de mots visuels. Les auteurs représentent une séquence vidéo par une séquence d'histogrammes de mots visuels, ces derniers étants extraits par un *clustering* k-moyennes sur un ensemble de descripteurs SIFT calculés sur les images de toutes les vidéos (cf. Figure 2.10). Ces séquences d'histogrammes sont ensuite utilisés pour entraîner un classifieur SVM avec un noyau adapté à la classification de séquences. Les expérimentations ont été effectuées sur une base de 100 vidéos de football comprenant 4 classes et le taux de classification obtenu est de 73,25% (nous étudierons plus en détails cette approche dans le chapitre 4).

2.3 Modèles d'apprentissage automatique de caractéristiques

Après avoir passé en revue les différentes caractéristiques manuelles utilisées pour chacun des sous-domaines de la classification de séquences vidéo, et après avoir constaté que les caractéristiques les plus populaires varient d'un domaine à un autre, nous allons nous intéresser dans la section suivante à une autre catégorie de caractéristiques, qui se distinguent de celles présentées précédemment par le fait qu'elles sont générées sans aucune connaissance à priori du domaine, mais par apprentissage automatique à partir d'exemples. Nous allons dans un premier temps présenter un bref état de l'art sur les modèles d'apprentissage de caractéristiques, pour des problématiques autres que la classification vidéo. Nous nous intéresserons ensuite à quelques modèles d'apprentissage pour lesquels l'application au cas de la vidéo a été étudiée.

2.3.1 État de l'art

Les modèles d'apprentissage de caractéristiques peuvent être regroupés en deux catégories : (i) Ceux qui sont entraînés de manière supervisée, c'est à dire qui apprennent des caractéristiques discriminantes entre les classes en faisant intervenir les labels lors de la phase d'apprentissage, et (ii) ceux qui sont entraînés de manière non supervisée, en apprenant une représentation décrivant les structures sous-jacentes des données, sans faire intervenir les classes, et donc sans chercher à délimiter leurs frontières. Pour cette dernière catégorie, une phase de classification (supervisée) suit généralement la phase d'apprentissage (non supervisé), en se basant sur la représentation apprise.

Plusieurs modèles d'apprentissage supervisé de caractéristiques ont ainsi été proposés dans la littérature. Nous aborderons plus en détail lors de la sous-section 2.3.3 une catégorie d'approches d'apprentissage supervisé très populaire, à savoir les modèles neuronaux. Mais plusieurs autres travaux se sont aussi intéressés à cette problématique. Par exemple, Goldberger et al. [GRHS04] ont présenté une approche, appelée NCA (pour *Neighborhood Components Analysis*), qui permet d'apprendre une transformation linéaire entre l'espace de représentation des données d'entrée et un espace de caractéristiques de dimension plus faible, en préservant les composantes voisines caractérisant chacune des classes. L'espace de caractéristiques est construit de manière à maximiser les performances en terme de classification d'une recherche des k plus proches voisins (k-ppv) selon une mesure de distance donnée. Une approche similaire a ensuite été présentée par Sugiyama [Sug07], qui se base sur l'analyse discriminante de Fisher, et qui est adaptée aux données multi-modales. Salakhutdinov et Hinton [SH07a] ont aussi introduit une variante de la NCA, dans laquelle la transformation apprise est non linéaire, ce qui

permet à la classification k-ppv d'être encore plus discriminante. Une approche similaire, appelée DrLIM (pour *Dimensionality Reduction by Learning an Invariant Mapping*) a été introduite par Hadsell et al. [HCL06], dont l'objectif est de minimiser la variance intra-classe et de maximiser la variance inter-classes des caractéristiques apprises. La transformation apprise est non linéaire, mais aussi invariante aux translations et aux changements d'illumination pour une problématique de reconnaissance d'objets dans les images.

Une deuxième catégorie de travaux se sont intéressés à l'apprentissage des caractéristiques, mais de manière non supervisée. Nous pouvons citer dans ce sens des exemples de modèles très populaires tels que l'analyse en composantes principales [Jol86], le *clustering* par k-moyennes ou encore la mise à l'échelle multi-dimensionnelle [CC94], qui ont été largement utilisés pour différentes problématiques.

Une autre famille de modèles d'apprentissage non supervisé des caractéristiques est basée sur une décomposition hiérarchique des parties apprises à partir des données d'entrée. L'apprentissage se fait couche par couche, les couches supplémentaires étant construites en combinant les couches précédentes (voir [FBL09] pour un exemple en reconnaissance d'objets).

D'autres approches proposent de projeter les données d'entrée dans un espace de représentation, puis de les reconstruire à partir des coordonnées de la projection, en minimisant l'erreur de reconstruction. Ces coordonnées (appelées généralement *code*) sont ensuite utilisées comme caractéristiques. Plusieurs approches basées sur ce principe ont été proposées dans la littérature, parmi lesquelles nous pouvons citer les auto-encodeurs neuronaux [RHW86] (pour lesquels la taille du code est faible), et les différentes variantes de codeurs parcimonieux [OF97, LS99, AEB05, RPCL06, RHBL07] (pour lesquelles les codes sont de grande dimension, mais avec une majorité de valeurs nulles). A noter que les machines de Boltzmann restreintes [Smo86, Hin02], que nous allons présenter plus en détails dans la sous-section 2.3.2, se basent aussi sur ce schéma d'auto-encodage.

Nous allons nous intéresser dans ce qui suit aux modèles d'apprentissage qui ont été utilisés afin d'apprendre des caractéristiques spatio-temporelles pour des problématiques de classification vidéo. Nous allons présenter dans les sous-sections 2.3.2 et 2.3.3 deux modèles parmi les plus populaires de l'état de l'art, et dont l'extension au cas de la vidéo a été étudiée dans la littérature. Nous allons ensuite aborder dans la sous-section 2.3.4 quelques autres approches.

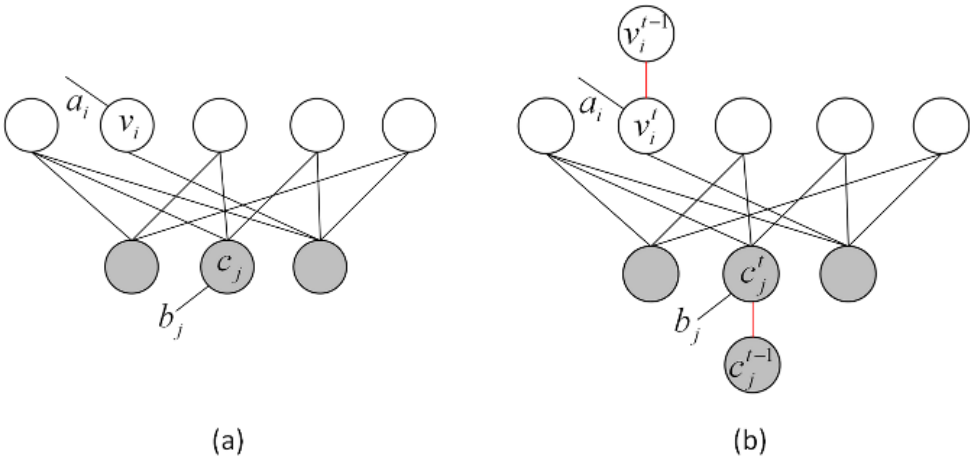

FIGURE 2.11 – (a) - Machine de Boltzmann restreinte [Smo86] (b) - Machine de Boltzmann restreinte temporelle [SHo7b].

2.3.2 Machines de Boltzmann restreintes

Les machines de Boltzmann restreintes (RBM pour *Restricted Boltzmann Machines*) sont des réseaux de neurones stochastiques génératifs qui ont été introduits par P. Smolensky [Smo86], puis popularisés plus récemment par les travaux de G. E. Hinton qui a proposé un algorithme d'apprentissage rapide [Hino2], ainsi qu'une version multi-couches appelée DBN (pour *Deep Belief Network*) [HOTo6, HSo6].

Nous allons dans un premier temps nous intéresser aux fondements théoriques et architecturaux des RBMs standards de P. Smolensky [Smo86], puis à quelques variantes utilisées pour la classification vidéo.

Un modèle RBM standard se compose d'un ensemble de neurones connectés entre-eux (que nous appellerons *unités*). Chaque unité fournit une décision en tenant compte de l'apport des autres unités. La Figure 2.11-(a) illustre l'architecture standard des RBMs : Une couche d'unités dites *visibles* (illustrées par des ronds blancs) connectée à une couche d'unités *cachées* (illustrées par des ronds gris). Les unités cachées fournissent des décisions binaires, alors que les unités visibles peuvent avoir des valeurs réelles. Les unités sont reliées par des connexions pondérées, dont les paramètres sont calculés durant l'apprentissage. A noter qu'il n'y a pas de connexions reliant les unités cachées (ou visibles) entre-elles. Cette restriction est d'ailleurs à l'origine du nom des RBMs, par rapport aux machines de Boltzmann standards (c'est à dire non restreintes), pour lesquelles toutes les connexions sont autorisées.

Si nous notons par v_i l'état d'activation de l'unité visible i, et par c_j celui de l'unité cachée j, les RBMs assignent une probabilité pour chaque configuration jointe des unités

visibles v et des unités cachées c :

$$P(v,c) = \frac{exp\left[-E(v,c)\right]}{Z} \tag{2.10}$$

où $E(v,c)$ est une fonction d'énergie (qui sera définie ci-après), et Z est une constante de normalisation appelée fonction de partition. A noter que cette dernière est analogue à celle utilisée pour les champs conditionnels aléatoires cachés (cf. section 3.2 du chapitre 3), et plus généralement pour les champs de Markov aléatoires (*Markov Random Fields* - MRF en anglais). En particulier, un RBM peut être considéré comme un cas particulier d'un MRF ayant une structure graphique et une fonction d'énergie spécifiques.

Le terme d'énergie $E(v,c)$ présent dans l'équation 2.10 est défini par :

$$E(v,c) = -\sum_{i,j} w_{ij} v_i c_j - \sum_i a_i v_i - \sum_j b_j c_j \tag{2.11}$$

où w_{ij} est le poids qui pondère la connexion entre i et j, a_i est le biais de l'unité i et b_j est celui de l'unité j.

Ce modèle est souvent entraîné par l'algorithme de divergence contrastive de Hinton et al. [Hino2], qui se base sur un schéma d'auto-encodage, et qui vise à minimiser la fonction d'énergie exprimée par l'équation 2.11. Concrètement, et sans rentrer dans les détails, la mise à jour d'un poids w_{ij} donné est exprimée, pour chaque exemple d'apprentissage, par :

$$\Delta w_{ij} = \epsilon \cdot \left[\left\langle v_i c_j \right\rangle_{données} - \left\langle v_i c_j \right\rangle_{reconstruction}\right] \tag{2.12}$$

où ϵ est le taux d'apprentissage (*learning rate* en agnlais), et :

- Le terme $\left\langle v_i c_j \right\rangle_{données}$ désigne la fréquence avec laquelle l'unité visible i et l'unité cachée j sont activées mutuellement, quand le réseau est stimulé (au niveau de la couche visible) avec les données d'apprentissage. Les états d'activation des unités cachées sont dans ce cas obtenus par :

$$P(c_j = 1|v) = \sigma\left(a_j + \sum_i w_{ij} v_i\right) \tag{2.13}$$

  où σ est une fonction sigmoïde.

- Le terme $\left\langle v_i c_j \right\rangle_{reconstruction}$ désigne la fréquence avec laquelle l'unité visible i et l'unité cachée j sont activées mutuellement, quand le réseau est stimulé (au niveau des couches cachées) avec des données reconstruites. Celles-ci étant obtenues en

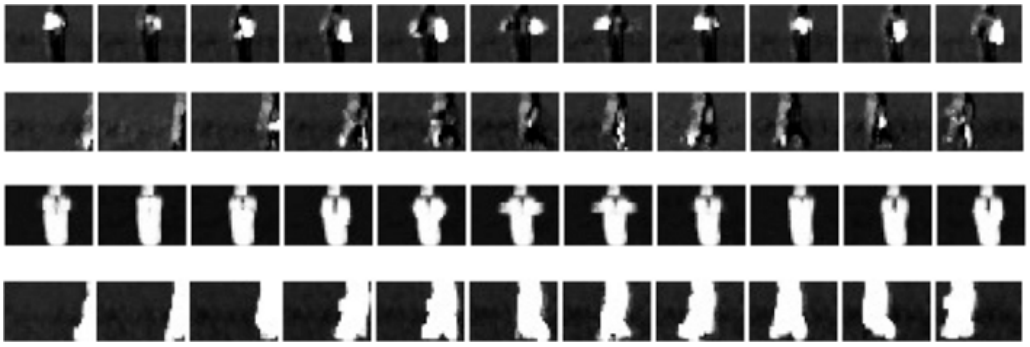

FIGURE 2.12 – Illustration de quelques caractéristiques apprises automatiquement par le modèle basé sur les RBMs présenté par Taylor et al. [TFLB10], appliqué à deux actions différentes. La première caractéristique (les deux premières lignes) semble encoder les parties du corps en mouvement, et la deuxième (les deux dernières lignes) semble segmenter le personnage et le fond. Figure extraite de [TFLB10].

calculant la probabilité conditionnelle :

$$P(v_i = 1|c) = \sigma\left(b_i + \sum_j w_{ij} c_j\right) \tag{2.14}$$

A noter que l'équation 2.14 est valable pour le cas binaire, et qu'une équation similaire, faisant intervenir une fonction Gaussienne, existe pour le cas des données réelles.

Cette procédure est ainsi répétée pour tous les exemples d'apprentissage, jusqu'à la convergence, c'est à dire quand l'erreur de reconstruction est inférieure à un certain seuil.

L'utilisation des RBMs pour le traitement vidéo n'a été étudiée que récemment avec les travaux de Sutskever et Hinton [SH07b]. Les auteurs ont proposé un modèle RBM temporel (appelé tRBM, pour *Temporal* RBM), qui consiste simplement en un RBM classique dans lequel l'information temporelle relative aux instants passés est incorporée en tant que biais supplémentaire (cf. Figure 2.11-(b)). Les auteurs ont proposé deux algorithmes d'apprentissage correspondant respectivement aux versions standards et multi-couches des tRBMs. Ces deux modèles ont été entrainés à générer des séquences vidéos à partir d'exemples synthétiques simples.

Un modèle similaire a aussi été introduit par Taylor et al. [THR07] qui utilise des valeurs réelles (et non binaires) pour les unités visibles, ce qui permet de modéliser des mouvements plus complexes telles que les actions humaines. Néanmoins, ces modèles (qui ont été conçus à l'origine pour traiter des mouvements acquis par des capteurs) n'ont pas été utilisés pour la classification, et opèrent sur des images de taille réduite,

représentées ligne par ligne sous forme de vecteurs. Plus récemment, Lee et al. [LGRN09] ont proposé un modèle convolutionnel basé sur les RBMs, qui permet (à travers le partage des poids) d'opérer sur des images de taille normale sans pour autant augmenter le nombre de paramètres et la complexité du modèle. Cette approche a été ensuite étendue au cas spatio-temporel par Taylor et al. [TFLB10] qui ont proposé un modèle similaire, mais qui opèrent sur des paires d'images successives d'une vidéo. Les auteurs ont évalué ce modèle pour une application de reconnaissance d'actions humaines, et démontrent que, d'une part, les résultats obtenus sont au niveau de l'état de l'art, et d'autre part, que les caractéristiques extraites par apprentissage (illustrées sur la Figure 2.12) sont visuellement pertinentes.

2.3.3 Réseaux de neurones à convolutions

Nous allons nous intéresser dans cette sous-section aux modèles neuronaux à convolutions. Nous allons commencer par rappeler les fondements des Perceptrons multi-couches. Nous nous intéresserons ensuite aux modèles neuronaux à convolutions $2D$, ainsi qu'à leur extension au cas de la vidéo.

2.3.3.1 Perceptrons multi-couches

Nous allons dans ce paragraphe nous intéresser au réseau de neurones le plus utilisé dans l'état de l'art, à savoir le Perceptron [Ros57], et plus précisément sa version multi-couches [RHW86]. A noter toutefois qu'il existe d'autres types de réseaux de neurones (par exemple les réseaux RBF [BL88], les réseaux de Kohonen [Koh88] ou encore les réseaux de Hopfield [Hop82]) que nous n'allons pas aborder ici. Nous suggérons au lecteur intéressé de se référer aux ouvrages de S. Haykin [Hay99] et de C. Bishop [Bis06].

Un Perceptron multi-couches (MLP pour *Multi-Layer Perceptron*) peut être vu comme un ensemble d'unités de traitement, appelés *noeuds* ou *neurones*, reliées entre elles par des connections pondérées. Les poids de ces connections étant les paramètres du modèle. Ces neurones et ces connections sont organisés en couches : (i) La première couche est appelée *couche d'entrée*, (ii) la dernière est appelée *couche de sortie* et (iii) la ou les couches du milieu sont appelées *couches cachées* (voir la Figure 2.13). Les neurones de ces couches cachées, ainsi que ceux de la couche de sortie appliquent deux traitements : (i) Une combinaison linaire de leurs entrées (dont les poids sont des paramètres du réseau), suivie par (ii) une fonction non-linéaire appelée *fonction d'activation*. Les deux fonctions les plus courantes sont la *tangente hyperbolique* et la *fonction sigmoïde*. Dans ce qui suit,

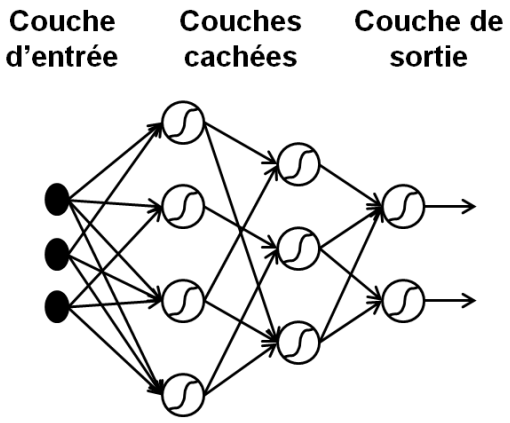

FIGURE 2.13 – Exemple d'un Perceptron multi-couches avec une couche d'entrée, deux couches cachées et une couche de sortie.

$\sigma_i(x)$ désignera la fonction d'activation correspondant à un neurone i.

Si nous considérons un Perceptron multi-couches avec N neurones d'entrée, activés par un vecteur d'entrée x (de taille N), et par $w_{ij}^{0,1}$ le poids correspondant à la connexion entre le neurone i de la couche 0 et le neurone j de la couche 1, la sortie a_j^1 de chacun des neurones de la première couche cachée sera exprimée par :

$$a_j^1 = \sigma_j\left(b_j^1 + \sum_{i=1}^{N} w_{ij}^{0,1} x_i\right) \tag{2.15}$$

où σ_j est la fonction d'activation décrite précédemment, et b_j^1 est un paramètre supplémentaire appelé *biais*, qui peut être considéré comme le poids d'une entrée constante égale à 1, et dont le rôle est de rajouter un degré de liberté supplémentaire en agissant sur la position de la frontière de décision (pour plus de détails, se référer à [Ros57]).

Ce même processus exprimé par l'équation 2.15 peut être répété pour les autres couches (cachées ou celle de sortie) : Chaque sortie d'une couche l joue le rôle d'entrée pour la couche suivante $l+1$. Ainsi, nous pouvons généraliser l'équation 2.15 à toutes les couches suivantes (y compris la couche de sortie) comme suit :

$$a_j^{l+1} = \sigma_j\left(b_j^{l+1} + \sum_{i=1}^{L} w_{ij}^{l,l+1} a_i^l\right) \tag{2.16}$$

où L est le nombre de neurones de la couche l.

Les Perceptrons multi-couches sont généralement utilisés pour des problématiques de classification supervisée. Ceci implique l'existence d'un ensemble de paires d'entrées-sorties (appelé *base d'apprentissage*) liés par une certaine relation, que le réseau va "ap-

prendre" en ajustant ses paramètres. L'apprentissage est effectué avec l'algorithme de *rétro-propagation du gradient* [RHW86].

Cet algorithme contient deux étapes : (i) La "propagation avant" (*forward pass*), durant laquelle des sorties du réseau sont calculées à partir des entrées (comme décrit précédemment), et (ii) Une "propagation arrière" (*backward pass*) durant laquelle les dérivées partielles d'une certaine fonction de coût E (généralement l'erreur quadratique moyenne entre la sortie prédite et la sortie souhaitée) par rapport aux paramètres du réseau sont rétro-propagés. Enfin, les poids du réseau sont mis-à-jour en fonction de cette dérivée partielle :

$$\Delta w_{ij} = -\epsilon \frac{\partial E}{\partial w_{ij}} \tag{2.17}$$

où ϵ est le taux d'apprentissage (*learning rate* en anglais), et où les poids $w_{ij}^{l,l+1}$ sont notés w_{ij} pour simplifier. Nous n'allons pas aborder ici la manière avec laquelle est explicitée la dérivée partielle $\frac{\partial E}{\partial w_{ij}}$ pour chacune des couches. Nous invitons le lecteur intéressé à se référer à l'article de Rumelhart et al. [RHW86].

Une version modifiée de l'équation 2.17 a été introduite par Plaut et al. [PNH86], afin d'accélérer la convergence de la rétro-propagation en prenant en compte les mises à jour précédentes. Ceci est fait en rajoutant un second terme appelé *momentum* :

$$\Delta w_{ij}\,(n) = -\epsilon \frac{\partial E}{\partial w_{ij}}\,(n) + \alpha \cdot \Delta w_{ij}\,(n-1) \tag{2.18}$$

où α est un coefficient de pondération compris entre 0 et 1, et l'indice n désigne les itérations.

Le processus décrit précédemment est répété pendant plusieurs itérations (appelées *epochs* en anglais) jusqu'à la convergence (c'est-à-dire quand l'erreur quadratique moyenne devient quasi-nulle). Le réseau retenu comme étant le plus performant est celui ayant obtenu l'erreur quadratique moyenne la plus faible sur une base indépendante de la base d'apprentissage, appelée *base de validation*. Ce procédé, consistant à arrêter l'apprentissage lorsque l'erreur quadratique moyenne sur la base de validation augmente, s'appelle "arrêt prématuré" (*early stopping*).

2.3.3.2 Réseaux de neurones à convolutions $2D$

Les réseaux de neurones à convolutions (qui seront désignés ci-après par *ConvNets*, pour *Convolutional Neural Networks*) peuvent être vus comme des réseaux de neurones MLPs particulièrement adaptés au traitement des signaux $2D$. Ces réseaux ont été inspirés par les travaux de Hubel et Wiesel sur le cortex visuel chez les mammifères [HW62],

notamment au niveau de leur architecture ainsi que certaines de leur propriétés comme le partage des poids (cf. ci-après). Les premiers *ConvNets* datent des années 1980 avec les travaux sur le *Necognitron* de K. Fukushima [Fuk80], mais c'est dans les années 1990 que ces réseaux seront popularisés avec les travaux de Y. Le Cun et al. sur la reconnaissance de caractères [LBD+90]. Les auteurs ont proposé une série de réseaux *ConvNets*, baptisés *LeNet* (de 1 à 5), qui se basent sur trois idées architecturales clés :

- Des champs récepteurs locaux associés à des convolutions qui permettent de détecter des caractéristiques élémentaires sur l'image, formant ainsi une **carte de caractéristiques**.

- Un principe appelé **partage des poids**, qui consiste à apprendre les mêmes paramètres (ou poids) d'une convolution (et par conséquence extraire les mêmes caractéristiques) pour toutes les positions sur l'image. Ce principe représente l'idée clé des *ConvNets*, puisqu'il permet de réduire considérablement la complexité en diminuant le nombre de paramètres à apprendre, et d'avoir ainsi des architectures multi-couches qui opèrent sur des entrées de grande dimension tout en étant de taille réaliste (ce qui n'était pas réalisable avec les MLPs). De plus, le partage des poids permet d'améliorer les performances en terme de généralisation du réseau, et d'être cohérent avec les études faites sur le cortex visuel [HW62].

- Des opérations de **sous-échantillonnage** qui permettent de réduire la sensibilité aux translations, ainsi que de réduire le coût du traitement.

Les réseaux *LeNet* consistent en une succession de couches qui comportent des cartes de caractéristiques et des cartes de sous-échantillonnage (cf. Figure 2.14). Nous allons nous intéresser à l'architecture du réseau *ConvNet* le plus populaire de l'état de l'art, à savoir le réseau *LeNet-5* [LBBH98] (qui est illustré sur la Figure 2.14), mais sachez toutefois que le nombre de couches, celui des cartes ainsi que leurs dimensions sont des paramètres architecturaux qui varient d'une problématique à une autre.

L'architecture illustrée sur la Figure 2.14 correspond à un réseau de neurones qui comporte 7 couches cachées, en plus d'une couche d'entrée. Ces couches cachées peuvent être classées en deux catégories : (i) Les 4 premières couches sont des cartes $2D$ (de caractéristiques ou de sous-échantillonnage), et (ii) les 3 dernières sont des neurones "classiques" (similaires à celles d'un MLP), où chaque neurone est connecté à tous les neurones de la couche précédente. Sur la Figure 2.14, les cartes de caractéristiques sont représentées en gris et notées C_i, les cartes de sous-échantillonnage sont représentées en bleu et notées S_i, et les neurones sont représentés par des ronds blancs et notés

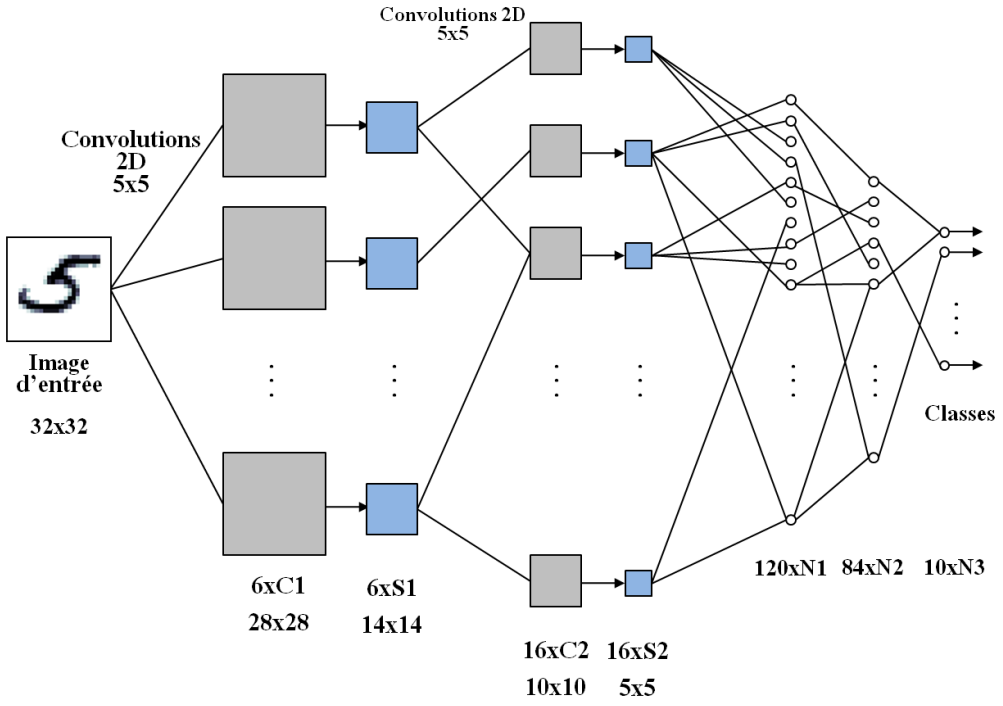

FIGURE 2.14 – Architecture du réseau de neurones à convolutions *LeNet-5* [LBBH98] : Les cartes de caractéristiques sont représentés en gris et notés par des C_i, les cartes de sous-échantillonnage sont représentés en bleu et notés par des S_i, et les neurones sont représentés par des ronds blancs et notés par des N_i. Figure extraite de [LBBH98].

N_i. Les couches C_i opèrent sur leurs entrées des convolutions $2D$ dont les noyaux sont les poids à apprendre, et alimentent les couches suivantes comme pour un MLP classique. Les couches S_i appliquent un moyennage spatial (généralement de facteur 2) sur leur entrées, puis multiplient le résultat par un poids. La succession de ces deux types de couches (la partie C_1, S_1, C_2 et S_2 sur la Figure 2.14) sert à extraire les informations saillantes à partir de l'image d'entrée, et à les encoder dans un vecteur au niveau de N_1. Les 3 dernières couches sont un MLP classique (cf. chapitre précédent) qui sert quant à lui à classer les données encodées. L'objectif est de construire automatiquement, à partir de l'image brute en entrée, une représentation de plus en plus haut-niveau de couche en couche. On parle alors d'apprentissage "profond" (*deep learning* - en anglais).

Le modèle est entraîné par une rétro-propagation avec *momentum* (cf. équation 2.18), qui tient en compte des particularités architecturales des *ConvNets* par rapport aux MLPs (partage de poids, cartes, ...). Nous présenterons plus en détail lors du chapitre 5 les équations de la mise à jour des poids pour chacun de ces composants architecturaux (afin d'introduire leur extension au cas 3D). Nous invitons néanmoins le lecteur intéressé par une description détaillée de l'algorithme d'apprentissage des ConvNets 2D à se

référer à l'article de LeCun et al. [LBBH98], ainsi qu'au manuscrit de thèse de S. Duffner [Duf07].

Les *ConvNets* 2*D* ont été appliqués avec succès à plusieurs problématiques de traitement d'images. A titre d'exemples, nous pouvons citer la reconnaissance de caractères [LBD+90, LBBH98, SG07, EGS11], l'analyse de visages [GD04, DG05, OCM07, NTF09], le suivi de gestes [NP95], la vision robotique [LMB+06], la détection de texte [DG08a], ou encore la détection de logos TV [DG06]. De plus, plusieurs solutions commerciales sont basées sur des *ConvNets* 2*D*, comme le système proposé par Google pour la détection de visages dans les images *StreetView* de l'application *Google Earth* [FCA+09], ou encore un OCR proposé par *Microsoft Research* [CPS06]. Pour un état de l'art complet des applications des *ConvNets* 2*D* en vision par ordinateur, le lecteur intéressé pourra se référer à l'article de Y. Le Cun et al. [LKF10].

Pour toutes ces applications, les *ConvNets* 2*D* ont très souvent démontré leur supériorité par rapport à d'autres caractéristiques, extraites manuellement ou par apprentissage. Ces bons résultats ont donc motivé l'étude de l'extension de ces modèles à d'autres types de signaux. Le paragraphe suivant présente succinctement quelques travaux qui ont proposé d'étendre ces modèles au cas $2D + t$ (signal vidéo).

2.3.3.3 Extension au cas de la vidéo

Nous allons nous intéresser dans ce paragraphe aux quelques travaux qui traitent de réseaux à convolutions $2D + t$, comme une extension du cas 2*D*. A noter toutefois qu'il existe d'autres travaux (que nous n'allons pas présenter dans ce paragraphe) qui appliquent des modèles neuronaux à convolutions 2*D* à des signaux spatio-temporels, généralement en opérant sur les images successives de la vidéo une à une. A titre d'exemples, nous citons les travaux de Ning et al. sur le phénotypage automatique des embryons en développement à partir de vidéos [NDL+05], ou encore ceux de Fan et al. sur le suivi de personnes dans une application de vidéo-surveillance [FXWG10].

L'extension des *ConvNets* 2*D* au cas 3*D* est un domaine encore ouvert. Quelques travaux s'y sont intéressés, surtout pour la problématique de la reconnaissance d'actions humaines. Dans [KLY07], Kim et al. proposent un modèle neuronal hybride qui comporte deux modules : (i) Un *ConvNet* 3*D* pour l'apprentissage des caractéristiques spatio-temporelles, combiné à (ii) un réseau de neurones à logique floue [Sim91] pour la classification. Le *ConvNet* 3*D* proposé par Kim et al., illustré sur la Figure 2.15, a une architecture multi-couches avec des convolutions et des sous-échantillonnages tridimensionnels. Ce modèle n'est pas entraîné avec les données brutes en niveau de gris, mais

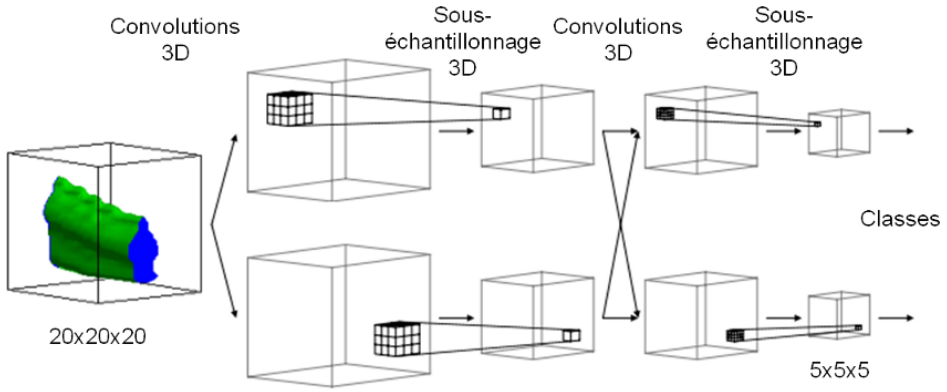

FIGURE 2.15 – Modèle *ConvNet 3D* proposé par Kim et al. dans [KLY07]. Figure extraite de [KLY07].

avec des volumes spatio-temporels (*spatio-temporal Volume* - STV en anglais [YS05]) obtenus en projetant les contours extérieurs du personnage en mouvement dans le plan de chaque image. L'architecture proposée prend en entrée des volumes $3D$ de taille fixe $20 \times 20 \times 20$ (c'est à dire que le label attribué à une vidéo est calculé en se basant sur 20 instants), et produit en sortie des cartes de caractéristiques spatio-temporelles de taille $5 \times 5 \times 5$ chacune, qui sont concaténées ligne par ligne pour former un vecteur de description utilisé pour la classification.

Plus récemment, Ji et al. [JXYY10]) ont proposé un modèle *ConvNet* $3D$ qui n'est pas hybride, et dont l'architecture ressemble beaucoup à celle du *ConvNet* $2D$ *LeNet-5* de Y. Le Cun, mais avec des convolutions et des sous-échantillonnages $3D$. Tout comme le modèle de Kim et al. [KLY07], cette approche n'opère pas sur les données brutes, mais sur une combinaison de plusieurs signaux (gradients, flot optique, ...) calculés manuellement à partir de la vidéo. Le deuxième point de ressemblance avec le modèle de Kim et al. [KLY07] est le fait que la classification se base sur quelques images uniquement de la vidéo (typiquement, 9 images dans [JXYY10]). L'attribution du label à la séquence entière s'effectue par un système de vote. Cette approche a été évaluée sur deux bases standards d'actions humaines et les résultats obtenus sont satisfaisants (cf. chapitre 7).

Enfin, il est à noter que certains travaux s'inspirent dans leur esprit et leur architecture des modèles *ConvNets* sans pour autant en faire partie. A tire d'exemple, nous citons les travaux de Jhuang et al. [JSWP07] qui proposent une architecture multi-couches avec une succession de cartes de caractéristiques et de sous-échantillonnages similaires à celles des *ConvNets*, mais avec quelques différences : La première c'est que les convolutions sont remplacées par des filtres de Gabor $3D$, et les cartes de caractéristiques sont donc calculées manuellement et non par apprentissage. De ce fait, les caractéristiques

de bas et moyen niveau extraites par ce modèle sont conçues manuellement, et l'apprentissage n'intervient que dans les dernières couches (MLP pour la classification). Le sous-échantillonnage quant à lui s'effectue non pas en moyennant les entrées mais en gardant la valeur maximale sur un voisinage donné (un procédé appelé *Max Pooling*).

2.3.4 Autres modèles

D'autres approches pour l'apprentissage automatique de caractéristiques spatio-temporelles ont été présentées dans la littérature. Dans [CO09], Cadieu et Olshausen ont proposé un modèle probabiliste à deux couches qui est capable de modéliser des mouvements complexes. La première couche sépare le signal spatio-temporel en une partie statique (l'amplitude), et une partie dynamique (la phase, qui représente le mouvement). La seconde couche est ensuite entraînée à modéliser les dépendances temporelles entre les variables de phases, et ainsi à caractériser le mouvement.

Dean et al. [DWC09] ont proposé d'appliquer de manière récursive l'algorithme de codage parcimonieux introduit par Lee et al. [LBRN07] pour l'étendre au cas spatio-temporel. Le modèle opère sur des patchs $3D$ extraits autour de points d'intérêts. Bien que la détection de ces points soit manuelle, les vecteurs de description sont quand à eux générés par apprentissage.

Enfin, une approche plus récente, présentée par Le et al. [LZYN11], a étudié l'extension de l'analyse en sous-espaces indépendants pour l'apprentissage non supervisé de caractéristiques à partir de séquences vidéo. Le modèle présenté s'inspire aussi de certaines caractéristiques architecturales des *ConvNets* (comme l'architecture multi-couches, ainsi que les convolutions) et obtient de très bons résultats sur quatre bases standards de la reconnaissance d'actions humaines (cf. chapitre 7).

2.4 Conclusion

Nous nous sommes intéressés dans ce chapitre aux caractéristiques visuelles utilisées dans l'état de l'art pour la classification de séquences vidéo. Nous avons dans un premier temps présenté une première catégorie de caractéristiques dont le point commun est le fait qu'elles soient conçues manuellement (c'est à dire reposant sur des connaissances a priori du signal d'entrée) et donc dépendantes de la problématique étudiée. Nous avons donc dressé un bref état de l'art des caractéristiques les plus utilisées dans trois domaines différents : La reconnaissance d'actions humaines, la reconnaissance d'expressions faciales et la classification de vidéos de sport. Ce qui ressort de cet état de l'art est que les caractéristiques manuelles les plus utilisées diffèrent d'un domaine d'application

à un autre (typiquement la reconnaissance d'actions humaines et celle des expressions faciales), et même pour un même domaine (par exemple pour le cas des vidéos de sport, où les caractéristiques dépendent du sport étudié, du niveau sémantiques des classes, ...). Ces caractéristiques extraites manuellement semblent donc très peu génériques.

Nous nous sommes donc intéressés dans un second temps à une deuxième catégorie d'approches qui génèrent des caractéristiques par apprentissage à partir d'exemples, et qui sont donc plus génériques. Nous avons mis l'accent sur les réseaux de neurones à convolutions (vues leurs excellentes performances pour le cas des signaux $2D$) ainsi que leur extension au cas $3D$. Nous avons ainsi constaté que ces modèles ont été très peu appliqués à la classification vidéo, et que les quelques travaux qui s'y sont intéressés soit n'exploitent pas l'information temporelle [NDL+05], soit n'opèrent pas sur des données brutes mais pré-traitées avec des opérations manuelles, souvent complexes [KLY07, JXYY10]. De plus, pour ces travaux, la classification est souvent faite sur des sous-séquences de courte durée, et avec un système de vote pour attribuer un label à la séquence complète. Or, il semble intéressant de prendre en compte les dépendances temporelles qui peuvent exister dans une séquence vidéo lors de la classification. Nous allons proposer dans les chapitres 5 et 6 deux modèles basés sur les réseaux de neurones à convolutions (entraînés respectivement de manière supervisée et non supervisée) dont le but est de remédier à ces limitations.

Au delà des caractéristiques utilisées, l'autre point clé de toute méthode de classification vidéo est le choix du classifieur de séquences. Nous allons nous intéresser dans le chapitre suivant à différents modèles de classification de séquences parmi les plus populaires de l'état de l'art.

Chapitre 3

Modèles de classification de séquences

Sommaire

3.1 Introduction

La classification automatique de séquences est un sous-domaine très important de l'apprentissage automatique, notamment de part les enjeux applicatifs qui y sont liés. En effet, de nombreuses applications en traitement audio et vidéo, en indexation multimédia, en reconnaissance de formes, et en traitement des langues, mais aussi dans d'autres domaines tels que la biologie ou la finance font intervenir des données sous forme de séquences, c'est à dire une suite ordonnée d'informations qui peuvent être des nombres, des symboles, ou encore des vecteurs. Cette dernière catégorie est celle qui nous intéresse dans le cadre de cette thèse. En effet, nous avons vu lors du chapitre précédent que pour le cas de la vidéo, la classification est généralement précédée d'une étape d'extraction de caractéristiques (manuellement ou par apprentissage) de manière à représenter

la vidéo par une séquence de vecteurs. Par conséquent, et sauf mention du contraire, les modèles de classification de séquences étudiés dans ce chapitre auront comme entrées des séquences de longueurs variables de vecteurs à valeurs réelles, et auront pour objectif d'y associer des labels (un par séquence).

Les modèles de classification de séquences que nous allons aborder dans ce chapitre sont entraînés de manière supervisée, c'est à dire qu'ils se basent sur des exemples d'apprentissage labellisés manuellement par un expert (ce processus s'appelle l'annotation) pour modéliser une certaine relation entre les séquences de vecteurs et les labels/étiquettes. Le but est par la suite de pouvoir estimer les labels d'autres séquences qui ne sont pas dans la base d'apprentissage (on parle alors de pouvoir de "généralisation" du modèle de classification). À noter toutefois qu'il existe dans la littérature des modèles de classification de séquences qui sont entraînés de manière non supervisée (c'est à dire en utilisant des données non-étiquetées) ou encore semi-supervisée (c'est à dire en utilisant à la fois des données étiquetées et non-étiquetées), mais que nous n'allons pas aborder dans ce chapitre.

Ce chapitre présente, de manière succincte, les fondements théoriques de quelques modèles de classification de séquences parmi les plus utilisés dans l'état de l'art. L'objectif est de fournir un point de départ pour une étude comparative entre ces différents modèles, qui sera présentée ultérieurement lors du chapitre 4.

Nous allons nous intéresser tout d'abord dans la section 3.2 aux modèles graphiques probabilistes, et plus particulièrement aux champs aléatoires conditionnels (*Conditional Random Fields* - CRF en anglais) [LMP01] et aux champs aléatoires conditionnels cachés (*Hidden Conditional Random Fields* - HCRF en anglais) [QWM+07]. La section 3.3 présentera ensuite les machines à vecteurs de support [Vap98] ainsi que leur adaptation au cas de la classification de séquences. Enfin, nous aborderons dans la section 3.4 les réseaux de neurones récurrents, et plus particulièrement les réseaux dits à longue mémoire à court terme [HS97].

3.2 Modèles graphiques probabilistes pour la classification de séquences

De nombreuses problématiques en traitement automatique des données, et plus particulièrement en classification, peuvent être modélisées de manière probabiliste : C'est à dire en définissant un certain nombre de variables aléatoires, et en cherchant les meilleures réalisations de ces variables, selon un certain critère. Ceci est fait généralement via la maximisation (ou la minimisation, selon les cas) d'une fonction objectif globale, qui as-

socie une probabilité ou une énergie à chaque réalisation de l'ensemble des variables.

Or en pratique, l'estimation de ces probabilités (dites conjointes) reste très complexe (voir non faisable) en l'absence de toutes contraintes. Ceci est dû notamment au fait que, sans contraintes, les variables aléatoires concernées peuvent toutes être interdépendantes, aussi bien directement que conditionnellement. Une solution à ce problème consiste alors à imposer des contraintes sur ces dépendances conditionnelles, en autorisant que certaines variables soient indépendantes conditionnellement (et non directement) d'autres variables, ce qui permet de factoriser la probabilité conjointe (mentionnée précédemment) en un produit de facteurs, faisant intervenir chacun un faible nombre de variables.

Les modèles graphiques probabilistes sont une famille de modèles qui présentent une solution pratique pour gérer ces indépendances conditionnelles. Ils permettent en effet de les modéliser simplement sous forme d'un graphe, dans lequel les sommets représentent les variables. Plusieurs modèles graphiques probabilistes ont ainsi été proposé, dont quelques uns adaptés au traitement des données séquentielles. L'exemple le plus connu dans cette catégorie sont les modèles de Markov cachés (HMM pour *Hidden Markov Models*).

Les fondements théoriques des modèles HMMs remontent au milieu des années 1960 / début des années 1970 et aux travaux de Baum et al. [BP66, BE67, Bau72]. Mais ces modèles n'ont été popularisés que dans les années 1980 avec les travaux de Bahl et al. [BJM83], de Poritz et al. [Por88] et surtout ceux de Rabiner et al. [Rab89] sur la reconnaissance de la parole. Sans rentrer dans les détails, les HMMs peuvent être vus comme un ensemble d'états qui transitent entre-eux et qui ne sont pas visibles directement (d'où l'emploi du terme "caché") mais à travers des observations. L'une des idées clés est de supposer que chaque variable à un instant t est indépendante conditionnellement de toutes les variables aux instants inférieurs ou égaux à $(t-2)$, sachant son état caché à l'instant $(t-1)$. Cette contrainte s'appelle la Markovianité (qui est à l'origine du nom des HMMs). Concrètement, si nous notons par X l'ensemble des variables aléatoires relatives aux observations, et par Y celles relatives aux étiquettes des états cachés, et si nous notons par x et y les réalisations de ces variables, la Markovianité stipule que :

$$P\left(y_t \middle| \left(y_1, \ldots, y_{t-1}\right)\right) = P\left(y_t | y_{t-1}\right) \tag{3.1}$$

où la notation $P(x)$ est utilisée ici pour désigner $P(X = x)$ pour simplifier, et où l'indice temporel t caractérise l'état à cet instant d'une variable ou d'une réalisation données. Dans la suite, nous utilisons également une notation connue désignant les

variables aléatoires par des symboles majuscules et leurs réalisations par des symboles minuscules.

Cette contrainte de Markovianité permet de maîtriser la complexité et de mettre au point des algorithmes d'inférence efficaces. Cette propriété représente d'ailleurs le principal point fort des modèles HMMs, qui ont ainsi été largement appliqués à des problématiques de classification de séquences, surtout dans le cas des signaux audio (un état de l'art complet peut être consulté dans [HAH01]), mais aussi pour la vidéo [BW98, PJZ01, ZGPBM05, CYC08, XG08, NDA09]. Ces modèles souffrent néanmoins de nombreuses limitations, que nous allons détailler dans la sous-section suivante.

3.2.1 Champs aléatoires conditionnels

Les modèles CRFs ont été introduit par Lafferty et al. [LMP01], afin de remédier à certaines limitations des HMMs. En effet, la première de ces limitations est qu'ils font partie d'une catégorie de modèles dits "génératifs", c'est à dire qui capturent le processus par lequel les observations sont générées. Ils sont définis par opposition aux modèles dits "discriminatifs", qui modélisent directement l'inférence des variables cachées à partir des observations, sans disposer d'informations sur le processus génération de ces dernières. Cela se traduit mathématiquement par le fait que les modèles génératifs fournissent un modèle probabiliste complet de toutes les variables, alors que les modèles discriminatifs ne modélisent que la probabilité de la (ou des) variable(s) cible(s) conditionnellement aux observations. Or même si les modèles génératifs sont en général plus adaptés quand il s'agit de caractériser des relations de dépendances complexes, ils sont moins performants en terme de classification, si nous ne disposons pas d'information sur le processus de génération des observations [LMP01]. Il faut cependant noter qu'il est impossible de parler d'une supériorité générale d'une famille de modèles par rapport à l'autre, la différence de performance dépend en pratique de l'application [Jor02]. A noter également que certains travaux ont proposé des modèles HMMs discriminants, mais dont les performances en classification restent inférieures aux autres modèles de cette catégorie [DA09].

Par ailleurs, au delà de la contrainte de Markovianité mentionnée précédemment (cf. équation 3.1), les modèles HMMs imposent une autre contrainte forte sur les observations, qui sont conditionnellement indépendantes sachant leurs variables cachées. En effet, si nous reprenons les notations de l'équation 3.1, cette propriété d'indépendance conditionnelle des observations se traduit par :

$$P(x_t|x, y) = P(x_t|y_t) \tag{3.2}$$

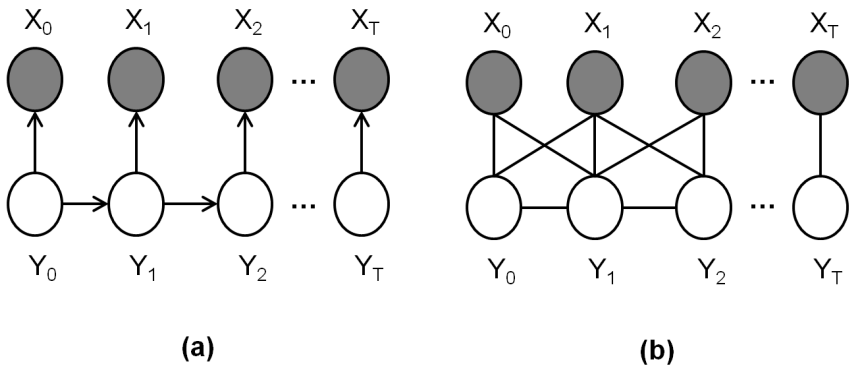

FIGURE 3.1 – Graphes de dépendances correspondant au : (a) - Modèle HMM [Rab89] (graphe modélisant la probabilité conjointe) (b) - Modèle CRF [LMP01] (graphe modélisant la probabilité conditionnelle).

Les observations dépendent donc juste de "leurs" états cachés, et non des autres états cachés ou observations. Cette hypothèse est nécessaire pour rendre l'apprentissage des paramètres des HMMs faisable, mais est très contraignante et très souvent non valide en pratique.

Les modèles CRFs apportent des solutions à ces deux limitations. En effet : (i) Ce sont des modèles discriminatifs, et leurs paramètres ne sont donc pas utilisés pour modéliser les observations, les rendant ainsi particulièrement adaptés à la classification, et (ii) Ils permettent de relâcher les indépendances conditionnelles des observations, ce qui offre un réel avantage par rapport aux HMMs dans le sens où une observation peut être liée à plusieurs variables cachées, et vice versa. Nous illustrons ce principe sur la Figure 3.1, qui représente les graphes de dépendances des deux modèles. Nous pouvons observer que, contrairement au graphe de dépendances des HMMs (cf. Figure 3.1-(a)), celui des modèles CRFs (illustré sur la Figure 3.1-(b)) présente des arêtes supplémentaires reliant les noeuds. Une autre différence réside dans le fait que ce dernier graphe est non orienté (comme pour tous les champs de Markov aléatoires), contrairement à celui des HMMs (qui sont des cas particuliers de réseaux Bayésiens).

L'inférence des variables cachées y à partir des variables observées x peut se faire en maximisant la probabilité a posteriori $P(y|x)$, qui s'exprime comme suit :

$$P(y|x) = \frac{1}{Z(x)} exp\left(\sum_{c,k} \lambda_{c,k} f_k(y_c, x)\right) \tag{3.3}$$

où $Z(x)$ est un terme de normalisation appelé fonction de partition, analogue à celui présenté dans le chapitre 2 pour les modèles RBMs, f_k sont des fonctions appelées

"fonctions de caractéristiques", et $\lambda_{c,k}$ sont les paramètres associés à une clique c et à la fonction f_k. Une clique est un sous-graphe complet d'un graphe, et dans ce contexte les cliques sont liées aux différents facteurs de la probabilité, c.à.d. aux différents termes de la fonction d'énergie. L'ensemble des $\lambda_{c,k}$ pour toutes les cliques et toutes les fonctions de caractéristiques forment le vecteur θ des paramètres du modèle. Le jeu de paramètres optimal θ^* est celui qui maximise la probabilité exprimée dans l'équation 3.3 pour les N données d'apprentissage. L'estimation se fait par maximum de vraisemblance, dans le domaine logarithmique :

$$\theta^* = \arg\max_{\theta} ln\left(P\left(y|x,\theta\right)\right) \tag{3.4}$$

En considérant le fait que les N échantillons d'apprentissage sont indépendants, le terme d'énergie de l'équation 3.4 peut être ré-écrit de manière à rendre la vraisemblance convexe sur les paramètres θ, ce qui permet de trouver une solution globale en appliquant des algorithmes d'optimisation classiques tels que la descente de gradient ou les méthodes Quasi-Newton.

Dans la sous-section suivante, nous allons nous intéresser à une extension des modèles CRFs, qui est adaptée au traitement des séquences complètes.

3.2.2 Champs aléatoires conditionnels cachés

Les HMMs et les CRFs partagent une faiblesse commune. Ils modélisent la distribution d'une classe de séquences, permettant d'inférer la séquence d'états cachés à partir d'une séquence d'observations. Dans un contexte de classification de séquences, où l'étiquette désirée n'est pas associée à un état mais à une séquence complète, il s'avère nécessaire d'apprendre un modèle HMM/CRF pour chaque classe de séquences et de choisir entre les modèles lors de la classification.

Les modèles HCRFs ont été introduits en 2007 par Quattoni et al. dans [QWM+07] pour palier à ce problème. Ils peuvent être vus comme une extension des modèles CRFs, dont la structure n'est pas restreinte à une chaîne, et dans laquelle une variable cachée supplémentaire L, liée à toutes les autres variables cachées, modélise la classe de la séquence entière. Ceci se traduit sur le graphe de dépendances par l'ajout d'un noeud supplémentaire connecté à tous les noeuds cachés (cf. Figure 3.2).

Ces modèles ont été initialement présentés pour la reconnaissance d'objets dans les images [QWM+07]. Depuis, ils ont été largement utilisés dans différentes applications de classification de séquences, aussi bien en traitement audio [GMAP05, SJ09, YDA09] que vidéo (par exemple la reconnaissance d'actions dans [WM08, LJ08, ZG10, KS12], ou encore la reconnaissance de gestes dans [WQM+06]).

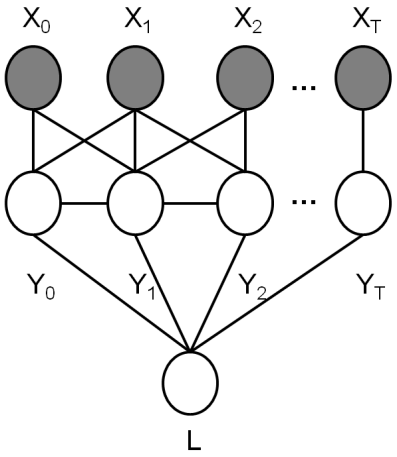

FIGURE 3.2 – Graphe de dépendances d'un modèle HCRF [QWM+07].

La formulation théorique des HCRFs est une extension de celle des CRFs, en prenant en compte la variable supplémentaire L (et sa réalisation notée l). En particulier, la probabilité a posteriori (cf. équation 3.3) s'exprime dans ce cas par :

$$P(y, l|x) = \frac{1}{Z(x)} exp\left(\sum_{c,k} \lambda_{c,k} f_k(y_c, l_c, x)\right) \tag{3.5}$$

où les notations sont les mêmes que celles de l'équation 3.3.

Comme pour le cas des CRFs, les paramètres optimaux θ^* du modèle sont obtenus par une maximisation dans le domaine logarithmique. En revanche, la fonction d'énergie ne peut pas être ramenée à une forme convexe, et les solutions obtenues par les méthodes d'optimisation classiques sont locales. Les auteurs montrent néanmoins que les méthodes du gradient stochastique et du gradient conjugué aboutissent à une solution globale quand ils sont initialisés correctement [QWM+07].

Les modèles HCRFs ont démontré leur supériorité aux modèles HMMs et/ou CRFs pour de nombreuses applications [GMAP05, WQM+06, SJ09, YDA09, ZG10]. Ils offrent aussi un avantage pratique dans le sens où ils permettent de modéliser toutes les classes des séquences par le même modèle, dans le cas de la classification de séquences entières (ce qui n'est pas le cas pour les HMMs et les CRFs, pour lesquels il faut entraîner un modèle par classe de séquences). Nous allons évaluer dans le chapitre 4 le pouvoir discriminant de ce modèle dans le cadre d'une étude comparative par rapport à d'autres modèles de classification de séquences (non probabilistes), qui seront présentés dans ce qui suit.

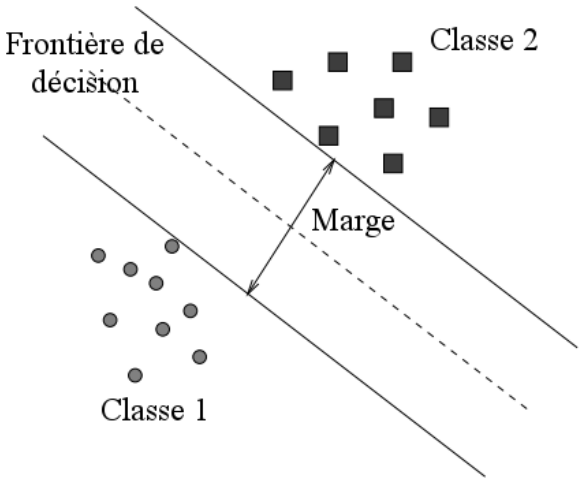

FIGURE 3.3 – Machines à vecteurs de support : Illustration de la classification binaire par maximisation de la marge entre deux classes.

3.3 Machines à vecteurs de support adaptées à la classification de séquences

3.3.1 Machines à vecteurs de support

Les machines à vecteurs de support (*Support Vector Machines* - SVM en anglais) sont une famille de classifieurs supervisés qui ont été introduits dans les années 1990 par Vapnik et al. [CV95, Vap98] et qui depuis sont devenus parmi les modèles les plus utilisés pour la classification et la régression.

Les SVMs opèrent sur des données plongées dans un espace vectoriel, et le but est de les catégoriser. L'idée clé est d'augmenter la dimension de la représentation de ces données, et ensuite de calculer la surface de séparation entre les classes dans cet espace de représentation (appelé "hyperplan"), partant du principe que les données sont linéairement séparables dans cet espace de grande dimension (principe appelé "principe de Cover" [Cov65]).

Dans leur version standard, les SVMs sont entrainés à séparer deux classes (classification binaire) par une frontière de décision (appelée aussi "séparatrice") définie comme étant celle qui maximise la plus petite distance (ou "marge") entre la frontière de décision et chacune des données d'apprentissage (cf. Figure 3.3).

Plusieurs approches ont été proposées pour étendre cette classification binaire au cas multi-classes. La méthodologie dominante (et la plus efficace) consiste à décomposer la

classification multi-classes en plusieurs classifications binaires [DK05]. Cela peut se faire de deux manières différentes :

- **La méthode "un contre tous"** : Dans laquelle chaque classifieur binaire est entraîné à distinguer entre l'une des classes et le reste des classes. Pendant la phase de test, la classe choisie est celle correspondant à la sortie maximale.

- **La méthode "un contre un"** : Dans laquelle les classifieurs sont entraînés à distinguer entre des paires de classes. La décision finale pour la phase de test étant obtenue par un système de vote.

Ainsi, tous les problèmes de classification SVM peuvent être reformulés comme un ensemble de classifications binaires. La formulation théorique de cette classification binaire est détaillée dans l'ouvrage de V. Vapnik [Vap98] que nous invitons le lecteur intéressé à consulter. Nous allons nous intéresser ici aux notions fondamentales de cette formulation.

Si nous disposons d'un ensemble de N vecteurs d'apprentissage $(x_i)_{i\in\{1,\cdots,N\}}$ de même dimension, et si nous notons par ϕ la fonction qui permet de les projeter dans l'espace de grande dimension décrit précédemment, le modèle SVM permet de calculer une frontière de décision $y(x_i)$ exprimée par :

$$y(x_i) = w^T \cdot \phi(x_i) + b \tag{3.6}$$

où w et b sont respectivement les paramètres et les biais du modèle.

La maximisation de la marge correspondant à cette frontière de décision lors de l'apprentissage s'effectue alors par une formulation duale, et fait intervenir un produit scalaire $\langle \phi(x_i), \phi(x_j) \rangle$ associé à l'espace de projection.

Or, le calcul explicite de ce produit scalaire est généralement coûteux en temps de calcul et en ressources. Des fonctions particulières dites "fonctions noyaux" ont donc été introduites afin de remédier à ce problème (pour cette raison, les SVMs font partie d'une catégorie d'approches en apprentissage statistique dite "méthodes à noyaux"). Ces fonctions permettent d'approximer le produit scalaire dans l'espace de projection sans passer par le calcul explicite de $\langle \phi(x_i), \phi(x_j) \rangle$.

Plusieurs fonctions noyaux ont ainsi été présentées dans la littérature, les plus connues étant les noyaux linéaires, les noyaux radiaux, les noyaux polynomiaux et les noyaux Khi-carré χ^2. Une étude comparative relativement récente entre ces différents noyaux a été effectuée dans [ZMLS07].

Le point commun entre ces différentes fonctions noyaux est le fait que tous les

vecteurs de leurs espaces d'entrée doivent avoir la même dimension. La classification SVM n'est donc pas directement applicable à la classification de séquences. Plusieurs approches ont été proposées dans la littérature afin d'adapter le modèle SVM à des applications liées à la classification de séquences. Le paragraphe suivant présente les différentes stratégies d'adaptation.

3.3.2 Stratégies d'adaptation des SVMs à la classification de séquences

Les différentes approches qui visent à adapter la classification SVM au cas des séquences peuvent être classées en trois catégories :

Les représentations de taille fixe : Cette première catégorie d'approches consiste à représenter les séquences de taille variable par des vecteurs de taille fixe, construisant ainsi une représentation plus adaptée à la nature du classifieur SVM.

Ceci est fait soit en opérant sur des sous-séquences (de taille fixe) choisies pour être représentatives du contenu de la séquence entière, soit en modifiant l'espace de représentation des données. Dans ce dernier cas, le choix de ce "nouvel" espace de représentation dépend de l'application, et est généralement lié à un type de caractéristiques.

Par exemple, en classification de séquences audio, les coefficients cepstraux MFCC (*Mel-Frequency Cepstral Coefficients*) sont combinés à différents traitements (un critère de distance euclidienne entre trames voisines dans [BK00] ou encore la division de chaque trame en un nombre fixe de sous-trames d'égale durée dans [LSTI02]) afin de calculer un vecteur de description de taille fixe qui représente le contenu le plus pertinent de la séquence.

Dans le cas des séquences vidéo, plusieurs approches ont été présentées, et qui peuvent être regroupées en deux catégories : (i) Celles qui se basent sur le découpage en sous-séquences comme dans le cas audio [SO05], et (ii) celles qui se basent sur les représentations par histogrammes (les sacs de mots [DRCB05], les histogrammes de caractéristiques locales [SLC04],...).

Les modèles hybrides : L'idée dans cette catégorie d'approches est de combiner la classification SVM avec d'autres classifieurs adaptés aux séquences. Plusieurs exemples de ces classifieurs dits "hybrides" existent dans la littérature, parmi lesquels nous citons la combinaison avec les modèles de Markov cachés (par exemple dans le cas audio [GHP00] ou vidéo [BKJ+05]), ou encore avec les champs aléatoires conditionnels cachés dans [LLK10]. L'idée générale est d'utiliser les classi-

fieurs adaptés aux séquences (HMM, HCRF ou autre) en amont afin, d'une part, de générer des séquences de taille fixe, et d'autre part, d'incorporer au critère de décision du SVM une certaine connaissance des entrées.

Les noyaux de séquences : Les deux familles d'approches décrites précédemment sont très utilisées, mais présentent néanmoins plusieurs limitations : Pour la première catégorie, la représentation des séquences par des vecteurs de taille fixe ne tient pas compte de l'aspect temporel des données au cours de la classification. En ce qui concerne les approches de la seconde catégorie, elles tiennent en général compte de cet aspect temporel des données, mais nécessitent la mise en place de modèles et de stratégies d'apprentissage complexes et coûteuses.

L'idée a été donc de proposer une troisième catégorie d'approches qui se base sur l'adaptation des fonctions noyaux aux séquences de vecteurs. Ceci est en général fait en remplaçant la distance euclidienne par une distance d'édition dans la fonction de noyau, permettant ainsi de comparer des séquences de tailles différentes. Une distance d'édition entre deux séquences peut être définie par le nombre d'opérations (insertions, suppressions, substitutions, ...) requises afin de transformer l'une d'elle en l'autre. La plupart des travaux de la littérature utilisent la distance de Needleman-Wunsch [NW70] puisque, d'une part, elle effectue un alignement global entre les deux séquences, et d'autre part, représente aussi une mesure de similarité. Il est à noter que ces noyaux de séquences ne représentent pas des fonctions noyaux valides (puisqu'elles ne vérifient pas l'une des conditions nécessaires du théorème de Mercer [Mer09] qui caractérise ces fonctions), mais elles ont néanmoins été validées empiriquement dans le cadre de plusieurs applications parmi lesquelles nous pouvons citer la reconnaissance de texte [LSST+02, BHB02], la classification de séquences de protéines [LEC+04], la reconnaissance structurelle des formes [NB06], ou encore la classification de séquences vidéo [MHV03, BBBS09, BBDBS10].

3.4 Réseaux de neurones récurrents à longue mémoire à court-terme

Nous allons nous intéresser dans cette section à une variante des réseaux de neurones récurrents particulièrement adaptées à la classification de séquences. Nous allons dans un premier temps rappeler quelques notions autour des réseaux de neurones récurrents classiques. Puis, nous allons présenter les réseaux de neurones récurrents dits à longue

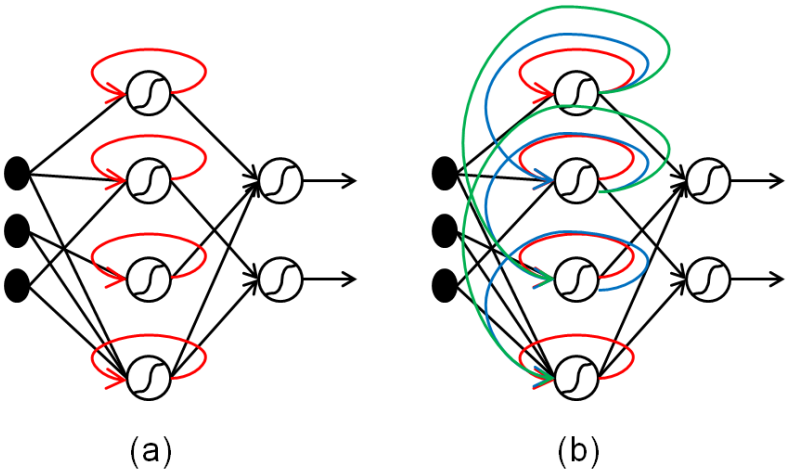

FIGURE 3.4 – Réseau de neurones récurrent avec une couche cachée : (a)- Réseau auto-récurrent basique (b)- Réseau récurrent totalement connecté.

mémoire à court-terme.

3.4.1 Réseaux de neurones récurrents

Nous avons présenté lors du paragraphe 2.3.3.1 les Perceptrons multi-couches, qui représentent une catégorie de modèles neuronaux dits "acycliques" (en anglais *feedforward neural network*), c'est à dire dans lesquels les flux d'information ne se propagent que dans un sens : De l'entrée du réseau vers sa sortie. Ces réseaux n'ont donc que des connexions directes, qui ne forment pas de boucles. Si cette contrainte est relâchée, nous obtenons les réseaux de neurones récurrents (RNN pour *Recurrent Neural Networks*). Plusieurs modèles neuronaux récurrents ont ainsi été présentés dans la littérature, et l'état de l'art sur les RNNs est très vaste (nous invitons le lecteur intéressé par un état de l'art complet à se référer à l'ouvrage de Medsker et Jain [MJ10]).

La Figure 3.4-(a) illustre le réseau de neurones récurrent le plus basique, dont la couche cachée est dite *auto-récurrente* (c'est à dire que chaque neurone de la couche cachée possède une seule connexion récurrente reliant sa sortie à son entrée). La Figure 3.4-(b) présente quant à elle un exemple de réseau de neurones récurrent plus complexe (dit totalement connecté), où tous les neurones de la couche cachée sont connectés entre-eux.

Si nous considérons le réseau récurrent basique illustré sur la Figure 3.4-(a), les équations caractérisant la stimulation, à chaque instant, sont définies de la même manière que pour le Perceptron multi-couches, en tenant compte en plus des connexions récurrentes de la couche cachée. Le réseau prend en entrée, non plus des vecteurs x de taille N, mais

des séquences de vecteurs $x(t)$ de taille N chacun. En reprenant les même notations que pour l'équation 2.16 présentée dans le chapitre 2, et en désignant par t l'indice temporel, nous obtenons :

$$a_j^{l+1}(t) = \sigma_j\left(b_j^{l+1} + \sum_{i=1}^{L_l} w_{ij}^{l,l+1} a_i^l(t) + w_{jj}^{l+1,l+1} a_j^{l+1}(t-1)\right) \tag{3.7}$$

Pour le cas des réseaux totalement connectés (cf. Figure 3.4-(b)), cette équation s'écrit :

$$a_j^{l+1}(t) = \sigma_j\left(b_j^{l+1} + \sum_{i=1}^{L_l} w_{ij}^{l,l+1} a_i^l(t) + \sum_{i=1}^{L_{l+1}} w_{ij}^{l+1,l+1} a_i^{l+1}(t-1)\right) \tag{3.8}$$

Les équations 3.7 et 3.8 sont appliquées de manière récursive afin de calculer la séquence d'activations correspondant à chaque neurone. L'apprentissage est effectué en utilisant une version de l'algorithme de rétro-propagation du gradient adaptée aux entrées/sorties temporelles. Deux principaux algorithmes ont été proposés dans la littérature : (i) La rétro-propagation dans le temps (BPTT pour *backpropagation through time*) [WZ95] et (ii) l'apprentissage récurrent en temps réel (RTRL pour *real time recuurrent learning*) [RF87]. L'algorithme BPTT est le plus utilisé dans l'état de l'art, et peut être vu comme une extension de la rétro-propagation classique (utilisée pour les réseaux de neurones acycliques), mais en tenant compte des connexions récurrentes (qui sont alors considérées comme des entrées supplémentaires). Une présentation détaillée de ces deux algorithmes peut être consultée dans l'ouvrage de Mandic et Chambers [MC01].

Plusieurs architectures récurrentes ont été définies dans l'état de l'art [Jor86, Elm90, LWH90, Jae01], dont le principe commun est d'apprendre une correspondance entre des séquences de vecteurs d'entrée, et des séquences de vecteurs désirés, en utilisant les connections récurrentes qui permettent de se "rappeler" d'un certain nombre d'états passés. Ainsi, à un instant t pour une séquence donnée, les RNNs font intervenir les instants passés lors du calcul de l'état présent (cf. équations 3.7 et 3.8) : On parle alors de *contexte*. Les capacités d'apprentissage de la tache souhaitée (typiquement la classification) seront donc directement dépendantes de la quantité d'information de contexte disponible pour le réseau.

La Figure 3.5-(a) présente un réseau récurrent unidirectionnel classique comme ceux décrits précédemment. Ce réseau peut être vu comme une succession de MLPs (un réseau pour chaque instant), avec des entrées classiques, mais aussi les sorties de la couche cachée du MLP correspondant à l'instant précédent. Ce principe est illustré sur la Figure 3.5-(b) et est appelé *vue éclatée* (qui sert de base à la méthode d'apprentissage BPTT évoquée précédemment).

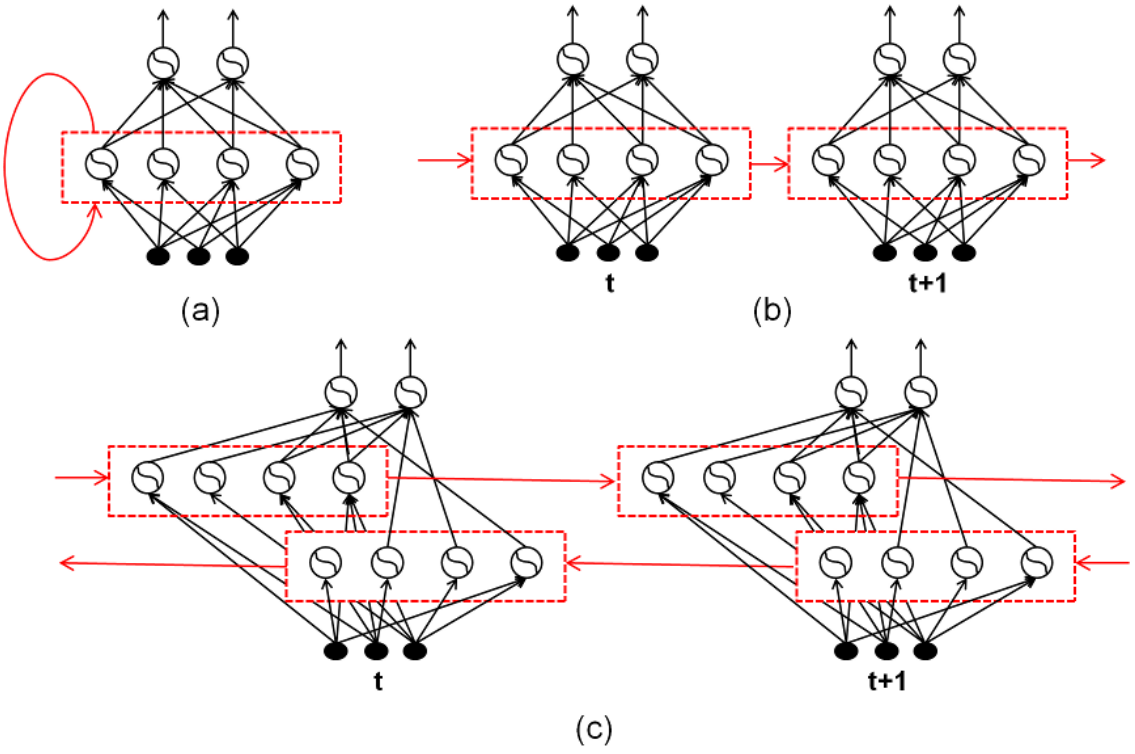

FIGURE 3.5 – (a)- Réseau récurrent unidirectionnel (b)- Réseau récurrent unidirectionnel en vue éclatée (c) - Réseau récurrent bidirectionnel en vue éclatée.

L'un des moyens proposés dans la littérature pour augmenter la quantité d'information de contexte est de permettre à un instant t l'accès aussi bien au futur qu'au passé. En pratique, ceci est fait en utilisant deux couches cachées : Une pour chaque direction. On parle alors de réseau récurrent *bidirectionnel* [SP97] (par opposition au réseau *unidirectionnel* présenté précédemment). La Figure 3.5-(c) représente la *vue éclatée* d'un réseau récurrent bidirectionnel : Les réseaux MLPs successifs comptent deux couches cachées (une pour chaque direction) qui sont connectées aux mêmes couches d'entrée et de sortie. Ces deux couches permettent, en théorie, au réseau à chaque instant d'avoir accès au contexte passé et futur d'une séquence donnée (tout se passe en fait comme si la séquence était présentée au réseau dans deux directions opposées).

Au niveau de l'apprentissage, les réseaux récurrents bidirectionnels sont entrainés de la même manière que les réseaux unidirectionnels, en tenant compte des sens de propagation différents pour les deux couches cachées. Plus de détails concernant les réseaux récurrents bidirectionnels peuvent être consultés dans [SP97].

3.4.2 Réseaux récurrents à longue mémoire à court terme

Comme évoqué dans le paragraphe précédent, l'intérêt majeur des RNNs est leur capacité à utiliser l'information contextuelle lors de l'apprentissage. Toutefois, même si en théorie cette propriété les rend particulièrement adaptés au traitement des séquences, en pratique les RNNs "classiques" sont incapables de traiter des séquences faisant intervenir des écarts temporels supérieurs à 10 instants entre les entrées et les sorties désirées correspondantes .

En effet, plusieurs travaux [Hoc91, BSF94] ont démontré que l'influence d'une entrée donnée sur les couches cachées (et donc sur les sorties du réseau) augmente ou se dissipe de manière exponentielle au fur et à mesure qu'elle passe par les connections récurrentes. Ceci est dû au fait que l'erreur locale à un instant t rétro-propagée dans le temps (pour les algorithmes d'apprentissage BPTT et RTRL) s'exprime de manière récursive en fonction des erreurs rétro-propagées aux instants passés. Ce problème est connu sous le nom de "décroissance exponentielle de l'erreur" (*exponential error decay*) [Hoc91, BSF94]. Pour plus de détails concernant ce problème, une démonstration théorique explicite a été présentée par Hochreiter et al. pour l'algorithme BPTT [HBF01].

Plusieurs solutions ont été proposées pour remédier au problème de la décroissance exponentielle de l'erreur [Moz93, BSF94, LHTG96], mais la plus utilisée dans l'état de l'art est celle des réseaux récurrents à longue mémoire à court terme (LSTM pour *long short-term memory*) [HS97].

L'architecture LSTM consiste en un ensemble de sous-réseaux récurrents particuliers (appelés "blocs de mémoire") situés au niveau de la couche cachée, et contenant chacun une ou plusieurs "cellules de mémoire". Cette architecture est définie par deux idées clés que nous allons présenter dans ce qui suit.

La première idée clé architecturale des réseaux LSTM est l'introduction d'un noeud spécial appelé CEC (pour *Constant Error Carousel*) qui possède une connexion autorécurrente avec un poids constant égal à $1,0$. Ce noeud assure la rétro-propagation d'une erreur constante dans le temps en l'absence de "nouvelles" entrées. Ceci permet donc de résoudre en partie le problème de la décroissance exponentielle de l'erreur. De manière plus intuitive, le CEC peut être vu comme une unité qui permet de "collecter" et de "conserver" les informations jugées pertinentes tout au long de la séquence, et de les "présenter" au reste du réseau.

La seconde idée clé est l'utilisation de "portes" multiplicatives qui sont des fonctions d'activation dont la sortie est un coefficient multiplicatif permettant d'*ouvrir* ou de *fermer* une connexion donnée. Dans la première version de l'architecture LSTM introduite

par Hochreiter et al. [HS97], chaque bloc de mémoire comportait deux portes multiplicatives, une pour l'entrée et une pour la sortie. Le rôle de ces portes est, d'une part, protéger le contenu du CEC des activations provenant des nouvelles entrées (pour le cas des portes d'entrée), et d'autre part, protéger le reste du réseau du contenu du CEC si celui-ci n'est pas pertinent (pour le cas des portes de sortie). Typiquement, tant que la porte d'entrée reste fermée (c'est-à-dire que son activation est proche de 0), l'activation du CEC sera gardée constante et ne sera pas mise à jour en fonction des nouvelles activations arrivant à l'entrée du bloc de mémoire. De même, tant que le contenu du CEC est jugé pertinent pour le reste du réseau, la porte de sortie sera maintenue ouverte. L'ouverture et la fermeture des portes sont apprises automatiquement à partir des données d'apprentissage.

D'autres améliorations de l'architecture LSTM ont depuis été présentées. Gers et al. [Ger01] ont proposé l'introduction d'une porte multiplicative supplémentaire (appelée *porte d'oubli*) afin de permettre à la cellule de remettre à zéro le contenu du CEC. En effet, les travaux de Gers et al. ont permis de démontrer que, même si l'architecture LSTM proposée par Hochreiter et al. [HS97] permet d'apprendre des tâches faisant intervenir des écarts temporels supérieurs à 10 instants entre l'entrée et la sortie désirée (ce qui n'était pas possible avec les RNNs classiques), l'état du CEC arrivait à saturation au delà d'une cinquantaine d'instants d'écart [Ger01]. L'idée est donc de placer une porte multiplicative au niveau de la connexion récurrente constante du CEC. Ainsi, le CEC pourra "mémoriser" les activations utiles tant que la porte d'oubli est ouverte, et de les "oublier" une fois que celle-ci est fermée.

Une autre amélioration proposée par Gers et al. [Ger01] est l'utilisation des *peepholes*, qui sont des connections supplémentaires entre le CEC et les différentes portes multiplicatives, qui permettent à ces derniers d'*espionner* le contenu du CEC, leur donnant ainsi accès à des informations supplémentaires lors de leurs ouvertures et fermetures. Concrètement, la porte de sortie est directement reliée au CEC par une connexion supplémentaire, et les portes d'entrée et d'oubli sont reliés au CEC par des connections introduisant un décalage temporel $\Delta t = 1$ (vu que ces deux portes ont besoin d'accéder à l'état du CEC à l'instant précédent et non pas courant). La Figure 3.6 résume tous ces concepts introduits dans l'architecture LSTM proposée par Gers et al. [Ger01].

Tout comme pour les RNNs classiques, l'architecture LSTM a aussi été étendue au cas bidirectionnel. Graves et al. [Gra08] ont en effet présenté une version bidirectionnelle de cette architecture (appelée BLSTM pour *Bidirectional LSTM*). Cette extension reprend le même principe que l'extension des RNNs unidirectionnels au cas bidirectionnel, à savoir utiliser deux couches cachées (une pour chaque direction) afin d'augmenter la

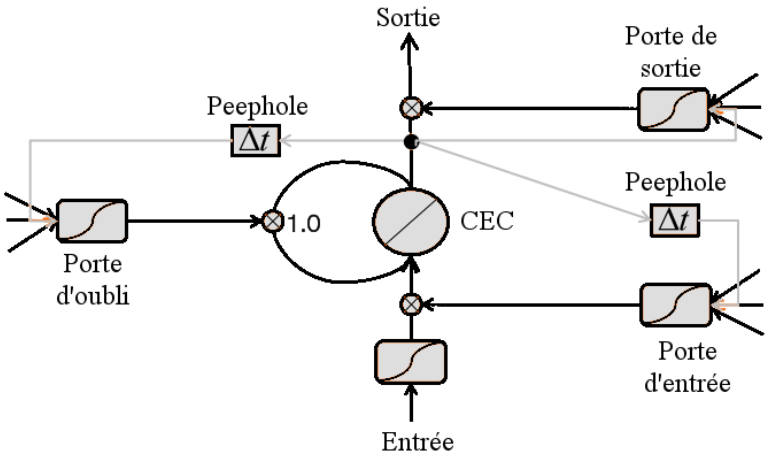

FIGURE 3.6 – Architecture LSTM proposée par Gers et al. [Ger01] : Illustration d'un bloc mémoire contenant une seule cellule.

quantité d'information de contexte dont dispose le réseau à un instant t.

Les réseaux LSTM unidirectionnels et bidirectionnels ont été testés pour de nombreuses applications de traitement de séquences : Reconnaissance de la parole [GS05, FGS07, WEK+09], de visages [LCS+08], d'émotions [WKES12], d'écriture manuscrite [GFL+08, GS09], structuration de vidéos de sport [Del06], apprentissage de rythmes [GSS03], composition de musiques [ES02], apprentissage de grammaires [GS01, SGE02, POGES03], prédiction de séries temporelles [SWG05], analyse structurelle de séquences de protéines [HHO07], et contrôle de robots [MGW+08]. Pour chacune de ces applications, les réseaux LSTM ont obtenu de meilleurs résultats que les RNNs classiques, ainsi que d'autres modèles de classification de séquences de l'état de l'art. A noter aussi que parmi ces applications étudiées, l'apport des réseaux LSTM par rapport aux autres modèles de classification est d'autant plus important que l'application étudiée fait intervenir des informations contextuelles à long terme. Les réseaux LSTM semblent donc bien adaptés au cas de la vidéo.

3.5 Conclusion

Dans ce chapitre, nous avons présenté trois modèles de classification de séquences parmi les plus utilisés de l'état de l'art, à savoir les modèles graphiques probabilistes (et plus particulièrement les modèles CRFs et HCRFs), les machines à vecteurs de support, et les réseaux de neurones récurrents LSTM. Pour chacun de ces modèles, nous avons dé-

crit succinctement leurs fondements théoriques, puis présenté les différentes méthodes d'apprentissage de leurs paramètres. Nous nous sommes aussi intéressés à l'utilisation pratique de ces modèles, à travers les travaux qui les ont appliqué à la classification de séquences (audio et vidéo).

Même si les différents modèles de classification cités précédemment ont été largement utilisés dans la littérature, rares sont les travaux qui ont comparé directement leurs performances sur une problématique de classification vidéo donnée. Nous allons mener dans le chapitre suivant une étude comparative entre ces modèles, prenant comme cadre applicatif la classification d'actions dans les vidéos de football. Les résultats de cette étude comparative serviront à choisir le modèle de classification que nous allons utiliser pour la suite.

Deuxième partie

Contributions de la thèse

Chapitre 4

Classification des vidéos de sport : Intégration du mouvement dominant et étude comparative

Sommaire

4.1 Introduction

Après avoir étudié dans la première partie de ce manuscrit l'état de l'art des différentes caractéristiques visuelles utilisées pour le cas de la classification vidéo, ainsi que les modèles de classification de séquences, nous allons présenter dans ce chapitre les deux premières contributions de cette thèse, à savoir : (i) L'introduction d'une nouvelle approche de classification de vidéos de sport, et (ii) une étude comparative entre différents modèles de classification de séquences.

Ainsi, nous allons nous intéresser à la problématique de la classification d'actions de football sur une base de séquences vidéo introduite par Ballan et al. [BBBS09]. Nous étudierons les caractéristiques manuelles adaptées à ce contenu, en commençant par celles proposées par Ballan et al. [BBBS09], et montrerons que l'introduction de nouvelles caractéristiques, décrivant le mouvement dominant de la scène permet d'améliorer considérablement les résultats de l'état de l'art sur la base d'actions de football utilisée.

Nous allons aussi mener dans ce chapitre une étude comparative dans laquelle nous évaluerons et comparerons les performances (en terme de taux de classification) des principaux modèles de classification présentés dans le chapitre 3. L'idée est d'utiliser en entrée des différents types de classifieurs de séquences les mêmes caractéristiques visuelles afin d'en déduire le plus discriminant.

Le reste de ce chapitre s'organisera comme suit : Nous présenterons tout d'abord dans la section 4.2 la base de vidéos de football étudiée. Les sections 4.3 et 4.4 s'intéresseront ensuite aux caractéristiques utilisées pour représenter les données de la base. Pour ce faire, nous présenterons dans un premier temps les caractéristiques visuelles utilisées dans l'article original décrivant la base (et qui seront utilisées pour l'étude comparative), puis nous introduirons l'approche basée sur les nouvelles caractéristiques de mouvement. La section 4.5 s'intéressera ensuite aux modèles de classification utilisés. Enfin, nous présenterons dans la section 4.6 le protocole d'évaluation adopté ainsi que quelques résultats expérimentaux, avant de dresser un bilan dans la dernière section de conclusion.

4.2 Problématique étudiée

Nous avons choisi de nous intéresser à la classification de vidéos de football, d'une part à cause des enjeux applicatifs et commerciaux importants qui y sont liés, mais aussi vu que ce type de contenus est généralement complexe et difficile à classer.

Plusieurs bases publiques de vidéos de football ont été présentées dans la littérature, nous avons choisi la base *MICC-Soccer-Actions-4*[1] qui a été introduite en 2009 par Ballan et al. [BBBS09]. Elle comprend une centaine de séquences vidéo encodées au format *MPEG-2* pleine résolution *PAL* (c'est à dire une résolution de 720×576 pixels, et 25 images/seconde). La base contient quatre types d'actions : *Six-mètres* (*goal-kick*), *coup franc* (*placed-kick*), *tir-au-but* (*shot on goal*) et *touche* (*throw-in*), qui sont illustrées sur la Figure 4.1.

Les séquences ont été générées par une détection de plans à partir de 5 vidéos de

[1] Disponible en ligne sur : http://www.micc.unifi.it/ballan/research/video-events/

FIGURE 4.1 – Quelques exemples correspondant aux quatre actions de la base *MICC-Soccer-Actions-4*.

matchs complets du championnat de football italien (saison 2007/2008), faisant intervenir 7 équipes et 5 stades différents, ainsi que des conditions d'éclairage variables (notamment 4 matchs se jouant sous un éclairage naturel à différentes heures de la journée, et le dernier sous un éclairage artificiel). Chacune des quatre classes est représentée par 25 séquences de longueurs variables, allant de 100 images (4 secondes) à 2500 images (100 secondes). L'angle de prise est le même pour toutes les séquences (vue globale du terrain qui suit le déroulement de l'action), mais la variabilité intra-classe reste néanmoins très importante puisque les actions se déroulent suivant plusieurs scénarii différents. Des ressemblances importantes existent aussi entre certaines classes (notamment entre *tir-au-but* et *coup franc*), ce qui augmente les confusions inter-classes. Une difficulté supplémentaire réside dans le fait que dans la plupart des vidéos, l'action est

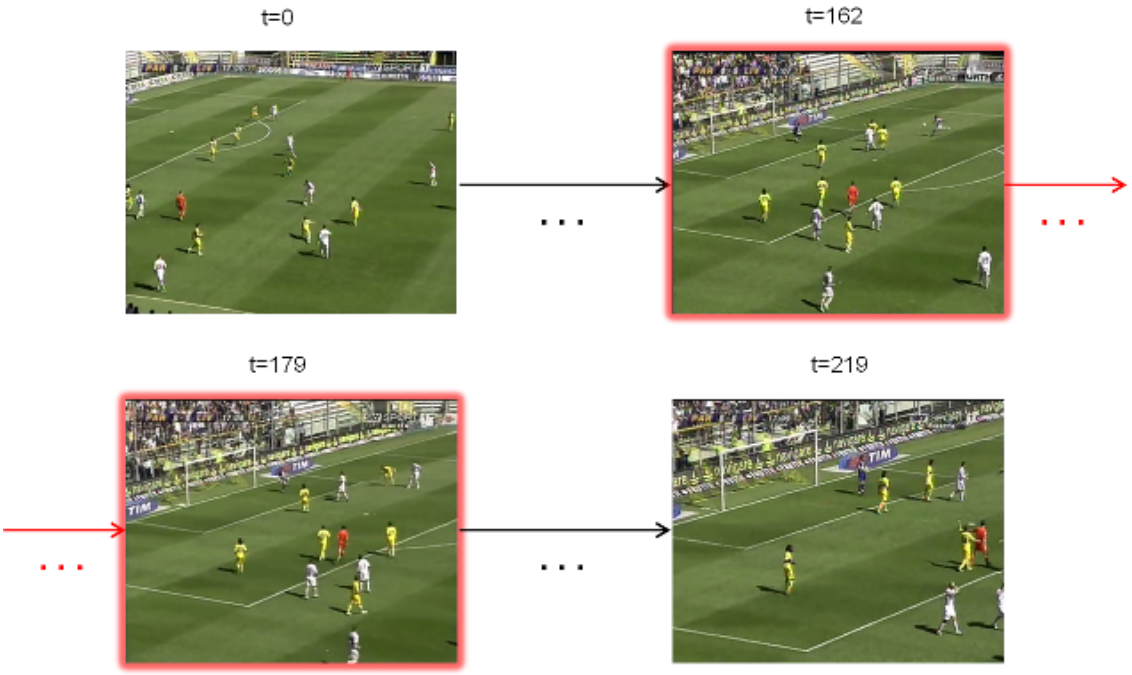

FIGURE 4.2 – Illustration sur un exemple de la base *MICC-Soccer-Actions-4* de la localisation temporelle des actions. La couleur rouge désigne les images correspondant à l'action *tir-au-but* (17 images sur 219).

temporellement localisée uniquement sur quelques instants successifs de la séquence, et que le reste des instants peut fausser la classification. La Figure 4.2 illustre un exemple où l'action est localisée dans 17 images sur les 219 que compte la vidéo.

Dans les deux sections suivantes, nous allons présenter les caractéristiques utilisées pour représenter les données de la base *MICC-Soccer-Actions-4*.

4.3 Les sacs de mots visuels

Dans cette section, nous allons nous intéresser aux caractéristiques visuelles utilisées dans l'article original de Ballan et al. [BBBS09]. Ces caractéristiques représentent une vidéo par une séquence de sacs de mots visuels. Concrètement, un détecteur de points d'intérêts SIFT est d'abord appliqué sur un large nombre d'images extraites des vidéos. Un dictionnaire de mots est ensuite généré en appliquant un *clustering* par k-moyennes sur les descripteurs de ces points d'intérêts (cf. sous-section 2.2.3.2 du chapitre 2). La séquence de caractéristiques est enfin générée en calculant, pour chaque image, l'histogramme des mots visuels formant le dictionnaire. Le descripteur pour chaque image a donc la taille du dictionnaire et chaque valeur représente la fréquence d'occurrence du mot du dictionnaire dans l'image. La Figure 4.3 illustre le processus de génération de

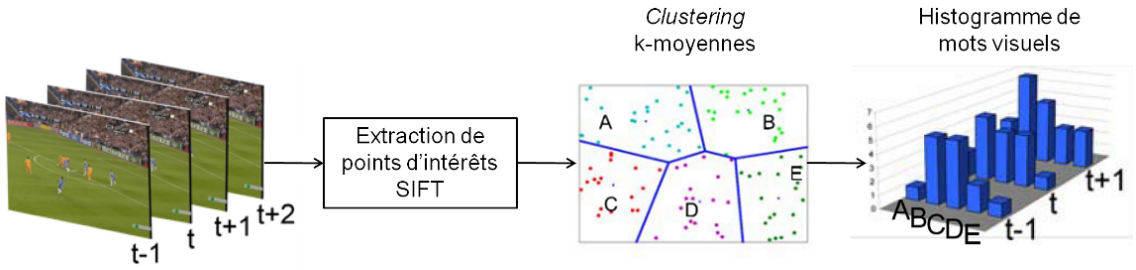

FIGURE 4.3 – Caractéristiques visuelles introduites par Ballan et al. [BBBS09] : Chaque vidéo est représentée par une séquence d'histogrammes de mots visuels. Illustration sur un exemple de la base *MICC-Soccer-Actions-4*.

ces caractéristiques sur un exemple de la base *MICC-Soccer-Actions-4*.

Ballan et al. ont utilisé ces caractéristiques pour entraîner deux types de classifieurs, à savoir une recherche des k plus proches voisins, et une machine à vecteurs de support à noyau de séquences. Ceci va donc nous servir de base pour évaluer, sur les données de *MICC-Soccer-Actions-4* et dans les mêmes conditions que [BBBS09], les performances de différents modèles de classification de séquences (cf. section 4.6).

Cependant, même si cette représentation permet de prendre en compte à la fois le contenu visuel et temporel de la vidéo (en modélisant les transitions entre les images et l'apparition ou la disparition des mots visuels), elle ne fait intervenir aucune notion de mouvement.

Nous allons dans la section suivante introduire un autre type de caractéristiques décrivant le mouvement dominant de la scène.

4.4 Intégration du mouvement dominant

Nous proposons dans cette section une nouvelle approche basée sur des caractéristiques décrivant le mouvement dominant. Nous définissons ce dernier comme étant le mouvement décrit par le plus grand nombre d'éléments d'une scène donnée. Typiquement, pour une action de football avec une vue globale du terrain (comme c'est le cas pour les actions de la base *MICC-Soccer-Actions-4*), le mouvement dominant se confond avec celui de la caméra. L'idée est de profiter de la connaissance de la scène par le réalisateur, qui se traduit par des mouvements de la caméra qui sont très caractéristiques des types d'actions. En effet, la manière de filmer une action de football donnée (zooms, mouvements, angle de vue, ...) est le résultat d'une modélisation de cette action par le réalisateur, faisant intervenir des connaissances a priori. Nous proposons donc d'exploiter ces informations haut-niveau, qui sont issues d'une interprétation des actions faite

par un cerveau humain.

Dans ce qui suit, nous allons présenter une approche qui permet d'estimer ce mouvement, et de l'exploiter pour la classification.

Estimation du mouvement dominant par appariement de points SIFT : Nous avons fait l'hypothèse d'un mouvement affine de la caméra, ce qui est généralement vérifié. Le principe est donc d'estimer la transformation affine T qui permet de passer d'une image I_t d'une vidéo à un instant t à l'image I_{t+1} à l'instant suivant.

Ceci peut être fait en utilisant une approche basée sur l'appariement des points SIFT [Low04] entre deux images successives de la vidéo, ce qui permet d'estimer la transformation affine entre ces deux images. L'appariement se fait par une recherche rapide du voisin le plus proche (au sens de la distance euclidienne entre les descripteurs SIFT) en utilisant un arbre k-dimensionnel (*kd-tree*) [BL97].

Si nous notons par $\left(x_i^{(t)}, y_i^{(t)}\right)$ pour $i \in \{1, \ldots, N\}$ les N coordonnées des points SIFT détectés sur l'image I_t et qui ont été appariés avec un point de l'image I_{t+1} de coordonnées $\left(x_i^{(t+1)}, y_i^{(t+1)}\right)$, la relation entre $\left(x_i^{(t)}, y_i^{(t)}\right)$ et $\left(x_i^{(t+1)}, y_i^{(t+1)}\right)$ sera de la forme :

$$\begin{bmatrix} x_i^{(t+1)} \\ y_i^{(t+1)} \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x_i^{(t)} \\ y_i^{(t)} \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} \tag{4.1}$$

où les a_i sont les coefficients relatifs à la rotation et au facteur d'échelle et les t_i ceux relatifs à la translation. En ré-écrivant l'équation 4.1 pour les N points appariés, nous pouvons nous ramener à un système linéaire de la forme :

$$A \cdot T = B \tag{4.2}$$

où :

$$A = \begin{bmatrix} & & \cdots & & & \\ x_i^{(t)} & y_i^{(t)} & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i^{(t)} & y_i^{(t)} & 0 & 1 \\ & & \cdots & & & \end{bmatrix} \tag{4.3}$$

T est la transformation affine à estimer :

$$T = [a_1 \; a_2 \; a_3 \; a_4 \; t_1 \; t_2]^T \tag{4.4}$$

et

$$B = \begin{bmatrix} \dots \\ x_i^{(t+1)} \\ y_i^{(t+1)} \\ \dots \end{bmatrix} \tag{4.5}$$

Le système d'équations 4.2 étant sur-déterminé, de manière générale il n'existe donc pas de solution exacte. Une solution optimale, dans le sens des moindres carrés, peut être exprimée par :

$$T = A^+ \cdot B \tag{4.6}$$

où A^+ est la matrice pseudo-inverse de Moore-Penrose de A, obtenue en décomposant cette dernière en valeurs singulières. Concrètement, A est factorisée comme suit :

$$A = U \cdot \Sigma \cdot V^T \tag{4.7}$$

où U et V sont des matrices carrées, et Σ est une matrice diagonale contenant les valeurs singulières de A. La matrice pseudo-inverse de Moore-Penrose de A est alors exprimée par :

$$A^+ = V \cdot \Sigma^{-1} \cdot U^T \tag{4.8}$$

Afin de ne garder que les appariements qui correspondent au mouvement dominant, et d'éliminer ceux correspondant aux mouvements locaux (par exemple ceux des joueurs), l'algorithme *RANSAC* [Fis81] est appliqué durant la résolution de l'équation 4.6 pour sélectionner les points "aberrants" (*outliers* en anglais) et les points "conformes" (*inliers* en anglais).

Concrètement, vu que l'équation 4.2 est à six inconnues, un triplet de points appariés est sélectionné itérativement et de manière aléatoire par l'algorithme RANSAC, afin d'estimer la transformation T correspondante. Le résidu de l'estimation est alors calculé comme étant la distance entre la position estimée du point dans la deuxième image et sa position réelle. Si cette distance est supérieure à un certain seuil, le point (et donc la correspondance) est considéré comme *outlier*. Sinon il est considéré comme *inlier*.

Ce processus est répété pour tous les triplets de points, et l'estimation finale de T est obtenue en utilisant uniquement les points sélectionnés comme étant des *inliers*. La figure 4.4 montre un exemple d'appariement de points entre deux images issues

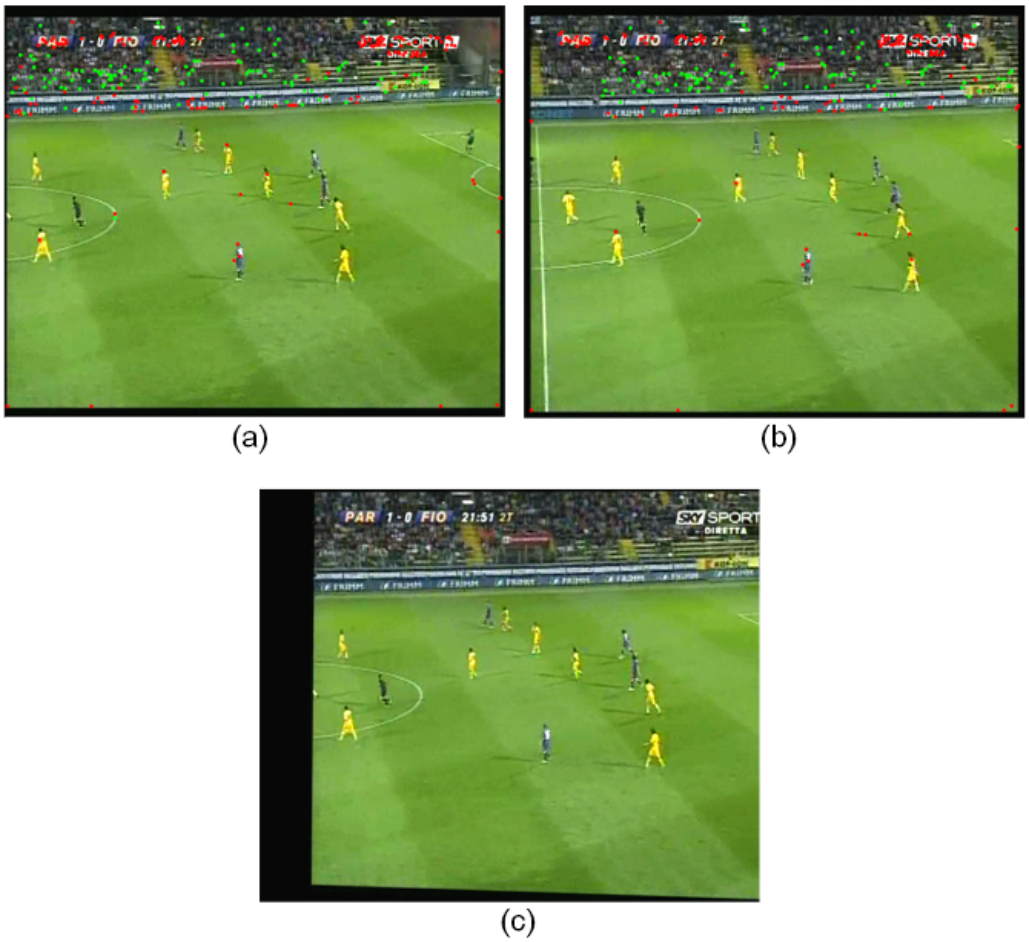

FIGURE 4.4 – Exemple d'estimation du mouvement affine entre deux images successives : (a,b) - *Inliers* (en vert) et *outliers* (en rouge) appariés entre les deux images (c) - Compensation du mouvement sur la première image.

de la même vidéo (Figures 4.4-(a,b)), ainsi que la compensation du mouvement affine estimé (Figure 4.4-(c)).

Nous remarquons sur cette figure que les *inliers* (représentés par des points verts) sont localisés au niveau du public (qui présente de fortes textures, et dont le mouvement correspond à celui de la caméra), alors que les *outliers* sont principalement localisés au niveau des joueurs et des logos et textes incrustés.

A noter que dans certains cas, le nombre de points SIFT détectés au niveau des logos et textes incrustés (scores, temps, ...) est plus important que celui correspondant aux zones texturées en mouvement, ce qui peut fausser le résultats de l'estimation, et conduire à un mouvement nul.

Le paragraphe suivant décrit la solution que nous avons adoptée pour remédier à ce problème.

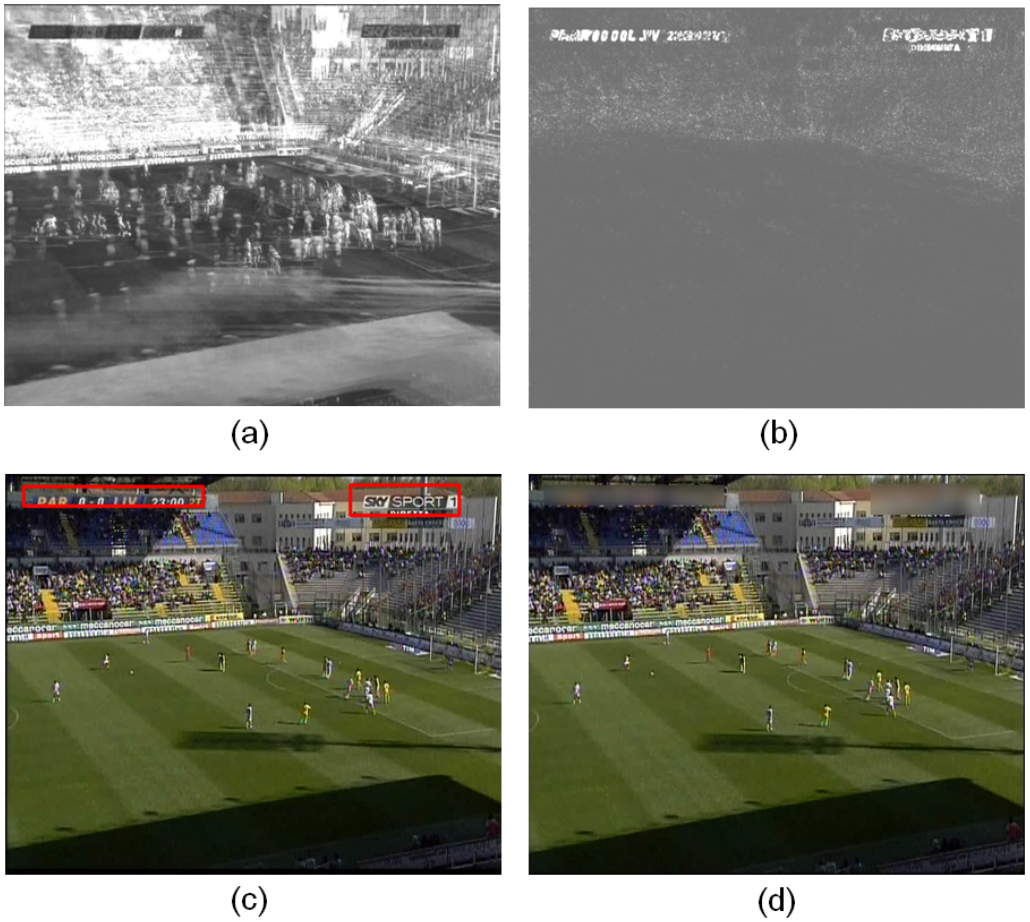

FIGURE 4.5 – Détection et floutage des logos et des textes incrustés : (a) - Carte de variance temporelle calculée à partir d'une vingtaine d'images sélectionnées aléatoirement (b) - Carte de score caractérisant les textes horizontaux (c) - Résultat de l'application de la recherche des rectangles englobants sur la carte des scores (d) - Lissage Gaussien $2D$ des pixels détectés comme appartenant à un logo ou à un texte incrusté.

Détection et floutage des logos et textes incrustés : Afin d'éviter que les logos ne soient pris en considération dans l'estimation, nous effectuons un pré-traitement sur toutes les vidéos pour détecter et flouter ces logos et textes incrustés. Pour ce faire, deux approches sont combinées. La première permet de détecter, à partir de l'analyse statistique de la vidéo, les pixels immobiles. Concrètement, nous calculons une carte de variance temporelle sur une vingtaine d'images sélectionnées aléatoirement (cf. la Figure 4.5-(a)) et nous caractérisons les pixels où la variance est faible. La deuxième approche s'inspire de la méthode introduite par Wolf et al. dans [WJC02] pour la détection des textes, et qui consiste à moyenner les gradients accumulés selon l'axe horizontal sur une vingtaine d'images sélectionnées aléatoirement, permettant ainsi d'obtenir une carte de scores caractérisant les textes horizontaux (cf. Figure 4.5-(b)).

Vu que les deux cartes obtenues caractérisent des informations complémentaires, nous proposons de les fusionner (par un ET logique) afin de générer une carte des scores, qui, après seuillage, permet de localiser les pixels pour lesquels les deux critères (une faible variabilité de leur valeur dans le temps, et la présence d'un texte horizontal) sont respectés. Un algorithme de recherche des rectangles englobants permet alors d'affiner le résultat, en associant une forme rectangulaire aux pixels détectés (cf. la Figure 4.5-(c)). La Figure 4.5-(d) montre aussi le résultat de l'application d'un lissage Gaussien $2D$ sur les pixels de ces rectangles détectés, introduisant ainsi un flou qui empêche la détection de points SIFT sur ces logos et textes incrustés.

4.5 Modèles de classification utilisés

Dans les deux sections précédentes, nous avons présenté les caractéristiques "manuelles" qui seront utilisées pour décrire les séquences vidéo. Dans cette section, nous nous intéresserons aux caractéristiques architecturales et aux paramètres utilisés pour les modèles de classification de séquences étudiés. Pour les deux premiers modèles, nous reprendrons les architectures et paramètres proposés initialement par Ballan et al. [BBBS09], et utiliserons leurs résultats pour servir de base de comparaison.

La recherche des k plus proches voisins : Le premier modèle de classification utilisé dans les expérimentations de Ballan et al. [BBBS09] est une recherche des k plus proches voisins (k-ppv), basée sur la distance d'édition de Needleman-Wunsch [NW70], afin de gérer les séquences de longueurs différentes.

Le modèle SVM à noyau de séquences : Ballan et al. [BBBS09] ont aussi proposé un modèle SVM à noyau de séquences. Tout comme le classifieur k-ppv, ce modèle se base sur une distance d'édition de Needleman-Wunsch [NW70]. Cette stratégie d'adaptation a été décrite plus en détail dans la section 3.3 du chapitre 3. Concernant le modèle SVM, les auteurs ont utilisé la librairie LIBSVM proposée par Chang et Lin dans [CL11], et la stratégie "un contre tous" pour la classification multi-classes.

Le modèle HCRF : Nous proposons d'évaluer les performances du classifieur HCRF (cf. section 3.2 du chapitre 3). Pour ce faire, nous avons utilisé l'implémentation de Morency et Stratou[2]. La taille des observations est égale à celle des vecteurs de descriptions des caractéristiques utilisées. Le nombre d'états cachés ainsi que l'ordre

[2]Disponible en ligne sur : http://sourceforge.net/projects/hcrf/

des dépendances temporelles ont été fixés respectivement à 50 et 3, de manière empirique. Ce modèle correspond à un nombre total d'environ $2 \cdot 10^4$ paramètres, selon la taille des observations. Le modèle a été entraîné avec l'implémentation de Hager et Zhang [HZ06] de la descente du gradient conjugué.

Les réseaux de neurones LSTM : Nous avons testé deux types de réseaux LSTM : Un réseau unidirectionnel et un réseau bidirectionnel (qui seront notés respectivement LSTM et BLSTM dans la section 4.6 des résultats expérimentaux). Pour le premier, nous avons utilisé la version introduite par Gers et al. [Ger01], c'est à dire celle qui contient en plus du CEC et des deux portes multiplicatives d'entrée et de sortie une troisième porte (dite porte d'oubli) et des connections *peepholes* (voir Figure 3.6 du chapitre 3). Pour le modèle bidirectionnel, nous avons utilisé le réseau de Graves et al. [Gra08], qui est une simple extension du modèle de Gers et al. au cas bidirectionnel. Pour plus de détails, se référer à la section 3.4 du chapitre 3.

Chacun des deux modèles contient trois couches : (i) Une couche d'entrée dont la taille est la même que celle des vecteurs de caractéristiques, (ii) une couche de sortie de taille 4 (une sortie par classe), avec des fonctions d'activation de type *Soft-Max*, et (iii) une couche cachée comportant des neurones LSTM unidirectionnels ou bidirectionnels, totalement inter-connectés et connectés au reste du réseau.

Nous avons noté qu'une augmentation importante du nombre de neurones LSTM conduit à un sur-apprentissage du réseau (et augmente considérablement la complexité). De même, un réseau de taille réduite conduit à une divergence de l'apprentissage. Nos expérimentations ont montré que 150 LSTMs (et deux fois 150 pour le BLSTM) pour la couche cachée permettent d'obtenir les meilleurs taux de classification pour cette base. Ainsi le nombre total de poids à optimiser est d'environ 10^5 (selon la taille de la couche d'entrée) pour le réseau unidirectionnel, et le double pour le réseau bidirectionnel. Les deux modèles ont été entraînés, en visant à chaque instant la classe de la séquence d'entrée, avec une version modifiée de l'algorithme BPTT, qui a été introduite par les auteurs [Ger01, Gra08].

4.6 Résultats expérimentaux

Nous allons décrire dans un premier temps le protocole d'évaluation adopté pour la base *MICC-Soccer-Actions-4*. Nous présenterons ensuite l'évaluation des modèles de classification étudiés en comparant, dans les mêmes conditions, les résultats obtenus avec ceux présentés dans [BBBS09]. Puis nous étudierons l'apport des nouvelles caractéristiques

de mouvement dominant, ainsi que sa complémentarité avec les sacs de mots visuels. A noter toutefois que pour les deux premiers classifieurs (à savoir la recherche des k plus proches voisins, et le classifieur SVM à noyau de séquences), nous n'avons pas reproduit les expérimentations, mais uniquement repris les résultats publiés par Ballan et al. [BBBS09].

4.6.1 Protocole d'évaluation

Nous avons gardé le même protocole d'évaluation que celui utilisé dans l'article original de Ballan et al. [BBBS09], à savoir que toutes les expérimentations correspondent à une validation croisée avec un partitionnement de la base en 3 parties (*3-fold cross validation*). Pour chacune de ces configurations, environ 2/3 des séquences sont utilisées pour l'apprentissage, et le 1/3 restant pour le test, selon la répartition décrite dans le tableau 4.1, de manière à ce que toutes les séquences soient testées une seule fois.

	Nombre de séquences utilisées pour l'apprentissage	Nombre de séquences utilisées pour le test
Config.1	68 (17/classe)	32 (8/classe)
Config.2	68 (17/classe)	32 (8/classe)
Config.3	64 (16/classe)	36 (9/classe)
	Total :	100 (25/classe)

TABLE 4.1 – Répartition du nombre de séquences entre apprentissage et test pour la validation croisée.

Le critère utilisé pour l'évaluation est celui du taux de bonne classification, en moyennant les résultats individuels de chacune des trois configurations.

4.6.2 Évaluation des performances des modèles de classification étudiés

L'évaluation a été faite en utilisant les caractéristiques visuelles présentés dans la section 4.3 de ce chapitre. Pour ce faire, la détection des points SIFT ainsi que le calcul des descripteurs sont effectués en utilisant la librairie *OpenSIFT* de Rob Hess [Hes10][3]. Un dictionnaire de 30 mots visuels est ensuite généré à partir d'une partie de la base (5 images extraites aléatoirement de chacune des 100 vidéos) avec une classification k-moyennes. Nous avons vérifié que l'augmentation de la taille du dictionnaire n'améliorait pas les résultats, mais augmentait considérablement la complexité, ce qui est en conformité avec les observations présentées dans [BBBS09]. Ceci peut s'expliquer par le fait que toutes les vidéos représentent une vue globale du terrain, et que la scène présente donc peu de détails. Ainsi, elle peut être représentée par un nombre limité de mots

[3]Disponible en ligne sur : http://robwhess.github.com/opensift/.

	k-ppv	SVM	HCRF	LSTM	BLSTM
Config.1	-	-	71,88	84,38	87,50
Config.2	-	-	68,75	71,88	75,00
Config.3	-	-	69,44	72,22	75,00
Moyenne	**52,75**	**73,25**	**70,02**	**76,16**	**79,17**

TABLE 4.2 – Taux de classification (en %) obtenus par les différents classifieurs étudiés, en utilisant les caractéristiques visuelles de Ballan et al. [BBBS09]. Les trois configurations correspondent à trois répartitions aléatoires des données entre apprentissage et test.

visuels, et l'augmentation de la taille du dictionnaire conduirait à une augmentation de la variabilité intra-classes en encodant des détails non discriminants entre les différents types d'actions.

Nous avons entraîné les différents classifieurs avec les séquences d'histogrammes de mots visuels, contenant les fréquences d'apparition par instant des 30 mots visuels. Le résultat de la classification pour chacun des modèles, est reporté sur le tableau 4.2, ainsi que les résultats présentés par Ballan et al. dans [BBBS09]. Nous reportons aussi sur la Figure 4.6 les matrices de confusion relatives aux classifieurs étudiés.

Comme attendu, la recherche des k plus proches voisins est la méthode de classification la moins performante, avec un taux de bonne classification de 52,75%. Le modèle HCRF obtient quant à lui des performances relativement faibles (70,02% de bonne classification), et surtout moins bonnes que celles obtenues par le modèle SVM. Ceci est plutôt surprenant compte-tenu du fait que les modèles HCRFs ont été conçus pour le traitement des séquences, ce qui n'est pas le cas pour les SVMs, qui nécessitent d'être adaptées à la classification de telles données (cf. chapitre 3). Ces faibles résultats peuvent être expliqués par la nature des données traitées, dans le sens où les actions ne sont localisées temporellement que sur des portions de courte durée de la vidéo (un exemple est illustré sur la Figure 4.2). Ceci fait que les approches qui modélisent les séquences dans leur globalité (comme c'est le cas pour les SVMs combinés aux distances d'édition) conviennent mieux que celles qui les modélisent par des séquences d'états.

Le tableau 4.2 montre aussi que les deux modèles neuronaux LSTM sont plus performants que les autres classifieurs étudiés, avec des taux de classification respectifs de 76,16% et 79,17% pour le modèle unidirectionnel et bidirectionnel. Ceci s'explique par le fait que ces classifieurs sont particulièrement adaptés aux longues séquences (et donc au cas des vidéos), et que les mécanismes de portes (cf. section 3.4) permettent d'apprendre à sélectionner les instants où sont localisées les actions. Ceci confirme les études présentées dans l'état de l'art qui montraient que les modèles LSTM obtiennent souvent de meilleurs résultats que les autres modèles de classification quand la problé-

	Six mètres	Coup franc	Tir-au-but	Touche
Six mètres	**79,63**	12,50	03,70	04,17
Coup franc	07,41	**81,02**	00,00	11,57
Tir-au-but	03,70	19,44	**64,35**	12,50
Touche	12,50	12,04	20,37	**55,09**

(a)

	Six mètres	Coup franc	Tir-au-but	Touche
Six mètres	**91,67**	08,33	00,00	00,00
Coup franc	07,41	**81,02**	00,00	11,57
Tir-au-but	00,00	19,44	**72,69**	07,87
Touche	12,50	12,04	16,20	**59,26**

(b)

	Six mètres	Coup franc	Tir-au-but	Touche
Six mètres	**91,67**	08,33	00,00	00,00
Coup franc	07,41	**81,02**	00,00	11,57
Tir-au-but	00,00	15,74	**80,56**	03,70
Touche	08,33	12,04	16,20	**63,43**

(c)

FIGURE 4.6 – Matrices de confusion relatives aux différents classifieurs entrainés avec les caractéristiques visuelles de Ballan et al. [BBBS09] : (a) - Modèle HCRF (b) - Modèle LSTM (c) - modèle BLSTM.

matique fait intervenir des données temporelles (cf. la section 3.4 du chapitre 3). Nous remarquons aussi que le modèle bidirectionnel obtient de meilleur résultats que le modèle unidirectionnel, vu que ce dernier dispose de moins d'informations de contexte lors de la classification.

Cette première série d'expérimentations a permis de sélectionner le classifieur neuronal récurrent à longue mémoire à court-terme, et plus particulièrement sa version bidirectionnelle BLSTM, comme étant le plus discriminant pour la classification de séquences vidéo. Nous retiendrons donc ce modèle de classification pour le reste des expérimentations de cette thèse.

	LSTM	BLSTM
Config.1	68,75	75,00
Config.2	75,00	78,13
Config.3	86,11	88,89
Moyenne	**76,62**	**80,67**

TABLE 4.3 – Taux de classification (en %) obtenus par les modèles LSTM et BLSTM, entraînés avec les caractéristiques de mouvement dominant. Les trois configurations correspondent à trois répartitions aléatoires des données entre apprentissage et test.

	Six mètres	Coup franc	Tir-au-but	Touche
Six mètres	**62,96**	29,17	07,87	00,00
Coup franc	07,41	**68,52**	07,41	16,67
Tir-au-but	08,33	00,00	**87,50**	04,17
Touche	08,33	00,00	04,17	**87,50**

(a)

	Six mètres	Coup franc	Tir-au-but	Touche
Six mètres	**70,83**	25,00	04,17	00,00
Coup franc	07,41	**72,22**	03,70	16,67
Tir-au-but	00,00	04,17	**91,67**	04,17
Touche	04,17	04,17	03,70	**87,96**

(b)

FIGURE 4.7 – Matrices de confusion relatives aux modèles neuronaux entrainés avec les caractéristiques de mouvement dominant : (a) - Modèle LSTM (b) - Modèle BLSTM.

4.6.3 Évaluation de l'apport du mouvement dominant

Nous allons dans un premier temps évaluer l'intérêt des caractéristiques de mouvement dominant pour la classification des actions de football, en utilisant les modèles de classification neuronaux (unidirectionnel et bidirectionnel). Nous étudierons ensuite la possibilité d'une combinaison entre les deux types de caractéristiques (sacs de mots visuels et mouvement dominant).

Pour cette étude, nous avons utilisé les mêmes réseaux que pour les expérimentations précédentes, en modifiant la taille de la couche d'entrée qui contient maintenant six valeurs par instant, ce qui correspond aux six coefficients de la transformation affine entre deux images successives. Pour chaque coefficient, nous calculons la moyenne m et l'écart type σ sur la base d'apprentissage. Les six valeurs d'entrée à chaque instant sont ensuite normalisées, entre -1 et 1, en tronquant ces valeurs de manière à fixer les extrema à $m \pm 2\sigma$. Ceci permet de prendre en compte 98 % de la masse, en faisant l'hypothèse d'une distribution Gaussienne.

Les résultats ainsi obtenus sont reportés sur le tableau 4.3, ainsi que sur la Figure

4.7, qui montre les matrices de confusion correspondantes. La première conclusion de cette étude est que les résultats de la classification sont, en moyenne, comparables à ceux obtenus par l'approche basée sur les sacs de mots visuels. Ainsi, le mouvement dominant seul permet de reconnaître 76,62% et 80,67% des séquences, avec respectivement les modèles LSTMs unidirectionnel et bidirectionnel. Ces résultats sont surprenants compte-tenu du fait que la classification ne se base que sur le mouvement de la caméra, sans aucune indication supplémentaire sur le contenu visuel des vidéos, et montrent que le choix des caractéristiques "manuelles" est une étape empirique cruciale pour la classification. Le fait que les performances obtenues soient comparables à celles correspondant aux sacs de mots visuels est aussi un résultat intéressant du point de vue de la complexité algorithmique.

De plus, la comparaison entre les matrices de confusion des Figures 4.6-(b,c) d'un côté, et 4.7-(a,b) de l'autre montre une complémentarité entre l'information visuelle et le mouvement dominant. En effet, les classes *touche* et *tir* sont très adaptées à l'approche basée sur le mouvement dominant, vu que ce dernier se confond avec celui de la caméra qui est très caractéristique de l'une et de l'autre (mouvement quasi inexistant pour la première, et très caractéristique pour la deuxième, notamment à cause des zooms sur la cage de but). En revanche, les classes *six-mètres* et *coup franc* sont caractérisées par des scénarii très variables (en terme de mouvement de la caméra) mais sont particulièrement adaptées à l'approche par le contenu visuel vu que dans les deux cas, l'ordre dans lequel apparaissent / disparaissent les mots est très caractéristique (notamment les mots visuels caractérisant la cage de but, qui apparaissent au début de la séquence pour les *six-mètres* et disparaissent après, et inversement pour les *coup francs*).

Nous proposons donc de concaténer les deux types de caractéristiques. Nous avons ainsi repris les expérimentations précédentes, dans les mêmes conditions, mais en concaténant les vecteurs d'entrée qui sont maintenant de taille 36 (30 valeurs correspondant aux sacs de mots visuels, et 6 valeurs pour le mouvement dominant). Les résultats obtenus sont reportés sur le tableau 4.4. Nous présentons aussi sur la Figure 4.8 les matrices de confusion correspondantes.

L'approche proposée est capable de classer correctement 92,25% des séquences avec un réseau unidirectionnel, et 93,98% avec un réseau bidirectionnel. Ces résultats sont, à notre connaissance, les meilleurs de l'état de l'art sur la base *MICC-Soccer-Actions-4*, avec une amélioration importante de +20,73 points par rapport aux travaux de Ballan et al. [BBBS09]. L'apport des caractéristiques de mouvement que nous avons introduites dans ce chapitre par rapport à l'utilisation des caractéristiques visuelles seules est aussi très important, avec des améliorations de +16,09 et +14,81 points respectivement pour

	LSTM	BLSTM
Config.1	96, 88	96, 88
Config.2	93, 75	90, 63
Config.3	86, 11	94, 44
Moyenne	**92,25**	**93,98**

TABLE 4.4 – Taux de classification (en %) obtenus par les modèles LSTM et BLSTM, entraînés avec la concaténation des caractéristiques visuelles de Ballan et al. [BBBS09] et celles du mouvement dominant. Les trois configurations correspondent à trois répartitions aléatoires des données entre apprentissage et test.

	Six mètres	Coup franc	Tir-au-but	Touche
Six mètres	**100,0**	00,00	00,00	00,00
Coup franc	03,70	**84,72**	07,87	03,70
Tir-au-but	00,00	11,57	**88,43**	00,00
Touche	04,17	00,00	00,00	**95,83**

(a)

	Six mètres	Coup franc	Tir-au-but	Touche
Six mètres	**95,83**	00,00	04,17	00,00
Coup franc	00,00	**96,30**	03,70	00,00
Tir-au-but	12,04	00,00	**87,96**	00,00
Touche	00,00	00,00	04,17	**95,83**

(b)

FIGURE 4.8 – Matrices de confusion relatives aux modèles neuronaux entrainés avec la concaténation des caractéristiques visuelles de Ballan et al. [BBBS09] et celles du mouvement dominant : (a) - Modèle LSTM (b) - Modèle BLSTM.

la classification unidirectionnelle et bidirectionnelle. Ceci permet, encore une fois, de confirmer la pertinence de ces caractéristiques pour ce type de contenus, et donc l'importance du choix empirique de celles-ci. Enfin, et de manière plus générale, ces expérimentations ont permis aussi de démontrer que les modèles neuronaux récurrents LSTM sont capables de gérer des données de natures différentes (fréquences d'apparition de mots visuels, et paramètres de transformation affine), et de les exploiter efficacement pour la classification simplement en les juxtaposant, ce qui représente également un résultat intéressant de cette étude.

4.7 Conclusion

Nous avons présenté au cours de ce chapitre les deux premières contributions de cette thèse, à savoir l'introduction de nouvelles caractéristiques basés sur l'estimation du mouvement dominant de la scène, ainsi qu'une étude comparative entre différents modèles

de classification de séquences.

En effet, nous nous sommes basés sur les caractéristiques visuelles introduites par Ballan et al. [BBBS09] afin d'évaluer et de comparer les performances de cinq modèles de classification de séquences. Les résultats obtenus montrent que la classification neuronale récurrente à longue mémoire à court-terme (LSTM) obtient les meilleures performances, et plus particulièrement sa version bidirectionnelle (BLSTM). Ceci vient confirmer les conclusions faites dans d'autres travaux [Ger01, GS05, SWG05, GS09] pour différentes applications. Ces travaux (et plusieurs autres qui ont été cité dans le chapitre 3) ont démontré que ces modèles neuronaux LSTM sont d'autant plus efficaces lorsque les tâches étudiées font intervenir des dépendances temporelles d'ordre élevé (ce qui est le cas pour la classification de séquences vidéo). Cette étude comparative que nous avons mené vient donc s'ajouter à ces conclusions, et nous a permis de valider le modèle neuronal bidirectionnel BLSTM comme étant celui que nous utiliserons pour le reste de nos expérimentations.

Nous avons également introduit des nouvelles caractéristiques pour cette base d'actions de football, visant à décrire le mouvement dominant de chaque scène. Nous avons démontré que ces caractéristiques seules conduisaient à des résultats comparables à ceux obtenus avec les sacs de mots visuels, et que la concaténation des deux obtenait les meilleurs résultats de l'état de l'art sur la base *MICC-Soccer-Actions-4*, à savoir 92,25% de bonne reconnaissance avec le modèle neuronal unidirectionnel, et 93,98% avec le modèle bidirectionnel.

Ces résultats ont permis de démontrer que le choix des caractéristiques "manuelles" influe grandement sur les résultats de classification par les modèles neuronaux à longue mémoire à court-terme. Par ailleurs, le choix de ces caractéristiques optimales pour la tâche étudiée est empirique pour représenter au mieux le contenu des séquences vidéos (comme c'est le cas par exemple pour le mouvement dominant, qui possède un fort pouvoir discriminant entre les classes de la base *MICC-Soccer-Actions-4*, mais qui peut ne pas l'être pour d'autres vidéos de sport qui ne contiennent pas de vue globale du terrain, ou encore d'autres problématiques comme la reconnaissance d'actions humaines ou la reconnaissance d'expressions faciales).

Nous proposons, dans les deux chapitres suivants, de remplacer ce choix empirique de caractéristiques "manuelles", par des caractéristiques apprises automatiquement et sans aucune connaissance a priori.

Chapitre 5

Apprentissage supervisé profond de caractéristiques spatio-temporelles

Sommaire

5.1 Introduction

Après avoir étudié et comparé lors du chapitre précédent les différents modèles de classification de séquences, nous allons nous focaliser dans ce qui suit sur les caractéristiques utilisées en entrée du modèle sélectionné (i.e. un BLSTM).

Nous avons vu dans le chapitre 2, ainsi que dans notre étude présentée dans le chapitre 4, que les caractéristiques dites manuelles (c'est à dire reposant sur des connaissances a priori) étaient dépendantes de la problématique étudiée, et ne convenaient donc pas aux objectifs de généricité que nous nous sommes fixés dans le cadre de cette thèse. Nous allons donc étudier d'autres types de caractéristiques, apprises automatiquement à partir d'exemples, qui semblent mieux adaptées aux contraintes prises en compte dans le cadre de nos travaux.

Nous allons proposer dans ce chapitre un premier modèle d'apprentissage automatique de caractéristiques spatio-temporelles, qui appartient à une catégorie de modèles dits *profonds* (c'est à dire qui apprennent automatiquement une représentation multi-niveaux des données -cf. sous-section 2.3.3 pour plus de détails-). Ces modèles profonds ont suscité beaucoup d'intérêt lors des deux dernières décennies, surtout pour des applications $2D$ (classification d'images). L'un des modèles profond les plus populaires est le modèle neuronal ConvNet [LBBH98, LKF10]. Nous avons vu dans la sous-section 2.3.3 du chapitre 2 que même si ces modèles ont été utilisés avec succès dans plusieurs applications de traitement d'images fixes, leur extension au cas de la vidéo était encore un sujet ouvert, et que les rares travaux qui ont étudié cette problématique soit n'exploitaient pas l'information temporelle [NDL+05], soit n'étaient pas entièrement automatisés [KLY07, JXYY10], puisque reposant sur des caractéristiques extraites manuellement de l'image (gradients, flot optique, ...). Dans ce chapitre, nous proposons un modèle *ConvNet* qui : (i) Exploite l'information temporelle via des convolutions $3D$, et (ii) opère sur les données brutes n'ayant subi aucun pré-traitement complexe.

Au delà de l'apprentissage des caractéristiques en elles-mêmes, un autre point clé est l'étape de classification qui peut être entraînée soit à partir de caractéristiques déjà apprises, soit conjointement durant l'apprentissage de ces dernières. De plus, même si la phase de classification est entraînée de manière supervisée, l'extraction des caractéristiques peut se faire, quant à elle, de manière supervisée ou non. Plusieurs modèles (illustrés sur la Figure 5.1) peuvent ainsi êtres envisagés :

Première solution : Elle consiste en un schéma complet dans lequel il y'a un couplage total entre l'étape d'extraction de caractéristiques et celle de la classification (cf. Figure 5.1-(a)). Concrètement, dans le modèle *ConvNet*, l'étape de classification par MLP est remplacée par une classification de séquences (tel que le BLSTM), et le schéma global (regroupant les deux étapes) est entraîné de manière supervisée à attribuer un label à la séquence complète. L'une des difficultés est alors le fait que la profondeur du réseau (en nombre de couches et temporellement) qui implique

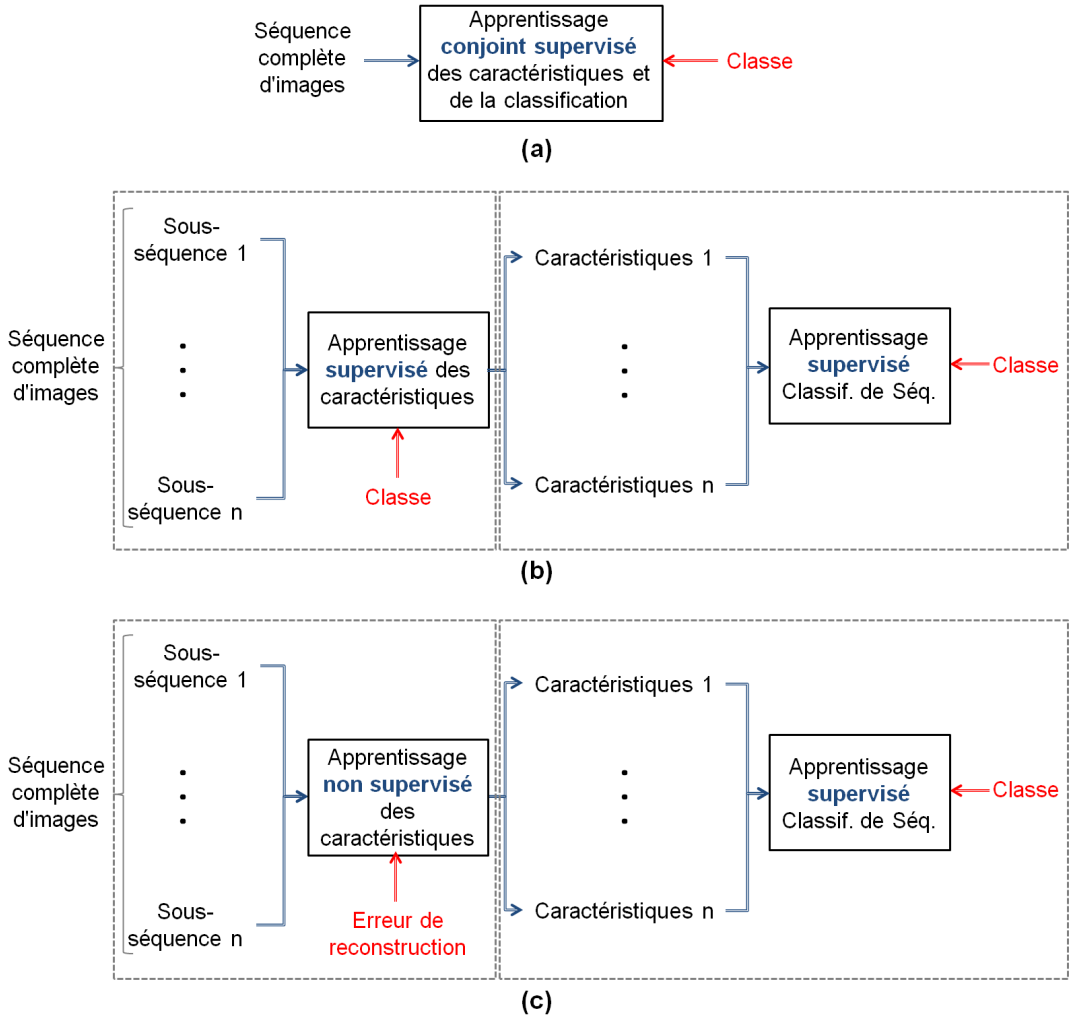

FIGURE 5.1 – Solutions envisagées pour le couplage entre l'apprentissage des caractéristiques et la classification : (a) - Couplage complet (b) - Apprentissage des deux étapes séparément de manière supervisée (c) - Apprentissage non supervisé des caractéristiques.

que l'erreur rétro-propagée diminue au fur et à mesure. Nous avons effectué une série d'expérimentations (que nous ne présenterons pas dans ce manuscrit) dans lesquelles nous avons tenté de coupler l'apprentissage neuronal des caractéristiques avec la phase de classification BLSTM. Nous n'avons néanmoins pas réussi à obtenir des résultats satisfaisants. Ceci est vraisemblablement dû à la complexité du modèle, et au fait que l'information liée à l'erreur globale n'est pas assez riche pour être rétro-propagée jusqu'aux premières couches.

Deuxième solution : Les étapes d'extraction des caractéristiques et de classification sont apprises séparément de manière supervisée (cf. Figure 5.1-(b)). C'est à cette solution que nous allons nous intéresser dans ce chapitre. Nous verrons que, dans

ce cas, le modèle *ConvNet* n'opère pas sur les séquences complètes, et que par conséquent l'extraction des caractéristiques se fait sur des segments de courte durée. L'intégration de ces différentes sous-séquences se fait donc lors de la classification, ce qui nécessite la mise au point d'une stratégie d'inter-connexion de ces deux étapes. De plus, puisque l'extraction des caractéristiques est apprise de manière supervisée, les dernières couches du modèle *ConvNet* consistent en un réseau MLP entraîné à produire certains vecteurs cibles (qui peuvent correspondre au label de la séquence, ou à d'autres informations). A noter que ce réseau MLP sert essentiellement à l'apprentissage des caractéristiques (c'est à dire qu'il n'intervient plus lors de la classification), mais qu'il peut aussi servir à évaluer le pouvoir discriminant intrinsèque du modèle *ConvNet* seul (plus de détails seront donnés ci-après).

Troisième solution : Les deux étapes sont entièrement découplées : Les caractéristiques sont apprises de manière non supervisée et le classifieur a posteriori, de manière supervisée (cf. Figure 5.1-(c)). L'information de classe n'intervient donc que lors de la classification. Cette troisième solution fera l'objet du chapitre 6.

Le reste de ce chapitre s'organisera comme suit : Nous allons commencer par introduire dans la section 5.2 les convolutions $3D$ qui représentent la caractéristique architecturale principale du modèle proposé. Nous allons ensuite décrire plus en détails dans la section 5.3 l'architecture du modèle *ConvNet* $3D$, puis présenter les points clés de l'apprentissage de ses paramètres. Nous allons enfin proposer dans la section 5.4 deux stratégies de classification (par vote et par BLSTM), qui seront associées au modèle *ConvNet* $3D$ afin de traiter les séquences complètes.

5.2 Convolutions $3D$

Dans le cas des modèles $2D$, les convolutions au niveau d'une couche donnée sont appliquées sur un voisinage local au niveau de la couche précédente. Un terme de biais est ensuite ajouté, et une fonction d'activation est appliquée à ce résultat, permettant ainsi de générer des cartes $2D$ de caractéristiques. La formulation mathématique qui permet de calculer ces cartes est la suivante : Si nous désignons par $a_{ij}^{x,y}$ la valeur du pixel (x, y) de la carte de caractéristiques j au niveau de la couche i, $a_{ij}^{x,y}$ aura pour expression :

$$a_{ij}^{x,y} = \sigma_{ij}\left(b_{ij} + \sum_{m \in v_{ij}} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} w_{ijm}^{p,q}\, a_{(i-1)m}^{x+p,y+q}\right) \tag{5.1}$$

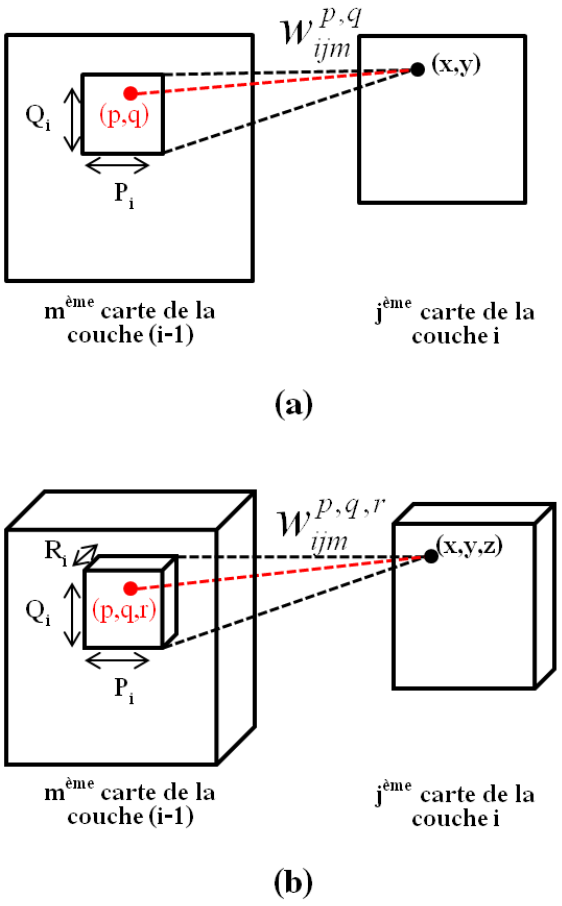

FIGURE 5.2 – Principe des convolutions pour les modèles *ConvNets* : (a) - Cas 2*D* (b) - Cas 3*D*.

où σ_{ij} et b_{ij} désignent respectivement la fonction d'activation et le terme de biais de la $j^{\text{ème}}$ unité de la couche i, v_{ij} est le nombre d'unités de la couche $(i-1)$ qui sont connectés à l'unité j, P_i et Q_i désignent la taille du voisinage spatial sur lequel est calculé la convolution, et $w_{ijm}^{p,q}$ est le paramètre de la convolution associé à la position (p,q) et à l'unité m de la couche précédente (la connectivité entre les couches successives étant fixe). La Figure 5.2-(a) illustre cette formulation en reprenant les notations utilisées dans l'équation 5.1.

Pour les modèles *ConvNets* 2*D*, si les noyaux des convolutions (et les autres paramètres du modèle) sont appris avec un critère bien déterminé (cf. sous-section 2.3.3 du chapitre 2), celles-ci permettent d'extraire, couche par couche, les motifs 2*D* caractérisant l'image au sens de la classification attendue.

Nous proposons d'étendre ce principe au cas de la vidéo. Pour ce faire, nous avons étudié plusieurs solutions : (i) Une extension directe des convolutions 2*D* au cas 3*D*, (ii)

l'utilisation de convolutions 2D spatiales suivies de convolutions 1D temporelles, et (iii) l'utilisation de convolutions 2D uniquement et l'intégration de l'aspect temporel lors de la classification. Nous avons retenu la première solution (vu que la deuxième ne modélise pas certains cas -comme par exemple une simple translation d'un bloc de l'image à une autre image de la vidéo-, et que la troisième capture uniquement des motifs spatiaux). La solution retenue se base sur des convolutions à noyau tridimensionnel, qui sont appliquées sur des volumes 3D (qui correspondent à un regroupement d'images 2D successives, voir Figure 5.2-(b)).

En reprenant les notations de l'équation 5.1, la $j^{\text{ème}}$ carte de caractéristiques 3D située sur la couche i s'exprime pour chaque position (x, y, z) par :

$$a_{ij}^{x,y,z} = \sigma_{ij}\left(b_{ij} + \sum_{m \in v_{ij}} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} w_{ijm}^{p,q,r}\, a_{(i-1)m}^{x+p,y+q,z+r}\right) \tag{5.2}$$

où R_i est la longueur temporelle du noyau 3D de la convolution, $w_{ijm}^{p,q,r}$ est le paramètre du noyau à la position (p, q, r) (cf. Figure 5.2-(b)).

A noter que le principe de *partage des poids* (cf. sous-section 2.3.3 du chapitre 2) dans le cas 3D est identique à celui du cas 2D. Concrètement, les paramètres du noyau de convolution 3D sont les mêmes pour toutes les positions du volume spatio-temporel sur lequel la convolution est appliquée.

Dans la section suivante, nous allons présenter le modèle ConvNet 3D proposé reposant sur les convolutions 3D que nous avons décrites dans de cette section.

5.3 Modèle *ConvNet* 3D proposé

Le modèle *ConvNet* 3D que nous proposons se base sur les convolutions décrites dans la section précédente pour apprendre automatiquement et de manière supervisée des caractéristiques spatio-temporelles pour la classification vidéo. Nous allons décrire dans un premier temps son architecture, puis nous nous intéresserons à l'apprentissage de ses paramètres.

5.3.1 Architecture du réseau

L'architecture proposée est illustrée sur la Figure 5.3 sur un exemple de la base d'actions humaines KTH [SLC04]. Cet exemple comporte 10 couches : Une couche d'entrée, une couche de sortie et 8 couches cachées.

Le réseau prend en entrée un volume spatio-temporel de taille $M \times N \times T$, c'est à

dire une suite de T images successives de tailles $M \times N$ pixels chacune. Le choix de T est alors important, étant donné que :

- Une valeur trop faible de T conduit à des segments qui ne contiennent pas assez d'informations spatio-temporelles discriminantes.
- Une valeur trop importante de T augmente considérablement la complexité du modèle.

En particulier, les deux cas "extrêmes" correspondent à :

- $T = 1$, c'est à dire que la vidéo est modélisée "image par image" (comme pour le cas des *ConvNets* $2D$), et l'aspect temporel est pris en compte uniquement lors de la classification.
- La valeur de T correspond à la longueur de la vidéo la plus courte de la base d'apprentissage.

Ces deux cas de figures ne sont évidemment pas ceux pour lesquels nous avons opté, vu que le premier génère des caractéristiques spatiales, et que le deuxième correspond à un nombre de paramètres très élevés, et à un modèle qui ne peut pas être appliqué à des séquences dont la longueur est inférieure à T.

Nous proposons donc d'entraîner le modèle *ConvNet* $3D$ à extraire des caractéristiques sur des sous-séquences obtenues en découpant les séquences vidéo entières en segments successifs de longueur T chacun. L'intégration de ces différents segments pour la labellisation de la séquence nécessite donc la mise en place de stratégies de classification des différentes sorties possibles du *ConvNet* $3D$ correspondant aux différents segments de longueur T de la séquence complète (cf. section 5.4).

La couche de sortie contient un ensemble de neurones (un par sortie désirée), et les couches cachées se répartissent en quatre catégories :

Des couches de convolution : Qui regroupent des cartes de caractéristiques $3D$ calculés en utilisant des convolutions $3D$ comme expliqué dans la section 5.2. Ces cartes de caractéristiques sont notées C_i sur la Figure 5.3. Les noyaux des convolutions C_i ont pour tailles $P_i \times Q_i \times R_i$, qui ont généralement des valeurs impaires afin que le voisinage pris en compte dans le calcul de la convolution soit centré sur un pixel donné. Ainsi, si nous considérons l'exemple de la couche C_1, les cartes de caractéristiques $3D$ obtenues auront pour taille $(M - P_1 + 1) \times (N - Q_1 + 1) \times (T - R_1 + 1)$ chacune. Le même principe est aussi appliqué pour le cas de C_2 et de C_3.

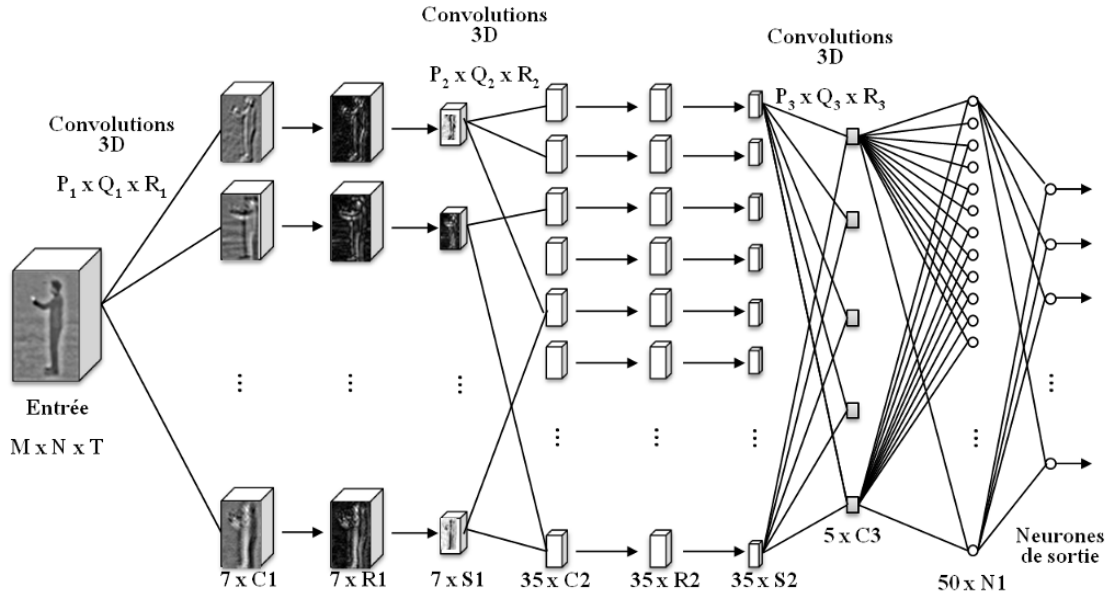

FIGURE 5.3 – Exemple d'architecture du réseau *ConvNet 3D* proposé. Illustration sur un exemple de la base KTH d'actions humaines [SLC04].

Des couches de sous-échantillonnage : Qui consistent en un ensemble de modules (notés S_i sur la Figure 5.3) qui appliquent chacun un moyennage local spatio-temporel sur leurs entrées, afin d'augmenter la robustesse aux faibles translations. Dans le cas du modèle que nous proposons, les modules S_i effectuent un sous-échantillonnage de facteur 2. A noter qu'il existe d'autres alternatives à ces modules de sous-échantillonnage (que nous n'avons pas testé), comme par exemple les modules de *Max Pooling* (que nous avons présentés dans la sous-section 2.3.2 du chapitre 2) qui caractérisent un voisinage local par sa valeur maximale, au lieu de sa moyenne.

Des couches de rectification : Qui sont notées R_i sur la Figure 5.3. Elles consistent en un ensemble de modules qui appliquent une simple valeur absolue sur leurs entrées. Elles ont été introduites par Jarret et al. dans [JKRL09] pour les *ConvNets* 2*D*. Les auteurs ont démontré que l'utilisation de ces modules entre les couches de convolution et de sous-échantillonnage améliorait considérablement les performances en terme de classification dans le cadre d'une application de reconnaissance d'objets [JKRL09]. Ceci est dû au fait qu'elles permettent de ne pas prendre en compte la "polarité" des changements de contraste (du clair vers le sombre et inversement), ce qui permet de réduire l'influence de la couleur des objets et des changements d'illumination sur le résultat de la classification.

Une couche de neurones : La dernière couche cachée avant la couche de sortie regroupe des neurones classiques. Les modules de cette couche sont notés N_1 sur la Figure 5.3.

Sur l'exemple de la Figure 5.3, la première couche cachée contient 7 modules C_1 entièrement connectés à la couche d'entrée. Chacun de ses modules est ensuite connecté à un module de rectification R_1, puis à un module de sous-échantillonnage S_1. Les 7 modules S_1 sont connectés à la couche suivante selon le schéma de connexion introduit par Garcia et Delakis dans [GD04]. Concrètement, chacun des 7 modules S_1 est d'abord connecté à deux modules de la couche suivante (donnant ainsi 14 modules C_2). Ensuite, chacune des 21 paires possibles de modules $S1$ est aussi connectée à un module C_2. Ceci conduit à un total de 35 modules C_2 au niveau de la quatrième couche cachée. Ce schéma permet de faire émerger des filtres complexes en combinant notamment des filtres appris dans les couches précédentes.

Vient ensuite une succession de couches R_2 et S_2 qui suit le même principe que celui utilisé pour R_1 et S_1. Les sorties de S_2 sont ensuite totalement connectées à cinq modules de convolution C_3. Enfin, la couche de neurones N_1 et la couche de sortie forment un MLP classique totalement connecté dont la couche d'entrée est la sortie de C_3.

Ainsi, ce modèle peut être vu comme une combinaison de deux parties qui ont des rôles distincts :

- Une première partie, qui regroupe les couches cachées allant de C_1 à C_3, et dont le rôle est de construire de couche en couche une représentation haut niveau des données d'entrée, qui capturent les informations spatio-temporelles saillantes et qui les encodent au niveau de la sortie des modules C_3 (qui ont une taille largement inférieure à celle de l'entrée).

- Une seconde partie réalisant la classification, qui se base sur les sorties de ces modules C_3 afin d'attribuer un label au niveau de la dernière couche, et qui fonctionne comme un MLP classique.

Les sorties de C_3 peuvent donc être considérées comme des descripteurs relatifs à des caractéristiques qui capturent l'information spatio-temporelle saillante de l'entrée, et qui sont utiles à la classification. De cette manière, au delà de la classification effectuée par le MLP qui est apprise lors de l'apprentissage supervisé décrit dans la section suivante, nous proposerons d'utiliser ces descripteurs (sorties de C_3) comme entrée du modèle de classification BLSTM. Ces deux pistes de classification de séquences seront décrites dans la section 5.4.

5.3.2 Apprentissage

L'apprentissage du modèle *ConvNet 3D* est effectué en prenant en entrée des sous-séquences de longueur T (comme expliqué précédemment), et en ciblant, pour chaque sous-séquence, un certain vecteur désiré. Pour ce dernier, nous avons opté pour celui correspondant au label de la séquence complète. A noter cependant que plusieurs autres cibles potentielles ont été envisagées et expérimentées, sans pour autant donner un résultat aussi satisfaisant. Ces cibles peuvent correspondre à des "sous-classes" qui décrivent de manière plus fine le contenu du segment mis en entrée.

A titre d'exemple, pour le cas de la reconnaissance d'actions humaines, ces sous-classes correspondent aux différentes étapes composant l'action (comme par exemple la succession des mouvements élémentaires des jambes pour l'action "marcher"). L'idée en ciblant ces sous-classes est d'entraîner le modèle à produire des caractéristiques qui soient plus en adéquation avec le contenu de l'entrée, et de lever certaines ambigüités dues à la présence de sous-classes communes pour des classes différentes.

La principale difficulté est alors de définir et de localiser temporellement ces sous-classes. Nous avons envisagé pour ce faire trois solutions possibles :

Une annotation manuelle : Ceci nécessite d'examiner chaque séquence vidéo image par image, et d'attribuer manuellement à chaque instant un label correspondant à la sous-classe. Cette première solution reste cependant fastidieuse et difficile à mettre en place vu le grand nombre des données d'apprentissage, et n'a donc pas été expérimentée.

Un *clustering* non supervisé : Cette deuxième solution consiste à appliquer un *clustering* non supervisé (par exemple un algorithme des k-moyennes) sur des descripteurs de caractéristiques manuelles calculées sur les données d'apprentissage, afin de regrouper les instants correspondant à des images similaires, et d'extraire ainsi les sous-classes. Nous avons menés des expérimentations avec les moments de Zernike [Tea80] calculés sur chaque image, sans pour autant obtenir des résultats satisfaisants, dans le sens où les centres des *clusters* ne correspondent pas à des sous-classes visuellement pertinentes.

Les "états" décodés par un modèle HCRF : Cette solution itérative consiste à entraîner, dans un premier temps, le modèle *ConvNet 3D* à cibler les classes correspondant aux séquences complètes, puis, une fois l'apprentissage terminé, d'entraîner un classifieur HCRF avec les caractéristiques ainsi obtenues. Les états cachés décodés par le modèle HCRF sont ensuite récupérés pour chaque sous-séquence et utili-

sés comme sous-classes cibles pour ré-entraîner un modèle ConvNet 3D. L'idée est de profiter de la modélisation faite par le HCRF de la séquence complète en suite d'états, et de les exploiter afin de produire, de manière itérative, des caractéristiques plus représentatives des données. Néanmoins, dans la pratique, nous avons observé que les états cachés décodés ne sont pas exploitables dans ce sens, puisqu'ils ne correspondent pas à des sous-classes visuellement pertinentes.

Nous avons donc choisi d'exploiter le schéma d'apprentissage supervisé ciblant la classe correspondant à la séquence complète. Le modèle est entraîné avec une extension au cas 3D de l'algorithme de rétro-propagation du gradient avec *momentum* proposé dans [LBBH98]. Pour rappel, cet algorithme reprend la même formulation que celle de l'apprentissage des MLPs, en prenant compte des modules de convolution, de rectification et de sous-échantillonnage introduits par les *ConvNets*.

Concrètement, pour le cas d'une couche de convolutions 2D, si nous reprenons les notations de l'équation 5.1, le terme de mise-à-jour du poids $w_{ijm}^{x,y}$ à l'itération n (analogue à celui exprimé par l'équation 2.18 du chapitre 2 pour le cas des MLPs) est obtenu par :

$$\Delta w_{ijm}^{x,y}(n) = -\epsilon \cdot \sum_{m \in v_{ij}} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \delta_{ij}^{p,q} \, a_{(i-1)m}^{x+p,y+q} + \alpha \cdot \Delta w_{ijm}^{x,y}(n-1) \tag{5.3}$$

où $\delta_{ij}^{p,q}$ est le gradient local (bidimensionnel) caractérisant le $j^{\text{ème}}$ module de convolution de la couche i. Ce gradient local s'exprime en fonction des gradients locaux de la couche suivante $(i+1)$, et a une expression différente selon le type de modules présents dans cette dernière (convolutions, sous-échantillonnage, rectification ou neurones -cf. [LBBH98] pour plus de détails-).

L'extension de l'équation 5.3 au cas 3D est directe, et se fait simplement en remplaçant dans celle-ci les signaux bidimensionnels par leurs équivalents tridimensionnels. En reprenant les notations de l'équation 5.2, nous obtenons ainsi :

$$\Delta w_{ijm}^{x,y,z}(n) = -\epsilon \cdot \sum_{m \in v_{ij}} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} \delta_{ij}^{p,q,r} \, a_{(i-1)m}^{x+p,y+q,z+r} + \alpha \cdot \Delta w_{ijm}^{x,y,z}(n-1) \tag{5.4}$$

A noter que la mise-à-jour des poids des couches de sous-échantillonnage 3D (que nous ne présentons pas ici) est aussi une extension directe du cas 2D.

Une fois l'apprentissage terminé, et vu que le processus de génération des caractéristiques spatio-temporelles est entièrement automatique, il est intéressant de vérifier si ces caractéristiques apprises sont visuellement pertinentes. Nous présentons sur la Figure 5.4 quatre exemples de cartes de caractéristiques C_1 (sur les sept existantes) qui sont

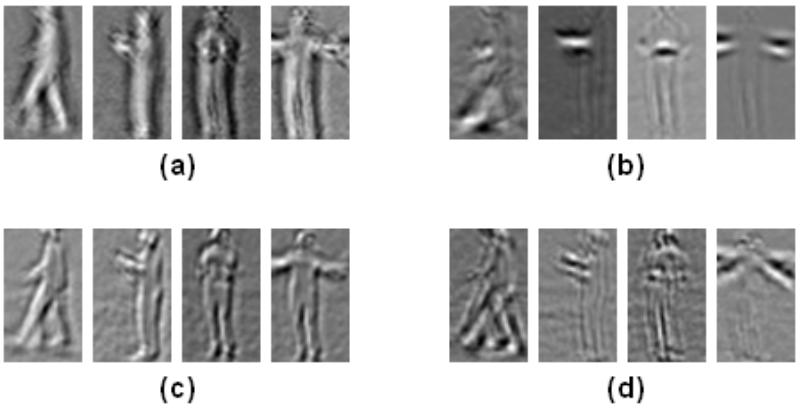

FIGURE 5.4 – Quatre exemples de cartes de caractéristiques C_1 apprises automatiquement par le modèle *ConvNet 3D* sur la base d'actions humaines KTH [SLC04]. Ces cartes semblent encoder : (a) - La silhouette du personnage (b) Les membres utilisés lors de l'action (c) - Les contours (d) - L'historique du mouvement.

obtenues pour le cas de la base d'actions humaines KTH [SLC04]. Même s'il est difficile (et pas forcément nécessaire) d'interpréter précisément les informations capturées par ces cartes et de trouver un lien avec des caractéristiques dites manuelles, elles semblent visuellement pertinentes. En effet, les Figures 5.4-(a), (b), (c) et (d) semblent encoder respectivement la silhouette du personnage, les membres utilisés lors de l'action, les contours et l'historique du mouvement.

Dans la section suivante, nous allons présenter les stratégies adoptées pour classer les séquences vidéo complètes à partir des sorties ou des descripteurs de caractéristiques apprises par le modèle *ConvNet 3D*.

5.4 Stratégies de classification des séquences vidéo complètes

Comme nous l'avons mentionné précédemment, une fois l'apprentissage terminé, le modèle *ConvNet 3D* extrait pour une séquence de test donnée un vecteur de sortie ou de description pour chaque sous-séquence. La classification d'une séquence complète nécessite alors l'intégration des sorties ou des descripteurs des différentes sous-séquences. Deux stratégies ont ainsi été mises au point, à savoir la classification par vote majoritaire, et la classification BLSTM.

5.4.1 Classification par vote

La première stratégie, illustrée sur la Figure 5.5-(a), se base sur un simple système de vote sur les décisions individuelles relatives à chaque sous-séquence. Typiquement, les

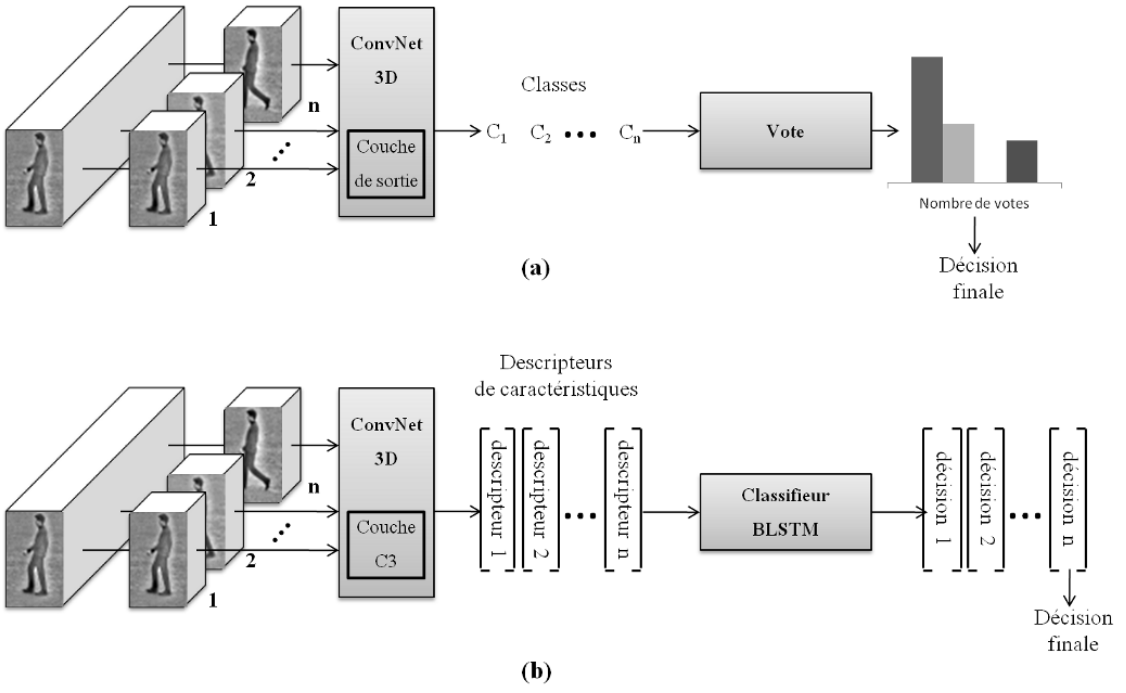

FIGURE 5.5 – Stratégies de classification des séquences vidéo entières : (a) - Classification par vote (b) - Classification BLSTM.

sorties du réseau *ConvNet 3D* sont récupérées pour chaque segment de longueur T, obtenant ainsi une séquence de labels (un label par segment). Un vote permet ensuite d'attribuer à la séquence complète la classe la plus représentée.

Cette stratégie de classification, bien que très simple, présente l'avantage de décrire globalement le contenu des séquences, sans modéliser les relations entre les segments. Or, souvent, une description globale d'un contenu est suffisante pour sa classification, les relations entre les parties n'étant pas toujours nécessaires. Les sacs de mots [SM86] sont d'ailleurs un exemple d'approches très populaires qui se basent aussi sur ce principe.

Nous verrons dans le chapitre 7 que cette stratégie permet d'obtenir des résultats relativement bons. Néanmoins, son principal inconvénient réside dans le fait que l'évolution temporelle n'est pas prise en compte lors de la classification. Cette stratégie nous servira donc uniquement pour évaluer le pouvoir discriminant intrinsèque du modèle *ConvNet 3D*.

A noter que nous avons aussi testé une autre approche dans laquelle nous moyennons les sorties individuelles correspondant à chacune des sous-séquences, et attribuons à la séquence complète le label correspondant à la sortie moyenne la plus élevée. Cette approche a donné des résultats (que nous ne présenterons pas dans le chapitre 7) équivalents ou inférieurs au vote majoritaire.

5.4.2 Classification BLSTM

Nous avons vu dans la section 5.3 que les sorties de la couche C_3 pouvaient être vues comme des descripteurs qui encodent le contenu spatio-temporel saillant des sous-séquences d'entrée.

Nous proposons donc une deuxième stratégie, illustrée sur la Figure 5.5-(b), qui consiste à utiliser ces descripteurs afin d'entraîner un modèle de classification de séquences BLSTM visant à classer la séquence complète. Pour ce faire, nous associons à chaque séquence complète une séquence de descripteurs générée en collectant les réponses de la couche C_3 pour les différents segments de la séquence.

Ces séquences de descripteurs sont ensuite présentées en entrée d'un modèle de classification BLSTM, qui prend en compte l'évolution temporelle des caractéristiques apprises. Le réseau BLSTM est entraîné de manière supervisée en visant la classe de la séquence à chaque instant. Lors de la phase de test, ce réseau rendra une suite de décisions individuelles (une décision à chaque instant) qui permettra d'attribuer le label final à la séquence complète. Nous verrons dans le chapitre 7 que cette deuxième stratégie tire profit du fort pouvoir discriminant des modèles BLSTM pour les données séquentielles, que nous avons évalué lors de l'étude comparative présentée dans le chapitre 4.

5.5 Conclusion

Nous avons proposé dans ce chapitre un modèle neuronal d'apprentissage supervisé de caractéristiques spatio-temporelles pour la classification de séquences vidéo. Ce modèle se distingue de la méthodologie dominante par le fait qu'il ne repose sur aucune connaissance a priori, et est obtenu par apprentissage automatique à partir d'exemples. Nous nous sommes pour cela inspirés des modèles dits profonds, c'est à dire qui apprennent plusieurs niveaux de représentation des données, correspondant à une hiérarchie de caractéristiques allant des données brutes à une représentation de haut niveau.

Concrètement, nous avons proposé un modèle *ConvNet* 3*D* qui permet d'étendre les modèles *ConvNet* 2*D* classiques au cas de la vidéo, en proposant une architecture multi-couches qui opère sur des volumes spatio-temporels obtenus en regroupant plusieurs images successives de la vidéo. L'architecture proposée se distingue des quelques travaux qui ont étudié l'extension des modèles *ConvNets* 2*D* au cas de la vidéo (par exemple [KLY07, JXYY10]) par le fait qu'elle opère sur les données brutes, sans aucun pré-traitement complexe. Ce modèle apprend de façon supervisée à partir d'une sous-séquence de courte durée à affecter le label de la séquence complète.

Nous avons aussi présenté deux stratégies de classification des séquences vidéo complètes : (i) La première se base sur un système de vote majoritaire et ne tient pas compte de l'aspect temporel des données, et (ii) la deuxième utilise les caractéristiques apprises (les sorties d'une couche intermédiaire du modèle *ConvNet* 3*D*) pour entraîner un modèle neuronal BLSTM de classification de séquences. Nous présenterons et comparerons au cours du chapitre 7 les résultats expérimentaux relatifs à ces deux stratégies.

Le chapitre suivant présentera un autre modèle d'apprentissage de caractéristiques spatio-temporelles, qui, à la différence du modèle *ConvNet* 3*D* proposé dans ce chapitre, est entraîné de manière non supervisée.

Chapitre 6

Apprentissage non supervisé de caractéristiques parcimonieuses

Sommaire

6.1 Introduction

Nous avons présenté lors du chapitre précédent le modèle *ConvNet 3D* qui permet d'apprendre, de manière totalement automatique et sans aucune connaissance a priori, des caractéristiques spatio-temporelles pour la classification vidéo. Vu que l'apprentissage des paramètres de ce modèle se fait de manière supervisée, le label correspondant à la séquence est pris en compte lors de la génération des caractéristiques pour chaque sous-séquence. Le *ConvNet 3D* aura donc tendance à produire des descripteurs similaires (pré-classés) pour des sous-séquences d'une même classe, réduisant l'information de variation au cours du temps au sein d'une même séquence. Ce processus pourrait nuire à l'étape de classification de séquences puisqu'à l'extrême (ce qui est bien entendu faux en réalité), toutes les sous-séquences d'une même vidéo pourraient produire les mêmes descripteurs rendant inutile l'utilisation d'un classifieur de séquences.

Inversement, si nous considérons par exemple le cas de la reconnaissance d'actions, il peut arriver que certaines actions aient des portions identiques, ou dont le contenu spatio-temporel varie très peu. Nous pouvons citer dans ce sens l'exemple des actions de la base KTH [SLC04] (que nous présenterons lors du chapitre 7), qui contiennent toutes des portions sur lesquelles le personnage n'effectue pas de mouvements particuliers. Dans ce cas de figure, lors de l'apprentissage supervisé des caractéristiques, le modèle attribue des labels cibles différents pour des sous-séquences quasi-identiques, donnant des objectifs antagonistes qui peuvent nuire à l'apprentissage des *ConvNets 3D*, mais aussi à la classification de séquences.

Il peut alors être intéressant d'envisager un apprentissage de caractéristiques qui ne repose pas sur une classification des sous-séquences, afin notamment d'avoir des caractéristiques qui représentent intrinsèquement le contenu spatio-temporel des vidéos, sans aucune prise en compte des classes. La classification, à proprement parler, est donc entièrement reportée sur l'étape de classification de séquences.

Nous allons proposer dans ce chapitre un modèle d'apprentissage non supervisé des caractéristiques décrivant le contenu d'un segment vidéo. Nous allons pour ce faire nous baser sur un schéma d'auto-encodage des données, qui est illustré sur la Figure 6.1. Comme nous l'avons évoqué lors de la sous-section 2.3.1 du chapitre 2, ce schéma permet d'extraire des caractéristiques puisqu'il arrive à reproduire les entrées à partir des coordonnées de leur projection dans un espace de représentation, et qu'il capture donc des motifs "saillants" nécessaires à cette reconstruction. A noter que le schéma de la Figure 6.1 est très général, et recouvre aussi bien des méthodes comme les auto-encodeurs neuronaux [RHW86], que les codeurs parcimonieux [OF97, LS99, AEB05, RPCL06, RHBL07].

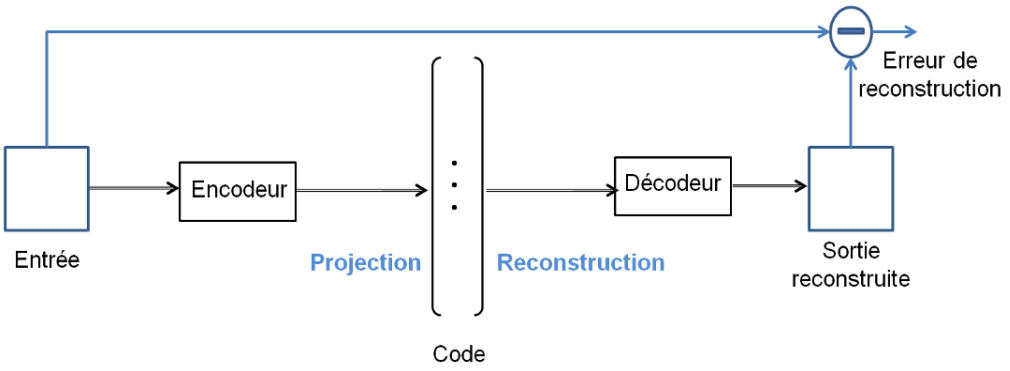

FIGURE 6.1 – Schéma général d'un modèle d'auto-encodage des données.

Il se compose d'un encodeur qui est entraîné à projeter une entrée donnée dans un espace de caractéristiques, et d'un décodeur qui reconstruit cette entrée à partir de ses coordonnées dans cet espace, qui sont appelées "code" (cf. Figure 6.1). L'encodeur et le décodeur sont des fonctions paramétriques qui sont entraînées à minimiser une fonction objectif bien déterminée.

Cette procédure a généralement pour objectif de produire une représentation compacte des données, dans le sens où la dimension du code est inférieure à celle des entrées. En effet, cette contrainte est nécessaire afin maîtriser la complexité de la représentation.

Au cours des dernières années, certains travaux dans le domaine de l'apprentissage non supervisé [OF97, TWOH03, RPCL06, RHBL07, MBPS09] ont préconisé de relâcher cette contrainte, en proposant une représentation dite "parcimonieuse sur-complète", c'est à dire dont la dimension du code est supérieure ou égale à celle des entrées, mais où seul un faible nombre de composants du code sont non nuls. Le fait que la représentation soit sur-complète permet de générer un dictionnaire très large, et d'effectuer par la suite la classification dans un espace de grande dimension, tandis que la parcimonie contraint la taille "réelle" du code utilisé pour représenter une entrée donnée. Ces travaux sur la représentation sur-complète parcimonieuse des données ont donné de très bons résultats dans les domaines du traitement audio et d'images, mais l'extension au cas de la vidéo est un domaine encore ouvert. Les quelques tentatives que nous pouvons citer (par exemple les travaux de Mei et Ling [ML11] et ceux de Lu et al. [LYYL12] sur le suivi dans les vidéos) se basent sur une représentation de l'information $2D$, et traitent la vidéo "image par image".

Par ailleurs, plusieurs stratégies différentes ont été proposées dans ces travaux afin de générer des codes parcimonieux, reposant la plupart du temps sur des algorithmes d'optimisation coûteux et globaux sur la base d'apprentissage ("hors ligne" ou *batch* en

anglais). Ces dernières années, plusieurs travaux se sont intéressés à la réduction de la complexité de ces algorithmes [RPCL06, RHBL07, MBPS09] et notamment à la définition d'algorithmes "en ligne" pour pouvoir traiter de très grandes bases de données de patchs issus d'images par exemple.

Dans ce manuscrit, nous nous intéresserons tout particulièrement à la méthode introduite par Ranzato et al. [RPCL06, RHBL07] dans le domaine de la reconnaissance d'images fixes, et qui repose sur des noyaux de convolution et sur une fonction non linéaire appelée "fonction de parcimonie". Ranzato et al. ont présenté un algorithme d'apprentissage de ce modèle, ainsi qu'une procédure spécifique afin de gérer l'invariance du code appris à la translation spatiale.

Dans ce chapitre, nous allons proposer un modèle d'apprentissage non supervisé de caractéristiques parcimonieuses à partir de séquences vidéo, qui reprend certains principes présentés par Ranzato et al. [RPCL06, RHBL07] pour le cas des images fixes (notamment la fonction de parcimonie), tout en introduisant un certain nombre de nouveautés. D'une part, notre modèle opère sur des patchs spatio-temporels ($3D$), et génère un code parcimonieux représentatif du contenu de ces patchs. D'autre part, nous présenterons une approche différente afin de gérer l'invariance de la représentation aux translations spatiales et temporelles.

Le reste de ce chapitre s'organisera comme suit : Le modèle proposé sera d'abord présenté dans la section 6.2. L'algorithme d'apprentissage et la fonction objectif qui y sont associés seront ensuite introduits dans la section 6.3. Les détails architecturaux relatifs à l'encodeur et au décodeur seront ensuite décrits dans la section 6.4. Enfin, la classification des séquences vidéo complètes en se basant sur la représentation parcimonieuse apprise par le modèle sera enfin abordée lors de la section 6.5.

6.2 Modèle proposé pour l'apprentissage non supervisé des caractéristiques

L'approche que nous proposons dans ce chapitre est hiérarchique : La séquence vidéo est d'abord décomposée en blocs spatio-temporels de courte durée (une suite de T images consécutives), qui sont eux-mêmes décomposés en patchs spatio-temporels de taille $M \times M \times T$ chacun (cf. Figure 6.2). Ces derniers sont le niveau de représentation sur lequel l'apprentissage des caractéristiques est effectué. Ce choix est justifié par le fait que les motifs spatio-temporels sont moins variables localement que sur les images complètes, ce qui permet de réduire la diversité du contenu à encoder.

Le modèle proposé est illustré sur la Figure 6.3, et contient deux modules principaux :

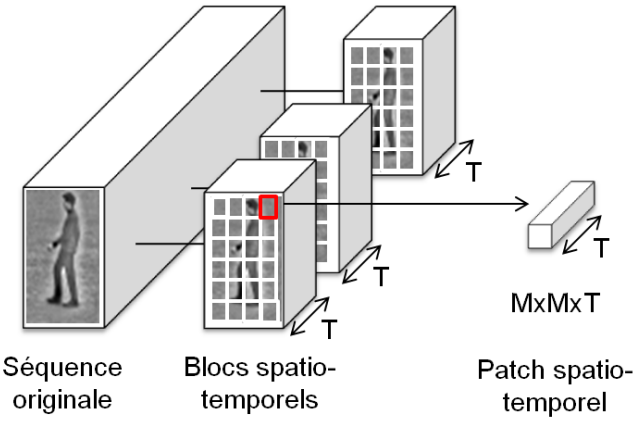

FIGURE 6.2 – Décomposition de la séquence vidéo en blocs spatio-temporels, qui sont eux-mêmes décomposés en patchs. L'apprentissage des caractéristiques est effectué au niveau des patchs.

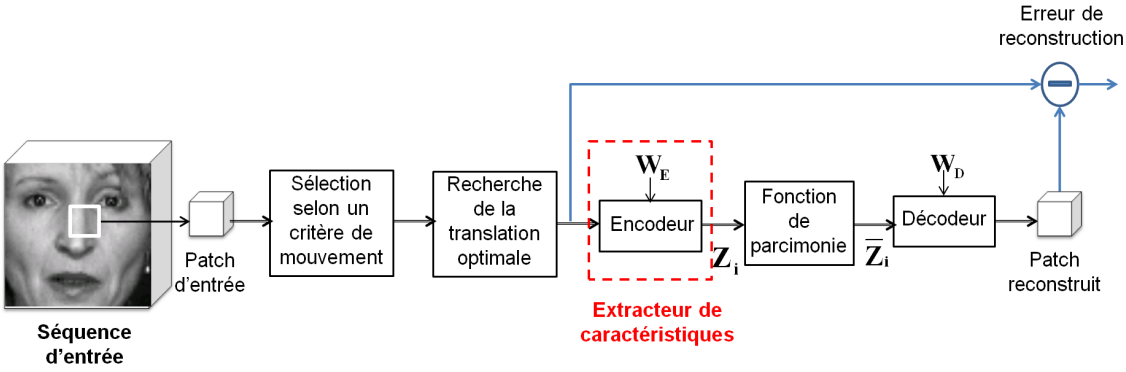

FIGURE 6.3 – Schéma global du modèle proposé.

- Un encodeur qui construit, directement à partir du patch spatio-temporel d'entrée, un code non parcimonieux Z_i qui représente son contenu, et dont les paramètres sont notés W_E. Cet encodeur joue le rôle d'extracteur de caractéristiques.

- Un décodeur (dont les paramètres sont notés W_D) qui apprend à reconstruire le patch d'entrée à partir d'une version parcimonieuse $\overline{Z}_i$ du code.

Le modèle présenté ci-après ne dépend pas du type d'encodeur et de décodeur utilisés, et peut donc être appliqué pour différentes architectures (celle qui a été retenue sera décrite dans la section 6.4). Hormis l'encodeur et le décodeur, le modèle proposé contient également d'autres modules (cf. Figure 6.3) que nous allons détailler un à un dans ce qui suit.

Fonction de parcimonie

Comme nous l'avons évoqué lors de l'introduction, de nombreuses approches ont été introduites dans la littérature afin de générer des codes parcimonieux. Une première catégorie [OF97, TWOH03] se base sur l'ajout d'un terme de parcimonie dans la fonction objectif du modèle, qui pénalise les unités du code qui sont activées, réduisant ainsi leur nombre.

Une seconde catégorie se base quant à elle sur l'ajout d'une fonction de parcimonie placée entre l'encodeur et le décodeur. Ce principe a été introduit par P. Foldiak dans [Föl90] et popularisé plus récemment par les travaux de Ranzato et al. [RPCL06, RHBL07], qui ont défini une fonction de parcimonie simple à paramétrer (deux paramètres uniquement), et adaptée à l'apprentissage "en ligne". C'est cette dernière que ne avons utilisé dans notre modèle. C'est une fonction non linéaire qui peut être vue comme une fonction *SoftMax* [Bri90] appliquée sur les échantillons successifs de chaque élément formant le code. Étant donnés le $i^{ème}$ exemple d'apprentissage, ainsi que le code $Z_i = \left\{z_i^{(k)}\right\}_{k\in[1..N]}$ correspondant (où N est la taille du code), le code parcimonieux $\overline{Z}_i = \left\{\overline{z}_i^{(k)}\right\}_{k\in[1..N]}$ est exprimé par :

$$\overline{z}_i^{(k)} = \frac{\eta e^{\beta z_i^{(k)}}}{\zeta_i^{(k)}} \tag{6.1}$$

où :

$$\zeta_i^{(k)} = \eta e^{\beta z_i^{(k)}} + (1-\eta)\,\zeta_{i-1}^{(k)} \tag{6.2}$$

Ici, η et β sont deux paramètres positifs qui contrôlent respectivement le degré de parcimonie et de lissage du code : Plus la valeur de η est faible, plus il y a d'échantillons utilisés pour calculer ζ dans l'équation 6.2. Pour β, plus sa valeur est élevée, plus le code obtenu est quasi-binaire.

Invariance à la translation

Afin d'assurer l'invariance aux translations dans les domaines spatial et temporel des représentations apprises (c'est à dire que le modèle attribue le même code aux versions translatées spatialement et temporellement d'une entrée donnée), nous proposons d'introduire un paramètre supplémentaire t_i (un vecteur de translation tri-dimensionnelle), sur lequel l'optimisation est effectuée.

Étant donné un patch spatio-temporel X_i (où $i \in [1..P]$, et P est le nombre d'exemples de voisinages spatio-temporels de l'apprentissage), l'idée est de représenter le voisinage spatio-temporel de X_i par un unique patch $\phi\,(X_i, t_i)$ sélectionné dans le voisinage de X_i

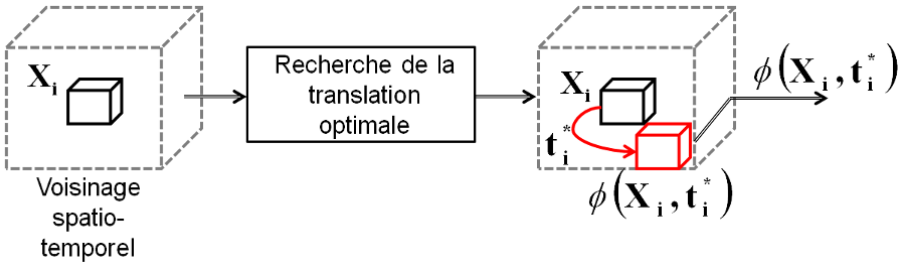

FIGURE 6.4 – Principe de la recherche de la translation optimale : Tous les patchs situés dans un certain voisinage spatio-temporel de X_i seront représentés par un seul patch translaté $\phi(X_i, t_i^*)$.

selon la translation t_i. Ce patch translaté est celui qui minimise une fonction objectif, qui sera présentée dans la sous-section 6.3.1, pour un jeu de paramètres (W_E, W_D) donné. Ce principe est illustré sur la Figure 6.4.

En pratique, cette procédure n'est entamée qu'après la fin de la première itération de l'apprentissage, afin d'assurer un minimum de pertinence des paramètres (W_E, W_D) courants lors de l'optimisation sur t_i.

A noter que cette approche que nous avons introduite pour gérer l'invariance aux translations est différente de celle présentée par Ranzato et al. [RHBL07]. Pour gérer cette invariance, leur méthode reposait sur deux idées clés :

- La première est de calculer la sortie de l'encodeur pour chaque patch du voisinage considéré afin de générer une carte de caractéristiques, puis d'appliquer pour chaque élément du code une unité de *Max Pooling* pour sélectionner la réponse maximale de chaque carte de caractéristiques. Seules les valeurs de ces réponses maximales sont utilisées en entrée de la fonction de parcimonie.

- La deuxième consiste à sauvegarder les localisations spatiales de ces réponses maximales détectées (les auteurs appellent ces localisations "paramètres de transformation"), et de les transmettre au décodeur pour la reconstruction. Ce dernier régénère des cartes de caractéristiques en plaçant, à chacune des localisations sauvegardées, la valeur du code parcimonieux correspondant. Ces différentes cartes sont ensuite combinées pour reconstruire l'entrée à l'aide du décodeur.

De cette manière, le code parcimonieux encodera uniquement la présence ou non d'un motif donné (caractérisé par un noyau de convolution), en faisant abstraction de sa localisation (qui est utilisée seulement pour la reconstruction). Ceci permet d'obtenir, comme dans le cas de l'approche que nous proposons, des codes robustes aux translations, vue que la réponse du *Max Pooling* sera la même pour deux versions translatées

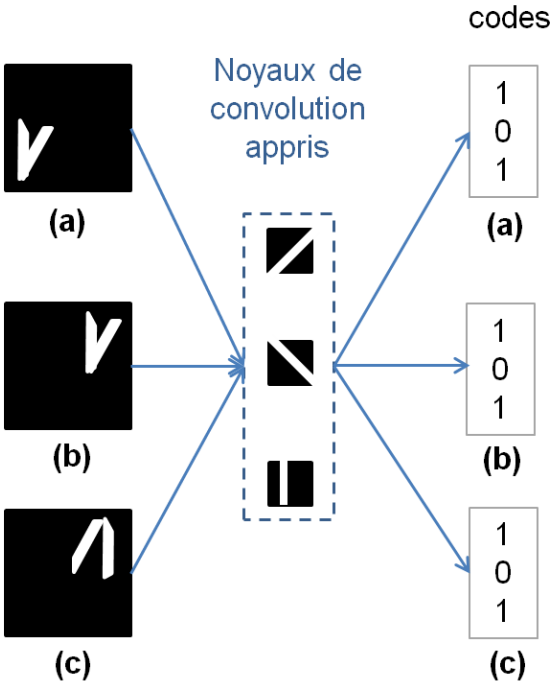

FIGURE 6.5 – Illustration sur un cas 2*D* de la limitation de l'approche introduite par Ranzato et al. [RHBL07] pour gérer l'invariance aux translations : Les entrées (a) et (b) seront encodées par le même code (vérifiant ainsi l'invariance à la translation), mais l'entrée (c) sera également encodée par le même code, bien qu'elle soit visuellement différente.

d'une même entrée.

Nous allons dans ce qui suit expliquer la différence principale entre les deux approches : La solution proposée par Ranzato et al. [RHBL07] utilise un "paramètre de transformation" par élément du code, alors que nous préconisons l'utilisation d'une seule variable de translation t_i pour l'ensemble des codes appris.

Ainsi, la méthode de Ranzato et al. pourra attribuer dans certains cas le même code à des entrées très différentes. Nous reportons sur la Figure 6.5 un exemple schématique (dans le cas 2*D*, pour simplifier) tel que présenté dans [RHBL07], et qui illustre cette limitation.

Avec la méthode de Ranzato et al., les noyaux de convolution 2*D* appris par l'encodeur sur ces exemples pourront correspondre à des motifs simples (tels que des traits horizontaux, verticaux et diagonaux). Ainsi, les codes correspondant aux entrées (a) et (b) (une version translatée de (a)) de la Figure 6.5 seront effectivement identiques, vérifiant ainsi l'invariance à la translation. En revanche, l'entrée (c) correspondra également à un code identique à celui de (a) et (b) bien qu'elle soit visuellement très différente. Ceci est dû au fait que chaque élément du code (les motifs appris par les noyaux de

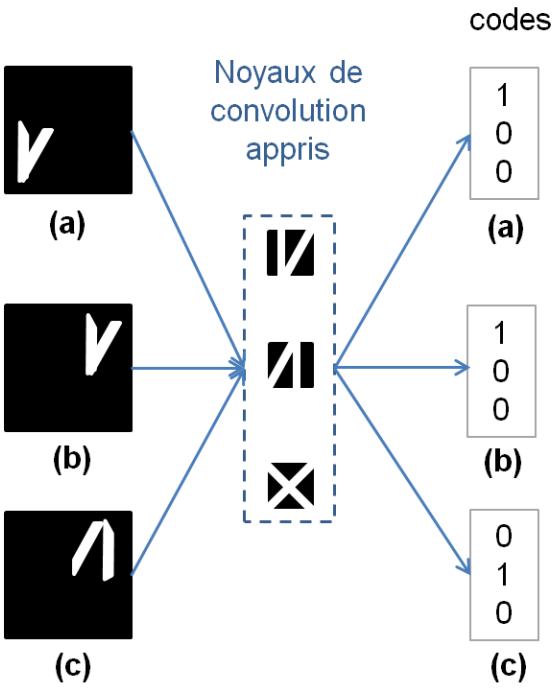

FIGURE 6.6 – Illustration du comportement de la méthode que nous proposons pour gérer l'invariance à la translation sur les exemples de la Figure 6.5 : Les entrées (a) et (b) seront également encodées par le même code, alors que l'entrée (c) sera encodée par un code différent.

convolution) peut avoir un paramètre de translation différent.

Nous illustrons sur la Figure 6.6 un schéma du comportement de la méthode que nous proposons sur les mêmes exemples utilisés pour la Figure 6.5. Notre approche recherchera dans les voisinages (a) et (c) les positions des patchs les plus représentatifs (selon la base d'apprentissage) et ne pourra donc pas générer le même code, puisqu'aucune sous-partie de (a) n'est une version translatée d'une sous partie de (c). Par contre, les exemples (a) et (b) obtiendront bien le même code, illustrant ainsi l'invariance à la translation.

A noter que nous allons évaluer et comparer les performances des deux approches dans le chapitre 7, dans le cadre de la reconnaissance d'actions humaines.

Sélection des patchs selon un critère de mouvement

Enfin, afin d'éviter d'encoder des informations non pertinentes pour la classification (par exemple la couleur ou la texture) et de réduire la quantité de données à encoder, nous proposons d'entraîner le modèle uniquement avec des patchs qui contiennent de l'information spatio-temporelle jugée conséquente.

Concrètement, un critère simple de sélection, basé sur le calcul de la différence en

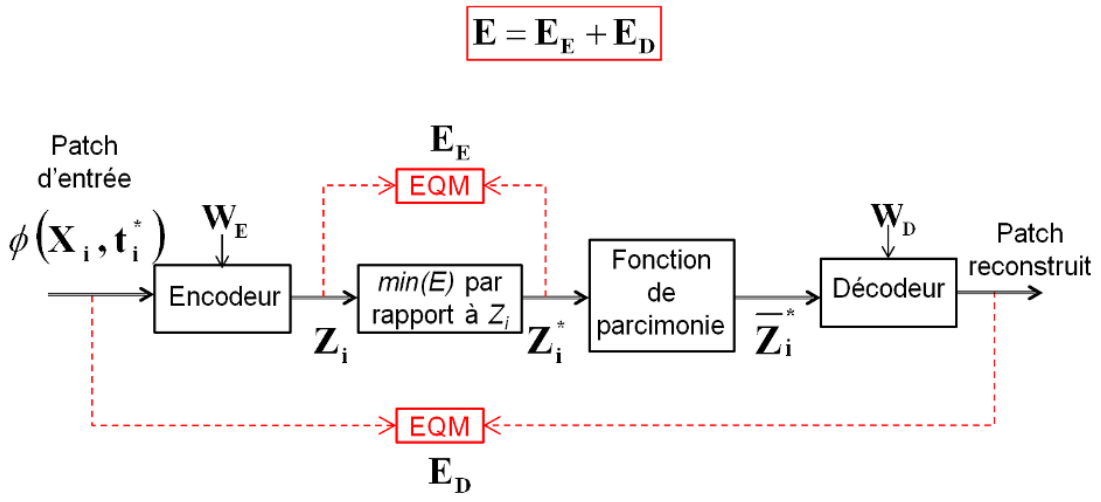

FIGURE 6.7 – Fonction objectif associée au modèle proposé. Les modules en amont de l'encodeur n'ont pas été représentés pour simplifier.

valeur absolue entre la première et la dernière image de l'entrée, est placé en amont. Ceci permet de ne garder que les patchs qui contiennent un pourcentage de pixels "en mouvement" qui est supérieur à un certain seuil. Cette sélection joue le même rôle que les détecteurs de saillance pour les caractéristiques manuelles (par exemple les points d'intérêt), mais elle n'utilise cependant aucun traitement complexe, vu que les patchs aberrants seront de toute façon filtrés lors de l'apprentissage du modèle, et ne seront pas encodés.

6.3 Apprentissage des paramètres du modèle

Nous allons nous intéresser dans cette section à l'apprentissage des paramètres du modèle présenté précédemment. Nous allons dans un premier temps introduire la fonction objectif globale associée à ce modèle, puis l'algorithme d'apprentissage qui permet de minimiser cette fonction objectif. Enfin, nous allons décrire quelques spécificités liées à l'algorithme de descente de gradient employé lors de l'apprentissage.

6.3.1 Fonction objectif

Comme dans [RPCL06, RHBL07], nous formulons l'apprentissage des paramètres du modèle sous forme d'une minimisation d'une fonction objectif globale.

Le modèle que nous avons décrit précédemment dépend de trois paramètres, à savoir ceux de l'encodeur W_E, ceux du décodeur W_D, et ceux relatifs à la translation optimale t_i. Cependant, un quatrième paramètre, correspondant au code Z_i, est aussi rajouté.

En effet, l'une des conséquences de l'introduction de la fonction de parcimonie (qui est fortement non linéaire) entre l'encodeur et le décodeur est la non faisabilité de l'apprentissage conjoint de ces deux modules. Si nous prenons l'exemple d'un auto-encodeur neuronal entraîné avec une rétro-propagation du gradient (même si le modèle proposé est générique, et reste généralisable à des modèles non neuronaux), l'erreur de reconstruction issue du décodeur est remise à zéro au niveau de la fonction de parcimonie, et ne parvient donc pas jusqu'à l'encodeur.

Ainsi, comme dans [RPCL06, RHBL07], les deux modules sont entraînés séparément : Les paramètres de l'un sont fixés pendant la mise à jour de ceux de l'autre, et inversement. Il est alors nécessaire d'introduire un module (entre l'encodeur et la fonction de parcimonie, comme illustré sur la Figure 6.7) afin de produire le code "optimal" Z_i^* pour la reconstruction par le décodeur (sans être éloigné de celui produit par l'encodeur).

Concrètement, le code Z_i est considéré comme un paramètre supplémentaire de la fonction objectif globale qui caractérise le modèle (Plus de détails seront donnés dans la sous-section 6.3.2).

La fonction objectif globale dépend donc des paramètres (t_i, Z_i, W_E, W_D), et est exprimée par :

$$\begin{aligned} E\left(X_i, t_i, Z_i, W_E, W_D\right) &= E_E\left(X_i, t_i, Z_i, W_E\right) + E_D\left(X_i, t_i, Z_i, W_D\right) \\ &= \left\|Z_i - Enc\left(W_E, \phi\left(X_i, t_i\right)\right)\right\|^2 + \left\|Dec\left(W_D, \overline{Z}_i\right) - \phi\left(X_i, t_i\right)\right\|^2 \end{aligned} \tag{6.3}$$

où $Enc\left(W_E, X_i\right)$ désigne la sortie de l'encodeur calculée pour un patch d'entrée X_i donné, et $Dec\left(W_D, \overline{Z}_i\right)$ est la sortie du décodeur obtenue par un code parcimonieux $\overline{Z}_i$.

E est une somme de deux termes, représentant respectivement l'erreur quadratique moyenne (EQM) de prédiction de l'encodeur, et l'EQM de reconstruction du décodeur (cf. Figure 6.7). A noter que cette fonction objectif est globalement la même que celle introduite par Ranzato et al. [RPCL06, RHBL07], à la différence : (i) Que les données X_i sont tri-dimensionnelles, et (ii) qu'elle intègre la variable latente supplémentaire t_i, ce qui modifie certains termes de l'équation.

La sous-section suivante présente la procédure employée pour minimiser cette fonction objectif par rapport aux paramètres (t_i, Z_i, W_E, W_D).

6.3.2 Algorithme d'apprentissage

Comme nous l'avons évoqué précédemment, le modèle d'auto-encodage proposé est entraîné "en ligne", c'est à dire qu'une mise à jour de ses paramètres est effectuée pour chaque exemple d'apprentissage, de manière à minimiser la fonction objectif E expri-

mée par l'équation 6.3. La procédure d'apprentissage reprend le même schéma global que dans [RPCL06, RHBL07], tout en tenant compte des nouveautés introduites dans notre modèle. L'apprentissage s'effectue en tois étapes, chacune d'entre-elles visant à minimiser E par rapport à l'un des paramètres t_i, Z_i et (W_E, W_D), en gardant les autres constants. Il s'agit alors de calculer les paramètres optimaux t_i^*, Z_i^* et (W_E^*, W_D^*) exprimés par :

$$t_i^* = \arg\min_{t_i} E\left(t_i | X_i, Z_i, W_E, W_D\right) \tag{6.4}$$

$$Z_i^* = \arg\min_{Z_i} E\left(Z_i | X_i, t_i^*, W_E, W_D\right) \tag{6.5}$$

$$(W_E^*, W_D^*) = \arg\min_{W_E, W_D} E\left(W_E, W_D | X_i, t_i^*, Z_i^*\right) \tag{6.6}$$

où la notation $E(a|b)$ fait référence au fait que l'optimisation est effectuée sur le(s) paramètre(s) a en gardant le(s) paramètre(s) b constant(s).

Les équations 6.4, 6.5 et 6.6 expriment chacune un problème quadratique localement convexe, qui peut être résolu simplement en optimisant, de manière itérative, chacun des paramètres en gardant les autres constants.

Ceci est d'autant plus simple si l'encodeur et le décodeur sont des fonctions linéaires, comme c'est le cas pour l'architecture que nous proposons (cf. section 6.4). L'algorithme utilisé est alors le suivant :

1. (W_E, W_D) sont initialisés aléatoirement.

2. Pour un patch X_i donné, une recherche exhaustive est effectuée sur un voisinage spatio-temporel de X_i afin de trouver la translation optimale t_i^* définie par l'équation 6.4, c'est à dire celle pour laquelle l'erreur quadratique moyenne de reconstruction du décodeur (E_D) sera minimale.

3. Étant donné le code Z_i correspondant au patch translaté $\phi\left(X_i, t_i^*\right)$, l'équation 6.5 est résolue en effectuant une descente de gradient sur le paramètre Z_i afin de calculer le code optimal Z_i^*. Les détails relatifs à cette descente du gradient sont donnés ci-après.

4. Les paramètres W_D du décodeur sont mis-à-jour avec une rétro-propagation du gradient, en ayant comme entrée $\overline{Z}_i^*$, et comme sortie désirée le patch translaté $\phi\left(X_i, t_i^*\right)$.

5. Les paramètres W_E de l'encodeur sont mis-à-jour avec une rétro-propagation du

gradient, en ayant comme entrée $\phi\left(X_i, t_i^*\right)$, et comme sortie désirée le code optimal Z_i^*.

A noter que les étapes (2), (3), (4) et (5) sont répétées pour chaque patch d'apprentissage X_i. A noter également que les étapes (4) et (5) supposent que les fonctions $Enc\left(.\right)$ et $Dec\left(.\right)$ sont dérivables par rapport aux paramètres W_E et W_D. Cela est le cas pour les architectures que nous avons choisies, et que nous allons détailler dans la section 6.4.

6.3.3 Descente du gradient

La descente du gradient effectuée lors de l'étape (3) de l'algorithme d'apprentissage est une descente du gradient classique (initialisée en Z_i, et avec un pas de descente variable), avec néanmoins les deux particularités suivantes :

- La dérivée partielle de E par rapport à Z_i (nécessaire lors du calcul du gradient) est approximée par une différence finie :

$$\frac{\partial E\left(Z_i|X_i, t_i^*, W_E, W_D\right)}{\partial Z_i} \approx \frac{E\left(Z_i + dZ|X_i, t_i^*, W_E, W_D\right) - E\left(Z_i - dZ|X_i, t_i^*, W_E, W_D\right)}{dZ} \tag{6.7}$$

  où dZ est une faible valeur positive non nulle.

- Nous appliquons un algorithme du signe dans lequel le calcul du gradient ne sert qu'à déterminer la direction de la descente. L'amplitude du déplacement est fixée à $0,01$ (valeur empirique), et elle décroît toutes les 10 itérations (en la multipliant par $0,8$). Ceci permet d'accélérer considérablement l'optimisation.

Nous allons décrire dans la section suivante les détails architecturaux de l'encodeur et du décodeur que nous avons utilisés pour l'apprentissage non supervisé des caractéristiques parcimonieuses.

6.4 Architecture de l'encodeur et du décodeur

L'architecture proposée est illustrée sur la Figure 6.8, et est basée sur les réseaux de neurones à convolutions $3D$ (cf. section 5.2 du chapitre 5). A noter néanmoins que l'approche décrite dans les sections précédentes (schéma général, apprentissage, ...) est générique et reste valable pour d'autres modèles d'auto-encodage tels que les RBMs.

Les détails architecturaux de chacun des modules d'encodage et de décodage de la Figure 6.8 sont présentés dans ce qui suit.

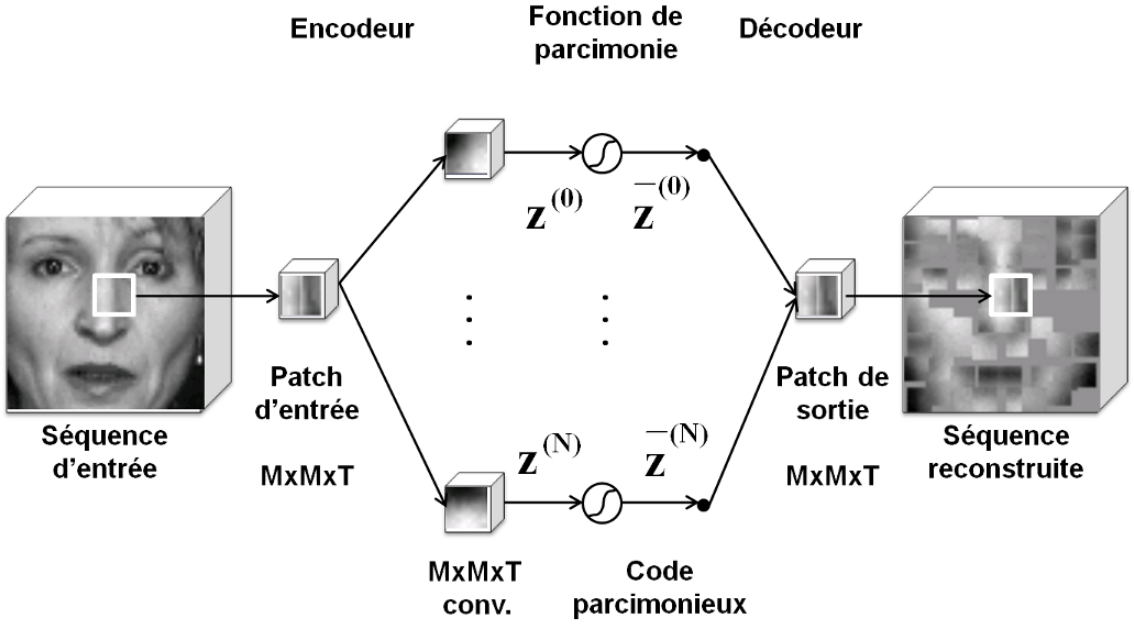

FIGURE 6.8 – Architecture de l'auto-encodeur parcimonieux à convolutions 3D proposé : Illustration sur un exemple de la base GEMEP-FERA d'expressions faciales.

6.4.1 L'encodeur

L'encodeur consiste en un ensemble de N noyaux de convolutions 3D, de taille $M \times M \times T$ chacun. Il prend en entrée un patch 3D de taille $M \times M \times T$ (issu de la recherche de la translation optimale sur un voisinage de taille $3M/2 \times 3M/2 \times 2T$) et apprend à générer un code non-parcimonieux de taille N, qui correspond aux réponses de chacune des convolutions (cf. Figure 6.8), et qui encode l'information spatio-temporelle saillante contenue dans le patch d'entrée.

Chacun des N noyaux de convolution contient, en plus de ses $M \times M \times T$ paramètres, un terme de biais supplémentaire. Ainsi, le nombre total de paramètres de l'encodeur est égal de $(M \times M \times T + 1) \times N$.

Enfin, et afin que la représentation soit sur-complète, N doit être supérieur ou égal à la dimension de l'entrée. Nous verrons dans le chapitre 7 que la forte corrélation qui existe entre les T images successives de la vidéo, et qui fait que l'entrée a une dimension "effective" (dans le sens où elle contient de l'information pertinente) inférieure à $M \times M \times T$, permet de considérer dans la pratique des valeurs de N inférieures à la dimension de l'entrée, tout en préservant le caractère sur-complet de la représentation.

FIGURE 6.9 – Quelques exemples d'éléments de la "base" obtenus par apprentissage sur les données : (a) - KTH d'actions humaines (b) - GEMEP-FERA d'expressions faciales. Dans les deux cas, chaque élément de la base est composé de trois images de tailles 8×8 chacune ($T = 3$ et $M = 8$).

6.4.2 Le décodeur

Le décodeur contient $M \times M \times T$ neurones de sortie totalement connectés aux sorties de la fonction de parcimonie, ainsi qu'à un terme de biais. Le nombre total de paramètres du décodeur est donc de $(N \times M \times M \times T) + 1$.

Chaque patch reconstruit en sortie peut être obtenu par une somme pondérée de plusieurs patchs spatio-temporels élémentaires. Ces derniers forment un ensemble généralement appelé "base" ou "dictionnaire", et correspondent aux N réponses du décodeur quand celui-ci est stimulé par chacun des N codes ayant une seule valeur non nulle égale à 1. En réalité, cet ensemble de patchs élémentaires forme une famille génératrice de l'espace dans lequel sont reconstruites les données, mais pas forcément une base vu que ses différents patchs élémentaires ne sont pas linéairement indépendants. Le terme "base", que nous emploierons dans ce qui suit, est donc uniquement utilisé par abus de langage.

Vu que le code est parcimonieux, seuls quelques éléments de la base sont utilisés pour reconstruire chaque patch de sortie (les autres éléments du code étant nuls). Typiquement, dans toutes nos expérimentations, le nombre d'unités activées a toujours été inférieur à $N/8$.

Nous reportons sur les Figure 6.9-(a) et 6.9-(b) quelques exemples d'éléments de la "base" obtenus par apprentissage respectivement sur les données KTH d'actions humaines [SLC04], et GEMEP-FERA d'expressions faciales [VJM+11]. Nous avons vérifié qu'aucun élément des bases que nous avons obtenu n'est une version translatée d'un autre, et qu'ils contiennent tous du mouvement (grâce à la recherche de la meilleure translation et à la pré-sélection des patchs qui contiennent du mouvement lors de l'apprentissage). Les éléments de la base semblent aussi visuellement pertinents puisqu'ils correspondent pour la plupart à des parties du corps (bras, jambes, tête,...) pour le cas des actions humaines, et à des parties du visage (contours des yeux, de la bouche, sour-

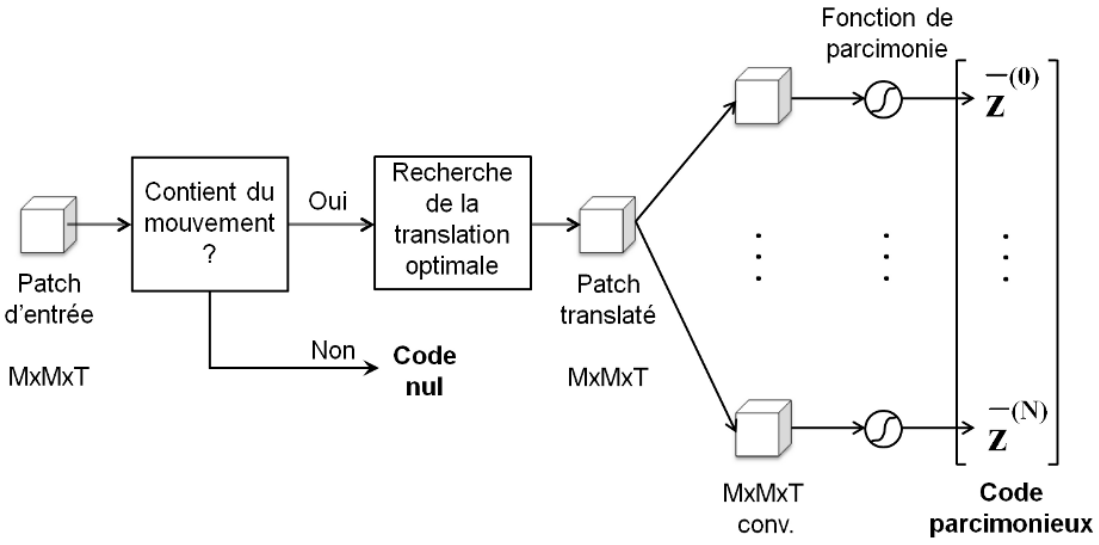

FIGURE 6.10 – Illustration du processus d'extraction d'un code parcimonieux à partir d'un patch spatio-temporel (après l'apprentissage).

cils, ...) de pour le cas des expressions faciales.

6.5 Classification des séquences vidéo complètes

Nous avons vu que le modèle d'auto-encodage décrit précédemment permettait de représenter l'information spatio-temporelle contenue dans le voisinage d'un patch $3D$ par un code parcimonieux de taille N. Chaque code décrit donc le contenu d'un certain voisinage local de taille réduite (spatialement et temporellement) par rapport à la vidéo complète.

Nous allons présenter dans cette section le processus de représentation des vidéos complètes par ces codes parcimonieux. L'extraction de ces codes pour un patch spatio-temporel donné sera d'abord décrite dans la sous-section 6.5.1. La génération des séquences de caractéristiques décrivant les vidéos complètes, ainsi que leur utilisation pour la classification BLSTM, sera ensuite détaillée dans la sous-section 6.5.2.

6.5.1 Extraction des codes parcimonieux

Une fois l'apprentissage terminé, l'extraction des codes parcimonieux à partir des patchs spatio-temporels se fait selon le processus illustré sur la Figure 6.10.

Les patchs qui contiennent une information spatio-temporelle conséquente sont sélectionnés selon un critère sur le nombre de pixels contenant du mouvement, comme

expliqué dans la section 6.2. Les patchs non sélectionnés se voient alors attribuer un code qui ne contient que des valeurs nulles.

Une recherche de la translation optimale sur un voisinage de taille $3M/2 \times 3M/2 \times 2T$ centré sur le patch courant est ensuite effectuée, et le patch translaté correspondant à l'erreur d'encodage et de décodage minimale est retenu. Ce dernier est utilisé pour stimuler l'encodeur, qui y applique les noyaux de convolutions appris, obtenant ainsi un code non parcimonieux.

La fonction de parcimonie, introduite par Ranzato et al. [RPCL06, RHBL07] et décrite dans la section 6.2, est enfin appliquée afin de calculer le code parcimonieux correspondant. A noter que cette fonction se transforme après l'apprentissage en une simple fonction logistique à gain fixe. En effet, durant l'apprentissage, les valeurs de ξ obtenues par l'équation 6.2 sont sauvegardées. Une fois celui-ci terminé, elles sont remplacées par sa valeur moyenne ξ_{moy} calculée sur tous les échantillons lors de la dernière itération. L'équation 6.1 devient alors celle d'une fonction logistique classique :

$$\bar{z}_i^{(k)} = \frac{1}{1 + e^{-\beta\left[z_i^{(k)} - \frac{1}{\beta} ln\left(\frac{1-\eta}{\eta}\xi_{moy}\right)\right]}} \tag{6.8}$$

Nous allons présenter dans la sous-section suivante comment ces codes parcimonieux ainsi obtenus sont utilisés pour générer des séquences de caractéristiques, et classer les vidéos complètes.

6.5.2 Génération des séquences de caractéristiques et classification BLSTM

La Figure 6.11 illustre le principe de la génération de ces séquences de caractéristiques à partir des codes parcimonieux, sur un exemple de la base KTH d'actions humaines [SLC04].

Les séquences de caractéristiques sont générées en concaténant les réponses des patchs quand ils sont placés sur la grille des emplacements possibles sur le bloc spatio-temporel (cf. Figure 6.11). Si nous notons par H et W respectivement le nombre de lignes et de colonnes des images originales de la vidéo, chacun des blocs contiendra $W/M \times H/M$ emplacements possibles pour les patchs (qui ont une taille de $M \times M \times T$ chacun). Les séquences de caractéristiques contiendront ainsi $W/M \times H/M \times N$ valeurs à chaque instant. A noter que chacun des $W/M \times H/M$ patchs est le centre d'un voisinage spatio-temporel de taille $3M/2 \times 3M/2 \times 2T$ (sauf pour le cas des bords) dans lequel est effectuée la recherche de la translation optimale.

Ces séquences de caractéristiques ainsi obtenues sont ensuite utilisées pour entraîner un modèle de classification BLSTM, qui prend en compte l'évolution temporelle des

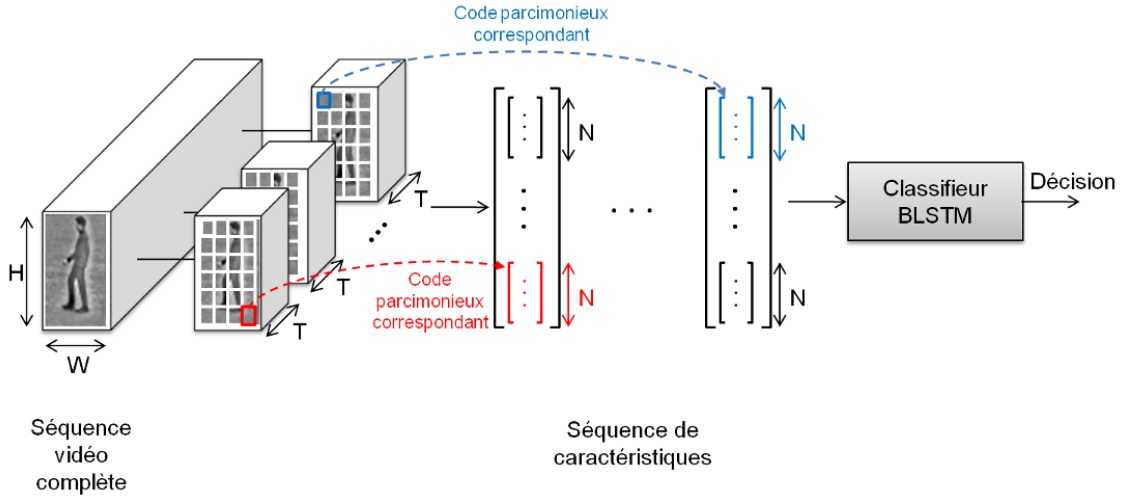

FIGURE 6.11 – Illustration de la génération des séquences de caractéristiques (à partir des codes parcimonieux appris), et de la classification BLSTM des séquences vidéo complètes.

caractéristiques apprises pour attribuer un label à la séquence complète, comme décrit dans les chapitres précédents. Pour rappel, le choix du classifieur BLSTM est justifié par son fort pouvoir discriminant qui été vérifié lors de l'étude comparative présentée dans le chapitre 4. Les résultats expérimentaux relatifs à cette classification BLSTM (ainsi que d'autres) seront présentés dans le chapitre 7.

6.6 Conclusion

Après nous être intéressés lors du chapitre 5 à l'apprentissage supervisé des caractéristiques spatio-temporelles, nous avons exploré dans ce chapitre la piste de l'apprentissage non supervisé. Ainsi, nous avons proposé un modèle d'auto-encodage qui permet de générer une représentation parcimonieuse sur-complète des entrées. Ce modèle reprend le schéma général introduit par Ranzato et al. [RPCL06, RHBL07] pour le cas des images fixes tout en proposant un certain nombre de nouveautés. Celui-ci est entraîné avec des patchs $3D$, et apprend à générer un code parcimonieux qui décrit les motifs spatio-temporels saillants présents dans ces patchs. Une pré-sélection des entrées contenant une information spatio-temporelle significative permet de ne pas encoder des informations non pertinentes pour la classification vidéo. La fonction objectif associée à ce modèle, qui dépend de quatre paramètres, a ensuite été présentée, ainsi que l'algorithme d'apprentissage associé qui permet de la minimiser.

Nous avons aussi proposé une nouvelle approche pour assurer l'invariance aux

translations spatiales et temporelles de la représentation apprise. Nous avons pour cela proposé de rajouter une variable latente et un terme supplémentaire à la fonction objectif globale du modèle, ce qui permet de représenter le voisinage spatio-temporel d'un patch donné par un seul code parcimonieux. Cette nouvelle approche sera évaluée et comparée à celle proposée par Ranzato et al. [RHBL07] (et plus précisément à son extension au cas de la vidéo) au cours du chapitre suivant.

Nous avons enfin présenté comment cette représentation parcimonieuse des données pouvait servir à classer les séquences vidéo complètes. Pour ce faire, nous avons décrit le processus de génération des séquences de caractéristiques à partir des codes parcimonieux, ainsi que l'utilisation de ces séquences de caractéristiques pour entraîner un modèle de classification BLSTM à attribuer un label à la séquence vidéo, en se basant sur l'évolution temporelle des codes parcimonieux.

Le modèle proposé dans ce chapitre, ainsi que celui basé sur l'apprentissage supervisé profond des caractéristiques spatio-temporelles (qui a été présenté lors du chapitre 5), seront évalués et comparés dans le cadre de deux problématiques différentes (à savoir la reconnaissance d'actions humaines et la reconnaissance d'expressions faciales) afin de vérifier leur généricité. Les différents résultats expérimentaux relatifs à ces deux modèles feront l'objet du chapitre suivant.

Chapitre 7

Résultats expérimentaux

Sommaire

7.1 Introduction

Ce chapitre est consacré à la présentation des résultats expérimentaux relatifs aux deux modèles introduits lors des chapitres 5 et 6, à savoir le modèle *ConvNet* 3*D* et le modèle d'auto-encodage parcimonieux.

Comme nous l'avons évoqué précédemment (cf. chapitres 5 et 6), l'une des principales motivations derrière l'introduction de ces deux modèles est leur généricité en comparaison avec les approches basées sur les caractéristiques manuelles, qui sont souvent liées à un domaine donné. Les expérimentations seront donc effectuées sur deux problématiques différentes : La reconnaissance d'actions humaines, et la reconnaissance d'expressions faciales, afin de vérifier cette généricité.

Par ailleurs, les résultats expérimentaux présentés dans ce chapitre serviront également à évaluer les performances des deux modèles proposés, et de les comparer entre-eux et aux principaux travaux de l'état de l'art sur les deux problématiques étudiées.

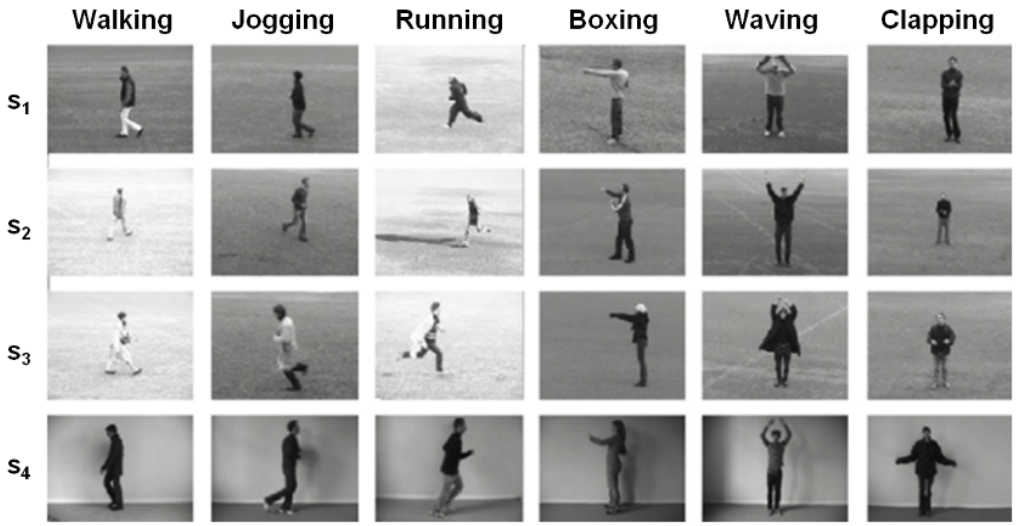

FIGURE 7.1 – Quelques exemples d'actions/scénarii de la base KTH d'actions humaines [SLC04].

Le reste de ce chapitre s'organisera comme suit : Nous nous intéresserons dans un premier temps dans la section 7.2 aux données utilisées pour chaque problématique, ainsi qu'aux protocoles d'évaluation et aux pré-traitements correspondants. Ensuite, nous présenterons dans les sections 7.3 et 7.4 les résultats expérimentaux relatifs aux deux modèles d'apprentissage des caractéristiques. Nous comparerons ensuite dans la section 7.5 les résultats à ceux obtenus par les principaux travaux de l'état de l'art sur les deux problématiques. Nous présenterons enfin dans la section 7.6 quelques résultats complémentaires, qui traiteront principalement de l'évaluation des différentes parties de chacun des deux modèles.

7.2 Données utilisées, protocoles d'évaluation et pré-traitements

7.2.1 Base KTH d'actions humaines

Présentation de la base

La base KTH d'actions humaines[1] a été introduite par Schuldt et al. [SLC04] en 2004, et a été depuis l'une des bases publiques les plus utilisées dans le domaine de la reconnaissance d'actions humaines.

La base contient six types d'actions : Marcher (*walking*), courir lentement (*jogging*), courir rapidement (*running*), boxer (*boxing*), faire un mouvement circulaire des bras (*wa-*

[1] Disponible en ligne sur : http://www.nada.kth.se/cvap/actions/

ving) et applaudir (*clapping*). Ces actions sont effectuées par 25 acteurs selon quatre scénarii différents : Scènes en extérieur (s_1), changement d'échelle (s_2), changement de vêtements (s_3) et scènes en intérieur (s_4) (cf. Figure 7.1).

La taille des images pour chaque vidéo est de 160×120 pixels, échantillonnées temporellement à 25 images par seconde. Un seul personnage est présent sur chaque vidéo, et il effectue une seule action sur un fond homogène, avec toutefois la présence d'ombres qui rendent difficile une éventuelle soustraction de fond. De plus, il existe d'autres types de variations comme la durée des actions, l'angle de prise de vue (avec notamment des zooms pour le scénario s_2) et les conditions d'éclairage (entre les scènes tournées en extérieur et celles tournées en intérieur). A noter que pour les classes pour lesquelles le personnage est en mouvement (*walking*, *jogging* et *running*), celui-ci peut-être effectué de gauche à droite ou de droite à gauche. A noter aussi que l'emplacement du personnage varie d'une vidéo à une autre, et que la caméra n'est pas centrée sur lui.

Il existe dans la littérature deux versions de la base KTH, comme l'ont souligné Gao et al. dans [GCHC10] :

- La première version (qui est la version originale présentée par les auteurs dans [SLC04]), est appelée KTH1, et contient 599 longues séquences (dont la durée varie entre 8 et 59 secondes). Chaque action est répétée 3 ou 4 fois dans une même vidéo, et il existe des images "vides" (sans action) entre les différentes répétitions.

- La deuxième version, appelée KTH2, est obtenue en découpant chaque vidéo en plusieurs courtes séquences, de manière qu'il n'y ait qu'une seule itération de l'action par séquence. Les informations relatives à la procédure de découpage (instants de découpage pour chaque séquence, nombre de sous-séquence extraites de chaque séquence, ...) sont fournies par les auteurs sur le site de la base, assurant ainsi le fait que le découpage est le même pour tous les utilisateurs de la base KTH2. Cette dernière contient ainsi 2391 courtes séquences, dont la longueur varie entre 1 et 14 secondes.

A noter que pour les deux versions, les vidéos sont réparties de manière équilibrée sur les six classes.

Protocoles d'évaluation

Plusieurs répartitions des séquences de la base KTH en apprentissage et test ont été utilisées dans la littérature. Dans [GCHC10], Gao et al. ont présenté une étude complète, spécifique aux données de KTH, qui mesure l'influence du protocole d'évaluation utilisé sur les résultats obtenus, et ont ainsi démontré que les performances d'une approche

donnée pouvaient varier de $\pm 9\%$ selon la répartition apprentissage/test utilisée, et de $\pm 5.85\%$ selon qu'elle soit évaluée sur KTH1 ou KTH2. Or, comme l'ont fait remarqué Gao et al. dans [GCHC10], les approches présentées dans la littérature sont souvent directement comparées entre-elles alors qu'elles n'utilisent pas les mêmes protocoles d'évaluation, et les mêmes données (KTH1 ou KTH2) ce qui peut fausser les conclusions.

Nous avons choisi dans le cadre de cette thèse de nous baser sur deux protocoles d'évaluation sur la base KTH, qui sont les plus utilisés dans l'état de l'art :

- Une première méthode dans laquelle les séquences correspondant à 16 personnes sélectionnées aléatoirement sont utilisées pour l'apprentissage, et celles correspondant aux 9 autres sont utilisées pour le test. Cette répartition est celle spécifiée sur le site de la base. Le taux de reconnaissance permettant d'évaluer le modèle est obtenu en moyennant ceux correspondant à 5 configurations générées aléatoirement. Ce protocole d'évaluation est appelé validation croisée à 5 répartitions (*5-fold cross validation* en anglais).

- Une deuxième méthode dans laquelle une seule personne à la fois est testée, en utilisant les 24 autres pour l'apprentissage. Le résultat final étant obtenu en moyennant les 25 résultats obtenus pour chacune des personnes testées. Cette procédure est appelée *leave-one-out* en anglais.

A noter qu'une attention particulière sera portée lors de la comparaison des performances avec l'état de l'art (cf. la section 7.5) sur les protocoles d'évaluation ainsi que sur les données utilisées par les travaux auxquels nous nous comparerons, afin de garantir la pertinence des conclusions.

Pré-traitements

Toutes les vidéos de la base KTH ont subi les pré-traitements suivants : (i) Un sous-échantillonnage spatial d'un facteur 2 horizontalement et verticalement afin de réduire la complexité et la taille en mémoire des données à traiter (ii) une extraction de la boîte englobante centrée sur le personnage, comme décrit dans [JSWP07, JXYY10], et (iii) l'application d'une normalisation locale du contraste sur des voisinages spatio-temporels de taille $7 \times 7 \times 7$, comme l'ont recommandé Jarret et al. dans [JKRL09]. A noter que contrairement à d'autres travaux [KLY07, JXYY10], nous n'utilisons aucun pré-traitement complexe (flot optique, gradients,...) afin de préserver la généricité de l'approche et le fait qu'elle opère sur des données quasi-brutes.

FIGURE 7.2 – Les cinq émotions représentées dans la base GEMEP-FERA d'expressions faciales [VJM+11].

7.2.2 Base GEMEP-FERA d'expressions faciales

Présentation de la base

La base GEMEP-FERA d'expressions faciales a été introduite récemment par Valstar et al. [VJM+11] dans le cadre du challenge FERA 2011[2] (*Facial Expression Recognition and Analysis*) qui s'est tenu en marge de la conférence internationale *Face and Gesture Recognition 2011*.

La base GEMEP-FERA contient deux parties correspondant chacune aux deux défis traités par le challenge FERA 2011, à savoir la détection des unités d'actions du système FACS (*Facial Unit Conding System*) et la reconnaissance d'émotions. C'est cette dernière partie de la base que nous avons utilisé dans le cadre de cette thèse. Elle contient cinq types d'expressions faciales : Colère (*anger*), peur (*fear*), joie (*joy*), soulagement (*relief*), et tristesse (*sadness*), jouées par dix acteurs différents, tout en prononçant une phrase vide de sens ou une voyelle soutenue "aaa" (cf. Figure 7.2).

La résolution des images est de 720×567, échantillonnées à 25 images par seconde. La durée des vidéos est comprise entre 1 et 6 secondes

Cette base est particulièrement difficile à traiter à cause notamment de l'existence d'une forte confusion intra-classe (entre *joie/soulagement* d'un côté, et *colère/peur/tristesse* de l'autre), ce qui rend la reconnaissance particulièrement difficile

[2] Pour plus d'informations sur le challenge FERA 2011 : http://sspnet.eu/fera2011/

même pour un humain. De plus, une difficulté supplémentaire réside dans le fait que les vidéos ne sont pas centrées sur les visages des acteurs, et comportent plusieurs mouvements non faciaux (mouvement des mains, du corps, ...). Ceci implique qu'une première phase de détection devra être effectuée avant la reconnaissance, ce qui n'est pas simple vu que de nombreux flous sont présents sur les vidéos.

Protocole d'évaluation

La répartition des données entre apprentissage et test est décrite dans l'article qui présente la base [VJM+11] : 155 vidéos sont utilisées pour l'apprentissage, et 134 pour le test. La répartition du nombre total de vidéos entre les classes est la suivante : 59 vidéos pour la classe "colère", 56 pour la classe "peur", 61 pour la classe "joie", 57 pour la classe "soulagement", et 56 pour la classe "tristesse".

La métrique utilisée pour évaluer les performances est le taux de bonne reconnaissance calculé sur la base de test. Cette dernière n'étant pas annotée, les performances sont évaluées par les organisateurs du challenge.

La base d'apprentissage contient 7 personnes, alors que la base de test en contient 6, dont 3 non présents dans la base d'apprentissage. Ceci permet d'évaluer, en plus de la performance globale d'un modèle donné, son pouvoir de généralisation en calculant deux scores : Le taux de reconnaissance PI (*Person Independant* en anglais, qui est calculé uniquement sur les 3 personnes non présentes dans la base d'apprentissage, et qui évalue donc le pouvoir de généralisation d'un modèle donné) et le PS (*Person Specific*, qui correspond aux 3 autres personnes).

Pré-traitements

Les vidéos de la base GEMEP-FERA ont subi les pré-traitements suivants : (i) Un sous-échantillonnage spatial de facteur 4 horizontalement et verticalement, (ii) un algorithme de détection de visages, et (iii) une transformation en niveaux de gris. La détection consiste à extraire une boîte englobante du visage de taille 64×64 en utilisant un modèle neuronal (appelé *Convolutional Face Finder*) qui a été introduit par Garcia et Delakis dans [GD04]. L'alignement des visages sur les boîtes englobantes consécutives est ensuite obtenu en appliquant l'algorithme proposé par Duffner et Garcia dans [DG08b], et le suivi de ces visages est fait selon l'algorithme décrit dans [MRG07]. Pour finir, les boîtes englobantes des visages sont mises à l'échelle pour obtenir des images de taille 64×64. A noter que ces différents pré-traitements sont communs à la quasi-totalité des participants au challenge FERA 2011, avec néanmoins des approches différentes, d'un participant à un autre, pour chaque type de pré-traitement.

7.3 Évaluation des performances du modèle *ConvNet* 3*D*

Nous allons nous intéresser dans cette section à l'évaluation des performances du modèle *ConvNet* 3*D* que nous avons introduit dans le chapitre 5, qui permet d'apprendre de manière supervisée des caractéristiques spatio-temporelles. Cette évaluation sera faite sur les deux bases présentées dans la section précédente, qui correspondent aux problématiques de la reconnaissance d'actions humaines et la reconnaissance d'expressions faciales.

7.3.1 Reconnaissance d'actions humaines

De nombreuses architectures, correspondant au modèle illustré sur la Figure 5.3 du chapitre 5, ont été testées. L'architecture retenue est la suivante : Elle prend en entrée des volumes spatio-temporels de taille $34 \times 54 \times 9$. La taille des noyaux des convolutions 3*D* au niveau des 7 modules C_1, des 35 modules C_2 et des 5 modules C_3 sont respectivement de $7 \times 7 \times 5$, $5 \times 5 \times 3$ et $3 \times 3 \times 3$. Enfin, les couches de sous-échantillonnage S_1 et S_2, de rectification R_1 et R_2, ainsi que les deux dernières couches de classification sont celles décrites dans le chapitre 5. Cette architecture correspond à un nombre total d'environ $17 \cdot 10^3$ paramètres, ce qui est 15 fois moins que l'architecture proposée dans [JXYY10]. Le modèle est entraîné en ciblant, pour chaque sous-séquence d'entrée, la classe de la séquence complète.

Le sous-échantillonnage effectué au niveau des couches S_1 et S_2 se fait uniquement dans le domaine spatial. En effet, nous avons vérifié que le sous-échantillonnage temporel réduisait la quantité d'informations (vu que les séquences de caractéristiques sont plus courtes), et diminuait ainsi les performances de la classification. A noter que nous avons également testé un certain nombre de variantes architecturales relatives à ce modèle, que nous ne présentons pas dans ce manuscrit. Ces différents tests ont permis par exemple de vérifier l'importance des couches de rectification pour la robustesse aux changements d'habits et d'éclairage. Nous avons également testé différentes tailles d'entrée en faisant varier le nombre d'images entre 3 et 15, ce qui a permis de sélectionner les entrées composées de 9 images successives comme étant celles qui donnent les meilleurs résultats pour la reconnaissance d'actions humaines.

Comme nous l'avions indiqué lors de la section 5.4 du chapitre 5, nous avons évalué deux stratégies différentes pour la classification des séquences vidéo complètes à partir des caractéristiques générées par le modèle *ConvNet* 3*D* :

- Une classification par vote majoritaire sur les décisions individuelles relatives à

chacun des blocs spatio-temporel composés de 9 images successives.

- Une classification BLSTM qui est entraînée avec les séquences de caractéristiques extraites de la couche C_3.

Compte tenu de l'architecture utilisée (décrite ci-dessus), de la taille des entrées et du nombre de modules de la couche C_3, cette dernière produit 5 cartes de caractéristiques de taille 3×8 chacune. Les séquences de caractéristiques contiennent ainsi 120 valeurs par instant. L'architecture du modèle BLSTM utilisé est similaire à celle utilisée pour l'étude comparative entre les modèles de classification décrite dans le chapitre 4, avec une couche d'entrée de taille 120, une couche cachée qui contient 50 neurones LSTM pour chaque direction, et une couche de sortie de taille 6 (une sortie par classe). Cette architecture du modèle BLSTM a été sélectionnée empiriquement, après avoir testé plusieurs autres configurations possibles.

A noter que dans nos expérimentations, nous avons observé qu'aucun sur apprentissage n'était observé (aussi bien pour le modèle *ConvNet* 3*D* que pour la classification BLSTM) si l'apprentissage est arrêté à la première itération pour laquelle l'erreur moyenne sur les séquences d'apprentissage ne diminue plus. Ceci permet de se passer d'une éventuelle base de validation, qui réduirait le nombre d'exemples utilisés pour l'apprentissage.

Nous reportons sur le tableau 7.1 les résultats ainsi obtenus par les deux stratégies de classification des séquences complètes. La protocole d'évaluation utilisé dans ce cas est celui de la validation croisée sur 5 configurations sélectionnées aléatoirement à partir des bases KTH1 et KTH2 (cf. sous-section 7.2.1).

La première observation à propos de ces résultats concerne la classification par vote majoritaire sur les sous-séquences (*ConvNet* 3*D* + vote dans le tableau 7.1) qui, bien que le vote ne porte que sur un faible nombre d'images successives (en l'occurrence 9 images), permet déjà d'obtenir des résultats relativement satisfaisants (avec 91,04% pour KTH1 et 89,40% pour KTH2). Nous verrons également dans la section 7.5 que cette approche simple permet d'obtenir des résultats équivalents à ceux obtenus par Ji et al. [JXYY10] (qui ont proposé un modèle neuronal à convolutions 3*D* qui contient 15 fois plus de paramètres, et qui opère sur des images de flot optique et de gradient en plus des entrées brutes -cf. la sous-section 2.3.3 du chapitre 2 pour plus de détails-). Par ailleurs, le tableau 7.1 montre que cette première approche donne des résultats stables sur les 5 configurations, contrairement aux observations faites par Gao et al. sur d'autres approches pour lesquelles beaucoup de fluctuations existent [GCHC10].

La deuxième conclusion démontrée par le tableau 7.1 (*ConvNet* 3*D* + BLSTM), est que

		Config.1	Config.2	Config.3	Config.4	Config.5	Moy.
KTH1	*ConvNet 3D* + vote	90,79	90,24	91,42	91,17	91,62	91,04
	ConvNet 3D + BLSTM	92,69	96,55	94,25	93,55	94,93	**94,39**
KTH2	*ConvNet 3D* + vote	89,14	88,55	89,89	89,45	89,97	89,40
	ConvNet 3D + BLSTM	91,50	94,64	90,47	91,31	92,97	**92,17**

TABLE 7.1 – Taux de classification (en %) obtenus par le modèle *ConvNet 3D* sur les bases KTH1 et KTH2. Ces résultats correspondent aux 5 configurations de la validation croisée 5-fold.

notre proposition d'introduire une classification BLSTM pour prendre en compte l'évolution temporelle des caractéristiques apprises, permet d'améliorer les résultats d'environ +3 points, avec 94,39% (respectivement 92,17%) de taux de reconnaissance sur KTH1 (respectivement sur KTH2). Le tableau 7.1 montre également que l'apport des BLSTM est plus important sur la base KTH1, ce qui confirme que ces modèles sont très adaptés pour les longues séquences.

Enfin, nous présentons sur le tableau 7.2 le détail des taux de reconnaissance par classe obtenus par la classification BLSTM entraînée avec les caractéristiques apprises automatiquement par le modèle *ConvNet 3D*. Les meilleurs résultats, aussi bien pour KTH1 que pour KTH2, sont obtenus pour les classes *boxing*, *waving* et *walking*, qui présentent des mouvements très caractéristiques (mouvement des bras pour les deux premières, et vitesse de déplaçement lente pour la troisième). En revanche, les classes *jogging* et *running* présentent une confusion élevée (entre-elles, mais aussi avec la classe *walking*). Certaines confusions existent aussi entre les classes *clapping* et *waving*. En définitive, cette approche permet de classer correctement 94,39% des séquences de la base KTH1 et 92,17% de la base KTH2. Ces résultats seront comparés à l'état de l'art lors de la section 7.5.

7.3.2 Reconnaissance d'expressions faciales

Le modèle *ConvNet 3D* a également été entraîné avec les vidéos de la base GEMEP-FERA. L'architecture utilisée est globalement la même que celle décrite dans la sous-section précédente pour la reconnaissance d'actions humaines, avec toutefois quelques

	Boxing	Clapping	Waving	Walking	Jogging	Running	Moy.
KTH1	98,22	92,66	96,00	98,22	89,80	91,44	**94,39**
KTH2	95,80	94,00	96,60	96,60	88,22	81,80	**92,17**

TABLE 7.2 – Récapitulatif des résultats (taux de reconnaissance par classe, et taux de reconnaissance moyen) obtenus sur les bases KTH1 et KTH2 par le modèle *ConvNet 3D* combiné à classification BLSTM. Ces résultats correspondent aux 5 configurations de la validation croisée 5-fold.

		Colère	Peur	Joie	Soulag.	Tristes.	Moy.
ConvNet 3D + vote	PI	100,0	53,00	80,00	44,00	33,00	62,08
	PS	100,0	100,0	70,00	90,00	80,00	88,00
	Total	**100,0**	**72,0**	**74,00**	**62,00**	**52,00**	**71,94**
ConvNet 3D + BLSTM	PI	85,71	73,33	90,00	62,50	26,66	67,64
	PS	84,61	100,0	90,00	100,0	80,00	90,92
	Total	**85,18**	**84,00**	**90,00**	**76,92**	**48,00**	**76,82**

TABLE 7.3 – Taux de classification (en %) obtenus par le modèle *ConvNet 3D* sur la base GEMEP-FERA d'expressions faciales.

modifications au niveau de la taille des entrées (qui sont de $64 \times 64 \times 9$), de la couche de sortie (qui regroupe maintenant 5 neurones de sortie, correspondant chacun à une classe), et de la taille des noyaux de convolutions 3D C_1, C_2 et C_3 (qui sont respectivement de $9 \times 9 \times 5$, $7 \times 7 \times 3$ et $5 \times 5 \times 3$). Cette architecture correspond à un nombre total de paramètres d'environ $37 \cdot 10^3$.

Comme pour la reconnaissance d'actions humaines, nous avons évalué deux pistes pour la classification des séquences complètes : Une première approche par vote majoritaire, et une deuxième basée sur une classification BLSTM. Pour cette dernière, les séquences de caractéristiques générées par le modèle *ConvNet 3D* contiennent 245 valeurs par instant (ce qui correspond au nombre de sorties au niveau de la couche C_3). L'architecture du classifieur BLSTM est la même que celle utilisée pour la la reconnaissance d'actions humaines, en modifiant les tailles des couches d'entrée et de sortie.

Nous reportons sur le tableau 7.3 les résultats ainsi obtenus. Pour chacune des deux approches, nous présentons les scores *PI* et *PS* (afin d'évaluer leur pouvoirs de généralisation respectifs) ainsi que le score global.

Le modèle *ConvNet 3D* combiné à l'approche de classification par vote majoritaire obtient 71,94% en taux de reconnaissance. Les performances les plus faibles correspondent aux classes *soulagement* et *tristesse*, qui présentent des confusions importantes. A noter que, comme nous l'avons déjà indiqué, ces deux classes sont très similaires dans cette base.

L'approche basée sur la classification BLSTM obtient des résultats globalement

meilleurs que celle basée sur le simple vote (avec un taux de classification de 76, 82%), mais le détail par classe du tableau 7.3 montre que la classe *colère*, qui était reconnue à 100% par le vote majoritaire, ne l'est plus qu'à 85% avec le BLSTM. La raison vient de la proximité des classes *colère* et *peur* (correspondant aux réponses du modèle *ConvNet 3D* aux sous-séquences de 9 images chacune). Le BLSTM apporte une réponse équilibrée pour ces deux classes (85% sur chacune) alors que le vote induit une préférence pour la classe *colère*.

Par ailleurs, il faut noter que l'apport le plus important du classifieur BLSTM concerne le score PI (avec une amélioration de $+5,56$ points). Ceci montre que la classification BLSTM améliore le pouvoir de généralisation du modèle par rapport au simple vote.

7.4 Évaluation des performances du modèle d'auto-encodage parcimonieux

Nous allons nous intéresser dans cette section à l'évaluation des performances du modèle d'auto-encodage parcimonieux (que nous appellerons *AE parcimonieux* dans ce qui suit) décrit dans le chapitre 6, qui permet d'apprendre de manière non supervisée des caractéristiques spatio-temporelles parcimonieuses. Cette évaluation sera faite sur les deux problématiques de la reconnaissance d'actions humaines et de la reconnaissance d'expressions faciales.

7.4.1 Reconnaissance d'actions humaines

Le modèle *AE parcimonieux* opère sur des patchs spatio-temporels de taille $8 \times 8 \times 3$ pixels, qui sont encodés dans un code parcimonieux de taille 192 (respectant ainsi le fait que la représentation soit sur-complète). Les valeurs de η et β utilisées pour la fonction de parcimonie sont quant à elles respectivement fixées à $0,02$ et $1,5$. La recherche du patch translaté optimal s'effectue sur un voisinage $12 \times 12 \times 6$ autour de la position initiale.

Enfin, les caractéristiques parcimonieuses ainsi générées sont utilisées pour entraîner un modèle de classification BLSTM, comme décrit lors du chapitre 6. L'architecture du classifieur BLSTM utilisé est composée de trois couches : (i) Une couche d'entrée de taille 4608 par instant (ce qui correspond à la concaténation de chaque code de taille 192 des patchs d'entrée placés sur la grille des 4×6 localisations possibles sur l'image complète, comme expliqué dans la sous-section 6.5.2 du chapitre 6), (ii) une couche cachée avec

cinq neurones LSTM pour chaque direction, et (iii) une couche de sortie comportant six neurones (un neurone par classe). Cette architecture correspond à un nombre total de paramètres d'environ $180 \cdot 10^3$. A noter que, même si la taille de la couche d'entrée est élevée (avec 4608 valeurs par instant), et par conséquent le nombre de paramètres du BLSTM, seul un faible nombre de connexions sont activées à chaque instant, et ce grâce à la parcimonie des caractéristiques utilisées.

Le protocole d'évaluation adopté pour la base KTH dans le cas des caractéristiques parcimonieuses est le *leave-one-out*. Concrètement, le score final est obtenu en moyennant les taux de reconnaissance individuels relatifs à 25 configurations. Chacune de ces configurations correspond à l'utilisation d'une seule personne pour le test, et des 24 autres pour l'apprentissage. Ce choix est justifié par les recommandations de l'étude faite par Gao et al. dans [GCHC10], et dans laquelle les auteurs ont montré que le protocole *leave-one-out* était celui qui fausse le moins les résultats.

A noter que le modèle *ConvNet 3D* quant à lui n'a pas été évalué en se basant sur ce protocole, mais sur une validation croisée de type 5-fold, pour des raisons de temps d'apprentissage du *ConvNet 3D*. Les résultats présentés lors de la sous-section 7.3.1 ne seront donc pas directement comparables à ceux qui vont être décrits ci-après. Nous consacrerons en revanche le dernier paragraphe de cette sous-section à la comparaison des performances des deux modèles, en ré-évaluant les résultats obtenus par le modèle *ConvNet 3D* sur une partie des configurations du protocole *leave-one-out*.

Le tableau 7.4 présente les résultats obtenus par la classification BLSTM en utilisant les caractéristiques parcimonieuses apprises par le modèle *AE parcimonieux*. Comme évoqué précédemment, ces résultats correspondent aux 25 configurations du protocole *leave-one-out*, sur les bases KTH1 et KTH2.

Ces résultats confirment les observations faites par Gao et al. [GCHC10], à savoir que les performances varient d'une configuration à une autre, avec des fluctuations pouvant aller jusqu'à ± 16 points aussi bien pour KTH1 que pour KTH2. Nous présentons également sur le tableau 7.5 le résultat final correspondant à la moyenne sur les 25 configurations de ces résultats individuels, ainsi que les taux de reconnaissance obtenu pour chaque classe.

Ainsi, les caractéristiques apprises automatiquement par le modèle *AE parcimonieux* et combinées à la classification BLSTM permettent de reconnaitre correctement 95.83% des séquences de la base KTH1, et 93.74% de la base KTH2. Les meilleures performances sont obtenues pour la classe *walking*, à cause de la faible vitesse qui caractérise cette action, et qui est une information très discriminante pour le classifieur BLSTM. Les performances les plus faibles concernent quant à elles les classes *running* et *jogging*, qui

	Config.1	Config.2	Config.3	Config.4	Config.5
KTH1	95,83	83,33	87,50	100,0	95,83
KTH2	94,79	81,25	87,50	93,61	87,15
	Config.6	Config.7	Config.8	Config.9	Config.10
KTH1	100,0	100,0	91,67	100,0	100,0
KTH2	98,96	94,79	93,61	97,92	100,0
	Config.11	Config.12	Config.13	Config.14	Config.15
KTH1	95,83	95,83	91,67	100,0	91,67
KTH2	97,85	93,75	89,58	95,76	90,49
	Config.16	Config.17	Config.18	Config.19	Config.20
KTH1	100,0	100,0	95,83	95,83	100,0
KTH2	92,41	96,80	95,83	92,71	95,83
	Config.21	Config.22	Config.23	Config.24	Config.25
KTH1	95,83	91,67	95,83	95,83	95,83
KTH2	95,35	93,33	95,76	92,71	95,83

TABLE 7.4 – Résultats obtenus sur les bases KTH1 et KTH2 par le modèle *AE parcimonieux* combiné à classification BLSTM pour les 25 configurations du protocole *leave-one-out*.

	Boxing	Clapping	Waving	Walking	Jogging	Running	Moy.
KTH1	98,00	98,00	98,00	99,00	93,00	89,00	**95,83**
KTH2	97,67	94,00	96,98	99,23	89,22	85,36	**93,74**

TABLE 7.5 – Récapitulatif des résultats (taux de reconnaissance par classe, et taux de reconnaissance moyen) obtenus sur les bases KTH1 et KTH2 par le modèle *AE parcimonieux* combiné à classification BLSTM. Ces résultats correspondent au protocole d'évaluation *leave-one-out*.

présentent une forte confusion. Enfin, comme pour le cas de l'approche basée sur le modèle *ConvNet* $3D$, les résultats sur KTH1 sont meilleurs que sur KTH2, confirmant ainsi le fait que la classification BLSTM est mieux adaptée aux longues séquences.

7.4.2 Reconnaissance d'expressions faciales

Comme pour le cas de la reconnaissance d'actions, le modèle *AE parcimonieux* a été entraîné avec des patchs de taille $8 \times 8 \times 3$ extraits à partir des images 64×64 correspondant aux boîtes englobantes centrées sur les visages. Chacun de ces patchs est encodé par un code parcimonieux de taille 128, en utilisant la même architecture et les mêmes paramètres (notamment pour la fonction de parcimonie) que ceux décrits dans la sous-section précédente. La taille du code (de 128 valeurs) pourrait ne pas apparaître comme sur-complète, mais les patchs $3D$ des visages sont très corrélés et cette taille est suffisante (les tests effectués avec 192 valeurs n'ont montré aucune amélioration).

Ainsi, les entrées pour le modèle BLSTM contiennent 8192 valeurs par instant (ce qui correspond à la concaténation de 8×8 codes parcimonieux de taille 128 chacun, voir la

	Colère	Peur	Joie	Soulag.	Tristes.	Moy.
PI	100,0	93.33	95,00	68,75	46,67	80,75
PS	92,31	100,0	100,0	100,0	100,0	98,46
Total	**96,30**	**96,00**	**96,77**	**80,77**	**68,00**	**87,57**

TABLE 7.6 – Taux de classification (en %) obtenus par le modèle *AE parcimonieux* sur la base GEMEP-FERA d'expressions faciales.

Figure 6.11 pour le détail). Le modèle BLSTM utilisé est analogue à celui décrit dans la sous-section précédente, avec des modifications uniquement au niveau des tailles des entrées et des sorties.

Les résultats obtenus (les scores PI, PS ainsi que le score total) sont présentés sur le tableau 7.6. Le taux de reconnaissance global ainsi obtenu est de $87,57\%$, ce qui représente une amélioration de $+10,75$ par rapport aux résultats présentés lors de la sous-section 7.3.2 pour le modèle *ConvNet* $3D$. Cette amélioration des performances concerne aussi bien les scores PI et PS, mais est plus importante pour le score PI qui est amélioré de $+13,11$ points. A noter que quasiment toutes les séquences correspondant à la configuration PS ont été bien reconnues, avec un taux de classification de $98,46\%$. Concrètement, seule une séquence de cette configuration (correspondant à la classe *colère*) a été mal reconnue.

7.5 Comparaison à l'état de l'art

Nous allons présenter dans cette section une comparaison des résultats décrits précédemment par rapport à ceux obtenus par les principaux travaux de l'état de l'art, dans un premier temps sur la base KTH d'actions humaines, puis sur la base GEMEP-FERA d'expressions faciales.

7.5.1 Reconnaissance d'actions humaines

Nous reportons sur le tableau 7.7 un récapitulatif des principaux résultats obtenus sur la base KTH d'actions humaines par les deux modèles neuronaux proposés, ainsi qu'une comparaison avec les meilleurs travaux de l'état de l'art sur cette base.

Pour comparer équitablement les résultats en se plaçant dans les mêmes conditions et en utilisant les mêmes données, selon les recommandations de Gao et al. [GCHC10], les résultats du tableau 7.7 sont regroupés par base (KTH1 ou KTH2), et par protocole d'évaluation (*leave-one-out* ou validation croisée).

A noter que nous faisons aussi la distinction entre les travaux qui se basent sur des caractéristiques apprises et ceux qui se basent sur des caractéristiques manuelles. Ainsi,

Base	Protocole	Caractérist.	Méthode	Résultat
KTH1	Leave-one-out	Apprises	**AE parcimonieux + BLSTM**	**95,83**
		Manuelles	Gao et al. [GCHC10]	96, 33
			Chen et Hauptmann [CH09]	95, 83
			Liu et Shah [LJ08]	94, 20
			Sun et al. [SCH09]	94, 00
			Wong et Cipolla [WC07]	86, 60
			Niebles et al. [NWFF08]	81, 50
	Validation croisée	Apprises	**ConvNet 3D + BLSTM**	**94,39**
			Le et al. [LZYN11]	93, 90
			Jhuang et al. [JSWP07]	91, 70
		Manuelles	Gao et al. [GCHC10]	95, 04
			Schindler et Gool [SVG08]	92, 70
			Rodriguez et al. [RAS08]	88, 70
			Willems et al. [WTVG08]	84, 30
KTH2	Leave-one-out	Apprises	**AE parcimonieux + BLSTM**	**93,74**
			Taylor et al. [TFLB10]	90, 00
		Manuelles	Kim et al. [KLY07]	95, 33
			Ikizler et al. [ICD08]	94, 00
			Ta et al. [TWL+10]	93, 00
	Validation croisée	Apprises	**ConvNet 3D + BLSTM**	**92,17**
			Ji et al. [JXYY10]	90, 20
		Manuelles	Gao et al. [GCHC10]	93, 57
			Laptev et al. [LMSR08]	91, 80
			Dollar et al. [DRCB05]	81, 20

TABLE 7.7 – Récapitulatif des résultats obtenus sur la base KTH d'actions humaines par les modèles *AE parcimonieux* et *ConvNet 3D* combinés à la classification BLSTM, et comparaison avec l'état de l'art.

le tableau 7.7 montre que les deux modèles proposés dans ce manuscrit obtiennent les meilleurs résultats parmi les méthodes de l'état de l'art utilisant des caractéristiques apprises automatiquement [JSWP07, JXYY10, TFLB10].

Plus particulièrement, les caractéristiques apprises de manière non supervisée par le modèle *AE parcimonieux* semblent être plus discriminantes que celles proposées par Taylor et al. dans [TFLB10], et qui se basent sur un modèle RBM entraîné lui aussi de manière non supervisée. De même, l'apprentissage supervisé des caractéristiques profondes par le modèle *ConvNet 3D* aboutit à des meilleurs performances que celles obtenues par les autres modèles d'apprentissage supervisé des caractéristiques, à savoir le modèle neuronal proposé par Ji et al. [JXYY10] (bien que le modèle compte 15 fois plus de paramètres que celui que nous proposons, et n'opère pas sur des données brutes), et le modèle HMAX proposé par Jhuang et al. [JSWP07] (bien que ce dernier soit de nature hybride, vu que les caractéristiques apprises sont générées à partir de caractéristiques bas et moyen niveau conçues manuellement).

Méthode	PI	PS	Total
AE parcimonieux + BLSTM	**80,75**	**98,46**	**87,57**
Yang et Bhanu [YB11]	75, 23	96, 18	83, 78
Tariq et al. [TLL+11]	65, 50	100, 0	79, 80
ConvNet 3D + BLSTM	**67,64**	**90,92**	**76,82**
Littlewort et al. [LWW+11]	71, 40	83, 70	76, 10
Dhall et al. [DAGG11]	64, 80	88, 70	73, 40
Meng et al. [MRPBB11]	60, 90	83, 70	70, 30
Dahmane et Meunier [DM11]	58, 00	87, 00	70, 00
Chew et al. [CLL+11]	62, 00	55, 00	60, 00
Valstar et al. [VJM+11]	44, 00	73, 00	56, 00
Baltrusaitis et al. [BMB+11]	44, 80	43, 30	44, 00

TABLE 7.8 – Récapitulatif des résultats obtenus sur la base GEMEP-FERA d'expressions faciales par les modèles *AE parcimonieux* et *ConvNet 3D* combinés à la classification BLSTM, et comparaison avec les meilleurs résultats obtenus lors du challenge FERA 2011.

Enfin, les deux approches que nous proposons obtiennent des résultats parmi les meilleurs de l'état de l'art, même quand ils sont comparés à des travaux qui se basent sur des caractéristiques manuelles spécifiquement adaptées à cette problématique. En effet, selon la base et le protocole d'évaluation utilisés, les deux modèles proposés obtiennent, à notre connaissance, des résultats parmi les trois meilleurs de l'état de l'art dans chacune des catégories. Ainsi, le modèle *AE parcimonieux* obtient par exemple le second meilleur score sur KTH1, et le troisième sur KTH2 pour le protocole *leave-one-out*.

7.5.2 Reconnaissance d'expressions faciales

Tout comme pour la reconnaissance d'actions humaines, nous présentons sur le tableau 7.8 un récapitulatif des résultats obtenus sur la base GEMEP-FERA d'expressions faciales, ainsi qu'une comparaison avec les meilleurs résultats obtenus sur cette base lors du challenge FERA 2011.

Le modèle *AE parcimonieux* obtient ainsi les meilleures performances sur la base GEMEP-FERA (avec un taux de reconnaissance de 87, 57%). Ce résultat représente une amélioration de +3.79 points par rapport à celui obtenu par Yang et Bhanu [YB11], qui sont les vainqueurs du challenge. Le modèle *ConvNet 3D* obtient quant à lui le quatrième meilleur score, avec un taux de reconnaissance de 76, 82%.

De plus, les résultats obtenus par les deux modèles sont particulièrement bons pour la configuration PI (avec le meilleur score PI pour le modèle *AE parcimonieux*, et le troisième meilleur score PI pour le modèle *ConvNet 3D*), ce qui montre le fort pouvoir de généralisation de ces approches, et le fait que les caractéristiques qu'elles génèrent éli-

	Config.1	Config.2	Config.3	Config.4	Config.5	Moy.
Caractéristiques apprises par le modèle *ConvNet* 3*D*	92,69	96,55	94,25	93,55	94,93	**94,39**
Caractéristiques manuelles [LL03]	84,87	90,64	88,32	90,12	84,95	87,78

TABLE 7.9 – Comparaison des performances obtenues par les caractéristiques apprises par le modèle *ConvNet* 3*D* et les caractéristiques manuelles Coins 3*D* introduites par Laptev et Lindeberg [LL03], combinées à la classification BLSTM. L'évaluation a été faite sur la base KTH2 avec une validation croisée 5-fold.

minent les spécificités liées aux personnes et capturent l'information spatio-temporelle saillante utile à la classification.

7.6 Expérimentations supplémentaires

Après avoir évalué les performances des deux modèles proposés, et les avoir comparé aux principaux travaux de l'état de l'art, nous allons présenter dans cette section quelques résultats complémentaires relatifs à chacun des deux modèles, ainsi qu'une comparaison des performances des deux modèles entre-eux. Toutes ces expérimentations supplémentaires présentées dans ce qui suit seront menées sur la base KTH2.

7.6.1 Modèle *ConvNet* 3*D*

Afin d'évaluer l'intérêt d'utiliser des caractéristiques apprises, nous avons mesuré, sur la base KTH2, les performances de la classification BLSTM combinée à des caractéristiques manuelles, et les avons comparé aux résultats obtenus par les caractéristiques apprises par le modèle *ConvNet* 3*D* (cf. sous-section 7.3.1). Nous avons pour ce faire eu recours au détecteur de "coins 3*D*" de Laptev et Lindeberg [LL03] (qui a été présenté lors du chapitre 2), combiné au descripteur *HOF* calculé autour de chaque point détecté (comme recommandé par Wang et al. pour KTH [WUK+09]).

Pour le détecteur de coins 3*D*, nous avons utilisé l'implémentation originale[3], ainsi que les paramètres standards. Un classifieur BLSTM a ensuite été entraîné en prenant comme entrée des suites temporellement ordonnées de descripteurs *HoF*, selon l'ordre d'apparition des points détectés dans la vidéo.

Les résultats obtenus sont illustrés sur le tableau 7.9, et montrent que ces caractéristiques conçues manuellement donnent de moins bons résultats que celles apprises

[3]Disponible en ligne sur : http://www.di.ens.fr/~laptev/

automatiquement par le modèle *ConvNet* 3*D* (avec une diminution des performances de plus de 4 points), bien que ces caractéristiques aient été conçues spécifiquement pour la reconnaissance d'actions humaines. Ces résultats montrent que l'approche basée sur l'apprentissage à partir d'exemples permet d'aboutir à des caractéristiques plus discriminantes (voir optimales), même en se passant des connaissances a priori relatives à la problématique étudiée.

7.6.2 Modèle *AE parcimonieux*

Nous avons également mené une série d'expérimentations supplémentaires sur le modèle *AE parcimonieux*, que nous présentons dans ce qui suit.

Évaluation de l'apport des différents modules du modèle d'auto-encodage

Au delà des performances globales de l'approche proposée, nous nous sommes aussi intéressés à l'évaluation de l'apport des différentes parties composant le modèle *AE parcimonieux*. Ainsi, nous avons effectué une série d'expérimentations dans lesquelles certains modules sont désactivés afin d'évaluer leurs contributions.

Concrètement, nous avons étudié l'apport de la parcimonie du code (à travers l'apprentissage d'un modèle *AE classique* -non parcimonieux-, sans fonction de parcimonie entre l'encodeur et le décodeur), de l'invariance à la translation (en mesurant les performances du modèle avec et sans la recherche de la translation spatio-temporelle optimale durant l'apprentissage), et de l'aspect temporel des données d'entrée (en remplaçant les convolutions 3*D* par des convolutions 2*D*, et en entraînant le modèle avec des patchs spatiaux -i.e. extraits sur une seule image-). Nous reportons les résultats de ces expérimentations, correspondant aux 5 premières configurations du protocole *leave-one-out*, sur le tableau 7.10.

La deuxième ligne de ce tableau (*AE parci.* 2*D* + inv. à la transl.) montre tout d'abord que les caractéristiques parcimonieuses générées par un modèle entraîné avec des patchs 2*D* au lieu de patchs spatio-temporels sont moins discriminantes entre les classes. Ces caractéristiques 2*D* aboutissent à un taux de reconnaissance de 86,17% sur les 5 premières configurations du protocole *leave-one-out*, soit 2,69 points de moins que l'approche basée sur les caractéristiques spatio-temporelles. Ceci met en évidence l'intérêt de la prise en compte de l'aspect temporel des données dans la représentation qui est apprise par l'auto-encodeur, pour considérer de façon conjointe la forme et son mouvement. A noter que nous avons aussi évalué l'impact de la profondeur temporelle des patchs spatio-temporels (en faisant varier le nombre d'images successives utilisées en

	Config.1	Config.2	Config.3	Config.4	Config.5	Moy.
Schéma complet	94,79	81,25	87,50	93,61	87,15	88,86
AE parci. 2*D* + inv. à la transl.	92,36	79,17	85,42	90,62	83,26	86,17
AE parci. 3*D* + pas d'inv. à la transl.	93,75	80,21	89,58	90,41	82,01	87,19
AE non parci. 3*D* + inv. à la transl.	93,75	77,08	83,33	87,97	84,93	85,41

TABLE 7.10 – Évaluation de l'influence de la temporalité des données d'entrée, de la parcimonie du code, et de l'invariance à la translation sur les performances de l'approche basée sur les caractéristiques apprises par le modèle *AE parcimonieux*, combinées à la classification BLSTM. L'évaluation a été faite sur les 5 premières configurations du protocole *leave-one-out*.

entrée, de 3 à 9 images), et avons obtenu des résultats équivalents ou moins bons (que nous ne présentons pas ici).

De la même manière, les troisième et quatrième lignes du tableau 7.10 (*AE parci.* 3*D* + pas d'inv. à la transl. et *AE non parci.* 3*D* + inv. à la transl.) montrent que l'absence de l'invariance à la translation ou de la parcimonie du code diminuent les performances de l'approche, avec une baisse allant jusqu'à $-3,45$ pour le cas de la parcimonie, justifiant ainsi l'intérêt de ces deux modules.

Invariance aux translations : Comparaison avec l'approche proposée par Ranzato et al.

Comme nous l'avons évoqué précédemment, Ranzato et al. [RHBL07] ont proposé une approche différente de la notre pour gérer l'invariance de la représentation aux translations spatiales, dans le cadre applicatif de la reconnaissance d'objets. Une description détaillée de cette approche ainsi que ses différences avec celle que nous proposons a été faite dans la section 6.2 du chapitre 6.

Afin de comparer les performances des deux approches, nous avons étendu celle introduite par Ranzato et al. [RHBL07] au cas de la vidéo (en remplaçant notamment les convolutions 2*D* par des convolutions 3*D*). Les performances ont été mesurées sur les 5 premières configurations du protocole leave-one-out, et les résultats obtenus (cf. tableau 7.11) montrent que l'approche que nous proposons est la plus performante des deux, avec néanmoins une amélioration qui reste relativement faible d'environ 1,3 points en

	Config.1	Config.2	Config.3	Config.4	Config.5	Moy.
Approche proposée	94,79	81,25	87,50	93,61	87,15	88,86
[RHBL07]	93,75	80,21	86,46	89,17	88,12	87,54

TABLE 7.11 – Comparaison des performances de l'approche proposée pour gérer l'invariance aux translations dans le modèle *AE parcimonieux* avec l'extension de celle introduite par Ranzato et al. [RHBL07] au cas 3*D*.

	Config.1	Config.2	Config.3	Config.4	Config.5	Moy.
AE parcimonieux + BLSTM	94,79	81,25	87,50	93,61	87,15	88,86
ConvNet 3*D* + vote	86,96	73,96	82,29	89,58	81,25	82,81
ConvNet 3*D* + BLSTM	89,41	79,17	85,71	91,11	83,95	85,87

TABLE 7.12 – Comparaison des performances, sur la base KTH2 d'actions humaines, des modèles *ConvNet* 3*D* et *AE parcimonieux* sur les 5 premières configurations du protocole d'évaluation *leave-one-out*.

moyenne.

7.6.3 Comparaison des performances des deux modèles

Enfin, nous avons comparé les performances du modèle *AE parcimonieux* à celles obtenues par le modèle *ConvNet* 3*D*. Vu que les deux approches n'ont pas été évaluées via le même protocole, nous nous sommes placés dans les mêmes conditions afin de pouvoir les comparer directement.

Pour ce faire, nous avons étudié les performances de la classification BLSTM en utilisant les caractéristiques apprises par le modèle *ConvNet* 3*D*, sur les 5 premières configurations du protocole *leave-one-out* de la base KTH2. Nous reportons sur le tableau 7.12 les résultats ainsi obtenus. Nous reportons aussi à titre indicatif les résultats correspondant au vote majoritaire.

Le tableau 7.12 montre que le modèle *AE parcimonieux* donne des meilleurs résultats que le modèle *ConvNet* 3*D*, et ce sur les 5 configurations. Les améliorations sont de $+6,05$ points pour le vote majoritaire, et de $+2,99$ points pour le cas de la classification BLSTM. Ceci est vraisemblablement dû au fait que l'apprentissage supervisé des caractéristiques converge vers un minimum local, ce qui induit des caractéristiques non optimales pour la classification de séquences. Ces résultats montrent donc l'intérêt de séparer la phase de classification de celle de l'extraction des caractéristiques, en

apprenant ces dernières de manière non supervisée.

7.7 Conclusion

Nous avons présenté au cours de ce chapitre les résultats expérimentaux relatifs aux deux modèles d'apprentissage de caractéristiques spatio-temporelles proposés dans le cadre de cette thèse, à savoir le modèle *ConvNet* 3*D* et le modèle *AE parcimonieux*. Les expérimentations ont été effectuées sur des données relatives à deux problématiques différentes : La base KTH d'actions humaines, et la base GEMEP-FERA d'expressions faciales.

Nous avons dans un premier temps évalué les performances du modèle *ConvNet* 3*D*. Nous avons ainsi pu vérifier que, pour les deux problématiques étudiées, l'utilisation des caractéristiques apprises pour entraîner un modèle de classification BLSTM aboutissait à de meilleurs résultats que le simple vote majoritaire sur les décisions individuelles relatives à chaque sous-séquence d'entrée. Nous avons aussi comparé les résultats obtenus avec ces caractéristiques apprises automatiquement sans aucune connaissance a priori du domaine avec ceux correspondant à des caractéristiques conçues manuellement pour être adaptées à l'une des problématiques étudiées, et avons pu démonter que les caractéristiques apprises proposées conduisaient à de meilleures performances, malgré leur nature générique.

Ensuite, nous nous sommes intéressés à l'évaluation des performances du modèle *AE parcimonieux* sur les deux problématiques. Nous avons étudié l'apport des différents modules qui composent le modèle, afin de mettre en évidence l'intérêt de chacun d'entre eux. Nous nous sommes aussi comparés à un autre modèle de l'état de l'art qui propose une approche différente pour gérer l'invariance des caractéristiques apprises aux translations, et avons démontré que notre approche obtenait de meilleurs résultats. Nous avons également démontré que le modèle *AE parcimonieux* était plus performant que le modèle *ConvNet* 3*D*, et ce sur les deux problématiques étudiées. Ceci a permis de confirmer l'intérêt de séparer la phase d'extraction des caractéristiques de celle de la classification (en apprenant les caractéristiques de manière non supervisée).

Enfin, nous avons comparé les résultats obtenus aux principaux travaux de l'état de l'art des deux problématiques étudiées. Les performances des deux modèles pour la reconnaissance d'actions humaines font partie des meilleurs de l'état de l'art sur la base KTH, même en se comparant à des approches basées sur des caractéristiques manuelles (avec 95,83% et 93,74% de bonne classification, respectivement pour KTH1 et KTH2). Pour le cas de la reconnaissance d'expressions faciales, le modèle d'auto-encodage par-

cimonieux obtient les meilleurs résultats de l'état de l'art sur la base GEMEP-FERA (avec 87,57% de bonne classification sur cette base).

Chapitre 8

Conclusion générale

Ce dernier chapitre de conclusion dresse un bilan des travaux effectués dans le cadre de cette thèse. Nous rappellerons dans un premier temps les principales contributions de ces travaux. Nous discuterons ensuite les limitations des approches proposées, ainsi que les améliorations potentielles à apporter. Nous présenterons ensuite quelques pistes de travaux futurs et des perspectives de recherche, ainsi qu'une liste des publications associées aux travaux de cette thèse.

8.1 Récapitulatif des contributions

Nous nous sommes intéressés dans cette thèse à la problématique de la classification automatique des séquences vidéo. L'idée était de se démarquer de la méthodologie dominante qui se base sur l'utilisation de caractéristiques conçues manuellement, en proposant des modèles qui soient les plus génériques possibles et indépendants du domaine. Ceci a été réalisé en automatisant la phase d'extraction des caractéristiques, qui sont dans notre cas générées par apprentissage à partir d'exemples, sans aucune connaissance a priori. Les contributions de cette thèse (qui vont être résumées ci-après) concernent donc principalement la phase d'apprentissage des caractéristiques. Néanmoins, nous nous sommes aussi intéressés à la phase de classification.

En effet, la première contribution de cette thèse est une étude comparative entre plusieurs modèles de classification de séquences parmi les plus populaires de l'état de l'art, à savoir : (i) la recherche des k plus proches voisins, (ii) les champs aléatoires conditionnels cachés, (iii) les machines à vecteurs de support adaptées à la classification de séquences, et (iv) les réseaux de neurones récurrents à longue mémoire à court-terme. Cette étude a été réalisée en se basant sur des caractéristiques manuelles adaptées à la problématique de la reconnaissance d'actions dans les vidéos de football. Concrètement,

les modèles de classification cités ci-dessus ont été entraînés avec des séquences d'histogrammes de sacs de mots visuels calculés sur les vidéos de la base *MICC-Soccer-Actions-4* d'actions de football [BBBS09]. Les résultats obtenus ont permis de sélectionner le modèle de classification neuronal bidirectionnel à longue mémoire à court-terme (BLSTM) comme étant le plus performant, avec un taux de reconnaissance de $79,17\%$, soit entre $+3,01$ et $+26,42$ points de plus que les autres modèles étudiés. Ceci a donc permis de justifier l'utilisation du classifieur BLSTM pour le reste des expérimentations de la thèse.

Nous avons ensuite introduit une nouvelle approche de classification de vidéos de football, qui se base sur des caractéristiques manuelles qui décrivent le mouvement dominant caractérisant une scène donnée (qui se confond avec le mouvement de la caméra pour les actions de sport avec une vue globale du terrain). L'idée est de tirer profit des connaissances a priori introduites par le réalisateur à travers le mouvement de la caméra, et d'exploiter cette information pour la classification. Pour ce faire, le mouvement dominant est estimé à partir d'un appariement entre les points SIFT détectés sur deux images successives de la vidéo. Nous avons démontré que la combinaison de ces caractéristiques avec les sacs de mots visuels obtenait les meilleurs résultats de l'état de l'art sur la base *MICC-Soccer-Actions-4*, avec $93,98\%$ de taux de reconnaissance. Ceci a permis de vérifier que les modèles BLSTMs sont particulièrement discriminants quand ils sont entraînés avec des caractéristiques qui sont bien représentatives du contenu des séquences vidéo. Ceci est généralement le cas des caractéristiques manuelles, qui sont choisies empiriquement de manière à être très adaptées aux contenus qu'elles décrivent. Elles ont néanmoins l'inconvénient d'être liées à un domaine donné (celui pour lequel elles ont été conçues), et sont très peu génériques.

Nous avons donc proposé une alternative à ce choix empirique des caractéristiques manuelles, en étudiant la possibilité d'apprendre automatiquement des caractéristiques à partir d'exemples. Nous avons pour ce faire exploré deux pistes, à savoir l'apprentissage supervisé et non supervisé. La troisième contribution principale de cette thèse a donc été de proposer un modèle neuronal d'apprentissage supervisé des caractéristiques spatio-temporelles, qui étend le principe des modèles *ConvNets* [LBBH98, LKF10] au cas de la vidéo. Le modèle *ConvNet 3D* que nous avons proposé opère sur des volumes spatio-temporels (des suites d'images successives d'une vidéo) est construit une représentation hiérarchique de ces entrées, allant du bas-niveau (les données brutes) jusqu'au haut-niveau (les classes), à travers l'utilisation d'une architecture multi-couches (un principe appelé *apprentissage profond*). L'idée clé des *ConvNet 3D* est l'utilisation de convolutions tridimensionnelles, dont le noyau (c'est à dire l'ensemble de paramètres qui lui sont associés) est calculé et mis à jour lors de l'apprentissage. Ces convolutions

permettent de capturer les motifs spatio-temporels saillants contenus dans leurs entrées, et de les utiliser pour la classification. Une fois l'apprentissage terminé, les sorties obtenues au niveau des couches cachées peuvent être vues comme une représentation intermédiaire du contenu spatio-temporel saillant des entrées. Nous avons donc proposé de les utiliser comme caractéristiques pour entraîner un modèle BLSTM à classer les séquences vidéo.

Nous nous sommes ensuite intéressés à l'apprentissage non supervisé de caractéristiques, afin de séparer complètement cette étape de celle de la classification. La quatrième contribution principale de cette thèse est donc d'avoir introduit un modèle d'auto-encodage qui permet d'apprendre, de manière non supervisée, une représentation parcimonieuse sur-complète des données d'entrée (volumes spatio-temporels), c'est à dire dont la dimension est équivalente à celle des entrées, mais où seul un faible nombre de valeurs sont non nulles. Nous nous sommes pour ce faire inspirés des travaux de Ranzato et al. [RPCL06, RHBL07] sur la reconnaissance d'objets dans les images fixes, en proposant un modèle similaire adapté à la classification vidéo. Concrètement, le modèle est composé d'un encodeur qui projette l'entrée spatio-temporelle dans un espace de représentation, et d'un décodeur qui la reconstruit à partir des coordonnées de la projection (appelées "code"). Une fonction de parcimonie est placée entre l'encodeur et le décodeur afin d'assurer la parcimonie du code, et une procédure spécifique, différente de celle décrite Ranzato et al. [RPCL06, RHBL07], a été introduite afin de gérer l'invariance aux translations de la représentation apprise. Le modèle est entraîné avec des patchs spatio-temporels, afin de réduire la diversité du contenu à encoder. Nous avons ainsi introduit une architecture neuronale d'auto-encodage parcimonieux (que nous avons baptisé *AE parcimonieux*), et un algorithme d'apprentissage correspondant basé sur la minimisation d'une fonction objectif globale. Une fois l'apprentissage effectué, le code parcimonieux appris est utilisé comme caractéristiques pour entraîner un modèle BLSTM à classer les séquences vidéo.

Afin de valider leur généricité, les deux approches proposées ont été évaluées sur deux problématiques différentes, à savoir la reconnaissance d'actions humaines (sur la base KTH [SLC04]), et la reconnaissance d'expressions faciales (sur la base GEMEP-FERA [VJM+11]). Ceci nous a permis de vérifier dans un premier temps que les caractéristiques apprises de manière non supervisée par le modèle *AE parcimonieux* étaient plus discriminantes que celles générées par le modèle *ConvNet 3D*. Nous avons aussi pu démonter, pour le cas du modèle *AE parcimonieux*, l'apport de la parcimonie du code et de l'invariance aux translations. Enfin, nous avons comparé les performances des deux modèles aux meilleures de l'état de l'art pour les deux problématiques étudiées. Pour

le cas de la reconnaissance d'actions, les résultats obtenus font partie des meilleurs sur la base KTH, même en se comparant à des approches basées sur des caractéristiques manuelles, puisque le modèle *AE parcimonieux* permet d'obtenir un taux de bonne classification de 95,83%. Les résultats sont encore meilleurs pour le cas de la reconnaissance d'expressions faciales, vu que les performances des modèles *ConvNet 3D* et *AE parcimonieux* se placent respectivement à la quatrième et à la première place du classement du challenge pour lequel la base a été proposée, avec des taux de reconnaissance respectifs de 76,82% et 87,57%. Ces différents résultats ont ainsi permis de valider les deux approches proposées.

8.2 Discussion sur les limitations des approches proposées

Les différentes approches proposées dans le cadre de cette thèse présentent un certains nombre de limitations, qui vont être discutées dans cette section.

Tout d'abord, lors de l'apprentissage du modèle BLSTM avec l'algorithme de rétropropagation dans le temps (BPTT), les séquences cibles utilisées contiennent la même valeur (qui est l'indice de la classe) à tous les instants. Or, dans les problématiques abordées, d'une part seule une partie de la séquence contient une information pertinente correspondant à la classe (certaines parties pouvant même être vides comme dans la base KTH), et d'autres part certaines parties peuvent être identiques entre deux classes différentes ou correspondre à plusieurs sous-classes (par exemple, les différents mouvements élémentaires composant une action humaine). Même si, comme nous l'avons vu, le contexte bidirectionnel des réseaux BLSTM permet d'apprendre avec cette cible unique, le fait d'entraîner ce modèle à attribuer le même label à tous les instants peut donc diminuer les performances de la classification. Nous allons proposer dans la sous-section 8.3.2 une piste d'amélioration possible qui pourrait permettre de surmonter cette limitation.

Enfin, nous avons jusque là vérifié la généricité des différents modèles proposés en les évaluant sur deux problématiques uniquement, et sur des données moins "réalistes" que celles utilisées pour des problématiques de classification vidéo dans un contexte industriel. Afin de valider définitivement la généricité de ces approches, il faudrait évaluer leurs performances sur d'autres problématiques, et d'autres données plus complexes.

8.3 Travaux futurs

Plusieurs pistes de travaux futurs existent à l'issue de cette thèse. Nous allons dans cette section en décrire quelques unes, dont certaines pour lesquelles nous présentons également un début de résultat.

8.3.1 Application à d'autres données / problématiques

La première piste envisagée est l'application des approches proposés à des données plus récentes et plus complexes, toujours dans l'optique de vérifier leurs généricité. Par exemple, pour le cas de la reconnaissance d'actions, même si la base KTH reste celle la plus utilisée actuellement dans l'état de l'art, de plus en plus de travaux s'intéressent à des nouvelles bases contenant des actions plus réalistes. Nous pouvons par exemple citer la base UCF d'actions de sport [RAS08], ou encore les bases Hollywood-2 [MLS09], YouTube [LLS09], UT-Interaction [RA09] et LIRIS [WML+12] d'actions humaines. Les approches de l'état de l'art (majoritairement basées sur des caractéristiques manuelles) obtiennent des résultats très variables d'une base à une autre. L'apport de l'apprentissage automatique pourrait être très important dans ce cas et permettre d'obtenir des performances plus stables. La difficulté dans ce cas sera de modéliser directement des actions complexes et de longue durée (comme par exemple une discussion entre plusieurs personnes, ou encore une personne qui abandonne un bagage). Une piste possible pour remédier à cette limitation serait de combiner une extraction des caractéristiques sur des petits volumes spatio-temporels, à une modélisation structurée (par exemple avec des graphes) sur les sorties du modèle neuronal.

Nous avons aussi étudié la possibilité d'appliquer les approches proposées à un cas "réel" d'utilisation, qui présente un fort potentiel applicatif et commercial pour Orange Labs. Nous avons choisi pour ce faire le cadre de la reconnaissance d'actions de sport, et plus particulièrement le cas du rugby. L'idée est de pouvoir ainsi proposer une solution de classification de segments vidéo de rugby, qui peut être utilisée pour des applications diverses telles que l'indexation, la navigation intra-programme ou encore les résumés automatiques.

Nous avons ainsi généré un corpus de vidéos correspondant à l'édition 2011 de la coupe du monde. La base contient 96 vidéos d'environ 45 minutes chacune (une vidéo par mi-temps) correspondant à trois chaînes différentes (France 2, France 3 et TF1), à une vingtaine d'équipes, et à douze stades différents. Les vidéos ont une résolution de 512×288 pixels, et sont échantillonnés à 25 images par seconde. Toutes les vidéos ont subi une segmentation en plans, et la reconnaissance des actions est faite sur chacun

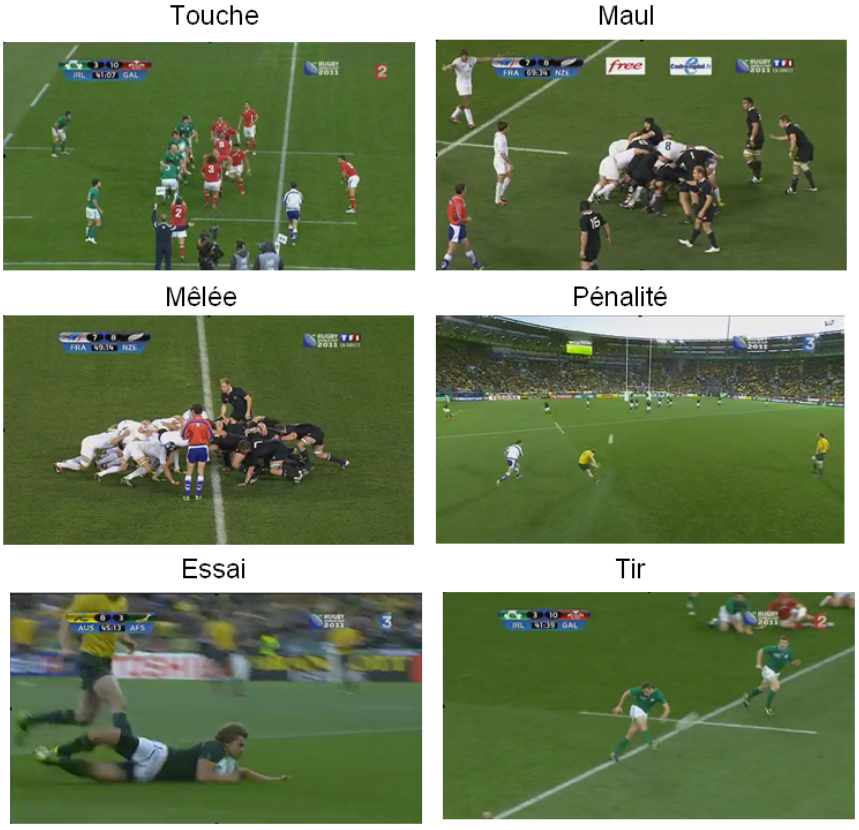

FIGURE 8.1 – Les six actions représentées dans la base "coupe du monde de rugby 2011".

des segments. A titre d'information, le nombre de segments pour chacune des vidéos est d'environ 450. La base comporte six types d'actions : touche, maul, mêlée, pénalité, essai, et tir (cf. Figure 8.1). Les 80 vidéos correspondant à la phase du premier tour sont utilisées pour l'apprentissage, et les 16 restantes pour le test.

Nous avons évalué sur cette base les performances du modèle *AE parcimonieux* combiné à la classification BLSTM. L'auto-encodeur est entraîné avec des patchs $12 \times 12 \times 3$ et encodés dans un code de taille 300. Le réseau BLSTM contient quant à lui 5 neurones dans chaque direction. Les premières expérimentations font état d'un taux de classification de 78% sur la base de test, ce qui représente un résultat plutôt satisfaisant au vue de l'annotation et de la qualité des segments. Nous envisageons de continuer les expérimentations sur cette base afin d'améliorer encore ce résultat. L'une des difficultés à surmonter concerne notamment la représentativité des classes pour l'apprentissage, qui est très déséquilibrée (la classe "essai" par exemple est beaucoup moins représentée que les autres) Nous envisageons également d'évaluer les performances du modèle *ConvNet* $3D$ sur cette base.

8.3.2 Classification temporelle connexionniste

Comme nous l'avons évoqué dans la section 8.2, la principale limitation de la classification BLSTM est qu'elle se base sur l'attribution de la même sortie cible (qui est celle correspondant à la classe) à tous les instants de la séquence, ce qui peut augmenter la confusion entre les classes. La classification temporelle connexionniste (CTC) a été introduite par Graves et al. [GFGS06] afin de remédier à ce problème. De manière schématique et sans renter dans les détails, la CTC peut être vue comme un module supplémentaire placé au niveau de la couche de sortie d'un classifieur BLSTM (ou celle d'un réseau de neurones récurrent en général), qui permet d'entraîner celui-ci avec des séquences de labels différents, au lieu d'affecter un même label à tous les instants, et ceci sans avoir à annoter les instants d'apparition de chaque label. Concrètement, la CTC permet, à partir des sorties du BLSTM et de la séquence de labels cible, de calculer les séquences de vecteurs d'erreurs à rétro-propager. Ce passage d'une séquence de labels cibles au vecteur d'erreurs se fait en utilisant une procédure inspirée de l'algorithme *backward-forward* des modèles de Markov cachés. Pour plus de détails, se référer à l'article de Graves et al. [GFGS06].

Dans le cadre du stage de fin d'études de Q. Lu [Lu12] (qui a été co-encadré par F. Mamalet et moi-même), nous nous sommes intéressés à l'utilisation de la CTC pour la reconnaissance d'actions humaines sur la base KTH. L'idée est de décomposer chaque action en un certains nombres de sous-actions (analogues à celles décrites dans la sous-section 5.3.2 du chapitre 5), et d'entraîner le modèle BLSTM-CTC, avec les caractéristiques apprises automatiquement par les modèles introduits dans le cadre de cette thèse, à localiser temporellement ces sous-actions. La Figure 8.2 montre un exemple de résultat obtenu sur une séquence de la base KTH correspondant à la classe *Waving*. Les trois couleurs utilisées correspondent aux trois sous-actions détectées.

Une piste de travaux futurs consiste à exploiter cette information de la localisation des sous-classes détectées pour améliorer les performances de la classification. Nous pouvons par exemple envisager d'entraîner le modèle *ConvNet 3D* à reconnaître ces sous-classes, afin de générer des caractéristiques plus représentatives du contenu de la vidéo (en comparaison à celles obtenues en ciblant la même classe pour tous les segments de 9 images successives).

8.3.3 Autres pistes

Outre les deux pistes présentées précédemment, plusieurs autres sont envisageables. Pour l'approche d'auto-encodage parcimonieux, l'une des pistes envisageables est de

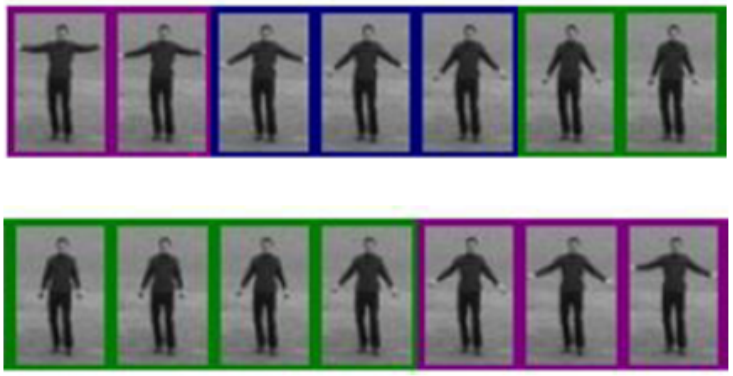

FIGURE 8.2 – Exemple de résultat présenté dans [Lu12] : Localisation temporelle de trois sous-actions correspondant à la classe *Waving* de la base KTH.

gérer l'invariance aux changements d'échelle, en plus de celle aux translations. Ceci peut être fait en rajoutant une variable cachée supplémentaire à la fonction objectif globale décrite dans le chapitre 6, et en effectuant une recherche exhaustive sur un certains nombre de facteurs de zooms, puis en sélectionnant la valeur optimale, de la même manière que pour le cas des translations.

Enfin, il serait intéressant d'étudier la possibilité de proposer un modèle "hybride", qui combine l'apprentissage supervisé et non supervisé. En effet, de nombreux travaux dans le domaine de la reconnaissance d'objets ont démontré l'intérêt de pré-entraîner de manière non supervisée les couches inférieures d'un modèle profond, et d'utiliser ce modèle "intermédiaire" pour initialiser un apprentissage supervisé (un procédé communément appelé *fine-tuning* dans la littérature). Les travaux de Hinton et al. [HOT06] sur les modèles RBMs ou encore ceux de Ranzato et al. [RPCL06, RHBL07] et de Bengio et al. [BLPL07] sur les modèles neuronaux ont permis de démontrer que cette procédure permet d'améliorer considérablement les performances, ainsi que la vitesse de convergence, dans les cas $1D$ et $2D$. Une extension directe de ce principe pour notre cas consisterait à initialiser les noyaux de convolutions d'un modèle *ConvNet* $3D$ avec les paramètres appris de manière non supervisée par le modèle *AE parcimonieux*.

8.4 Liste des publications relatives à cette thèse

Conférences internationales avec comité de lecture et actes

- Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia et Atilla Baskurt. *Spatio-Temporal Convolutional Sparse Auto-Encoder for Sequence Classification*. Dans British Machine Vision Conference (BMVC), 2012. Présentation orale.

- Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia et Atilla Baskurt. *Sparse Shift-Invariant Representation of Local 2D Patterns and Sequence Learning for Human Action Recognition*. Dans International Conference on Pattern Recognition (ICPR), 2012. Présentation orale.

- Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia et Atilla Baskurt. *Sequential Deep Learning for Human Action Recognition*. Dans International Workshop on Human Behavior Understanding (HBU), 2011. Présentation orale.

- Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia et Atilla Baskurt. *Action Classifcation in Soccer Videos with Long Short-Term Memory Recurrent Neural Networks*. Dans International Conference on Artificial Neural Networks (ICANN), 2010. Présentation orale.

Conférences nationales avec comité de lecture et actes

- Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia et Atilla Baskurt. *Une approche neuronale pour la classification d'actions de sport par la prise en compte du contenu visuel et du mouvement dominant*. Dans Compression et Représentation des Signaux Audiovisuels (CORESA), 2010. Présentation orale.

Rapports de recherche

- Christian Wolf, Julien Mille, Eric Lombardi, Oya Celiktutan, Mingyuan Jiu, Moez Baccouche, Emmanuel Dellandréa, Charles-Edmond Bichot, Christophe Garcia et Bülent Sankur. *The LIRIS Human activities dataset and the ICPR 2012 human activities recognition and localization competition*. Rapport de recherche RR-LIRIS-2012-004, Laboratoire d'Informatique en Images et Systèmes d'Information (LIRIS), INSA de Lyon, 2012.

Bibliographie

[ABCDB02] J. Assfalg, M. Bertini, C. Colombo et A. Del Bimbo : Semantic annotation of sports videos. *IEEE Multimedia*, 9(2):52–60, 2002. xv, 26, 27

[AEB05] M. Aharon, M. Elad et A. Bruckstein : K-SVD and its non-negative variant for dictionary design. *In Optics and Photonics*, volume 5914, 2005. 30, 98

[AHP04] T. Ahonen, A. Hadid et M. Pietikäinen : Face recognition with local binary patterns. *European Conference on Computer Vision*, pages 469–481, 2004. 20

[Bau72] L.E. Baum : An equality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. *Inequalities*, 3:1–8, 1972. 45

[BBBS09] L. Ballan, M. Bertini, A.D. Bimbo et G. Serra : Action categorization in soccer videos using string kernels. *In IEEE International Workshop on Content-Based Multimedia Indexing*, pages 13–18, 2009. xv, xvi, xvii, xix, 5, 28, 53, 64, 66, 67, 72, 73, 74, 75, 76, 78, 79, 80, 140

[BBDBS10] L. Ballan, M. Bertini, A. Del Bimbo et G. Serra : Video event classification using string kernels. *Multimedia Tools and Applications*, 48(1):69–87, 2010. 53

[BD96] A. Bobick et J. Davis : An appearance-based representation of action. *In International Conference on Pattern Recognition*, volume 1, pages 307–312, 1996. 19, 23

[BD01] A.F. Bobick et J.W. Davis : The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001. 19, 23

[BE67] L.E. Baum et J.A. Eagon : An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. *Bull. Amer. Math. Soc*, 73(3):360–363, 1967. 45

[BGX09] M. Bregonzio, S. Gong et T. Xiang : Recognising action as clouds of space-time interest points. *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 1948–1955, 2009. 17

[BHB02] C. Bahlmann, B. Haasdonk et H. Burkhardt : Online handwriting recognition with support vector machines-a kernel approach. *In IEEE International Workshop on Frontiers in Handwriting Recognition*, pages 49–54, 2002. 53

[Bis06] C.M. Bishop : *Pattern recognition and machine learning*, volume 4. Springer, New York, 2006. 34

[BJM83] L.R. Bahl, F. Jelinek et R.L. Mercer : A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):179–190, 1983. 45

[BK00] I. Bazzi et D. Katabi : Using support vector machines for spoken digit recognition. *International Conference on Spoken Language Processing*, 20:48, 2000. 52

[BKJ+05] T. Bae, C. Kim, S. Jin, K. Kim et Y. Ro : Semantic event detection in structured video using hybrid HMM/SVM. *Image and Video Retrieval*, 3568:595–595, 2005. 52

[BKK02] N. Babaguchi, Y. Kawai et T. Kitahashi : Event based indexing of broadcasted sports video by intermodal collaboration. *IEEE Transactions on Multimedia*, 4(1):68–75, 2002. 25

[BL88] D.S. Broomhead et D. Lowe : Radial basis functions, multi-variable functional interpolation and adaptive networks. Rapport technique, DTIC Document, 1988. 34

[BL97] J.S. Beis et D.G. Lowe : Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. *In IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1000–1006, 1997. 68

[BLB+02] M.S. Bartlett, G. Littlewort, B. Braathen, T.J. Sejnowski et J.R. Movellan : A prototype for automatic recognition of spontaneous facial actions. *Advances in neural information processing systems*, 15:1271–1278, 2002. 23

[BLPL07] Y. Bengio, P. Lamblin, D. Popovici et H. Larochelle : Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153, 2007. 146

[BMB+11] T. Baltrusaitis, D. McDuff, N. Banda, M. Mahmoud, R. El Kaliouby, P. Robinson et R. Picard : Real-time inference of mental states from facial expressions and upper body gestures. *In IEEE International Conference on Automatic Face & Gesture Recognition*, pages 909–914, 2011. 132

[BP66] L.E. Baum et T. Petrie : Statistical inference for probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563, 1966. 45

[Bri90] J. S. Bridle : Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. *In Neurocomputing*, pages 227–236. Springer, 1990. 102

[BSF94] Y. Bengio, P. Simard et P. Frasconi : Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2): 157–166, 1994. 57

[BTVG06] H. Bay, T. Tuytelaars et L. Van Gool : SURF : Speeded up robust features. *European Conference on Computer Vision*, pages 404–417, 2006. 13, 15

[BW98] J.S. Boreczky et L.D. Wilcox : A hidden Markov model framework for video segmentation using audio and image features. *In IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 6, pages 3741–3744, 1998. 46

[CC94] T. Cox et M. Cox : *Multidimensional scaling*. Chapman and Hill, 1994. 30

[CET01] T.F. Cootes, G.J. Edwards et C.J. Taylor : Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001. 23

[CH09] M. Y. Chen et A. Hauptmann : MoSIFT : Recognizing human actions in surveillance videos. Rapport technique CMU-CS-09-161, School of Computer Science, Carnegie Mellon University, Pittsburgh PA 15213, September 2009. xv, 16, 131

[CHFT06] Y. Chang, C. Hu, R. Feris et M. Turk : Manifold based analysis of facial expression. *Image and Vision Computing*, 24(6):605–614, 2006. 24

[CL11] C.C. Chang et C.J. Lin : LIBSVM : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27, 2011. 72

[CLL+11] S.W. Chew, P. Lucey, S. Lucey, J. Saragih, J.F. Cohn et S. Sridharan : Person-independent facial expression detection using constrained local models. *In IEEE International Conference on Automatic Face & Gesture Recognition*, pages 915–920, 2011. 132

[CO09] C. Cadieu et B. Olshausen : Learning transformational invariants from natural movies. *Neural Information Processing Systems*, pages 209–216, 2009. 41

[Cov65] T.M. Cover : Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, 14(3):326–334, 1965. 50

[CPS06] K. Chellapilla, S. Puri et P. Simard : High performance convolutional neural networks for document processing. *In International Workshop on Frontiers in Handwriting Recognition*, 2006. 39

[CRA+04] J.F. Cohn, L.I. Reed, Z. Ambadar, J. Xiao et T. Moriyama : Automatic analysis and recognition of brow actions and head motion in spontaneous facial behavior. *In IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 610–616, 2004. 24

[CS01] S. Carlsson et J. Sullivan : Action recognition by shape matching to key frames. *In Workshop on Models versus Exemplars in Computer Vision*, volume 1, 2001. 18

[CSB01] D. Chen, K. Shearer et H. Bourlard : Text enhancement with asymmetric filter for video OCR. *In International Conference on Image Analysis and Processing*, pages 192–197. IEEE, 2001. 21

[CV95] C. Cortes et V. Vapnik : Support-vector networks. *Machine learning*, 20(3): 273–297, 1995. 50

[CYC08] N. Cuntoor, B. Yegnanarayana et R. Chellappa : Activity modeling using event probability sequences. *IEEE Transactions on Image Processing*, 17(4):594–607, 2008. 46

[DA09] T.M.T. Do et T. Artières : Large margin training for hidden markov models with partially observed states. *In International Conference on Machine Learning*, pages 265–272, 2009. 46

[DAGG11] A. Dhall, A. Asthana, R. Goecke et T. Gedeon : Emotion recognition using PHOG and LPQ features. *In IEEE International Conference on Automatic Face & Gesture Recognition*, pages 878–883, 2011. 132

[Dau85] J.G. Daugman : Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Optics and Image Science*, 2:1160–1169, 1985. 21

[Del06] E. Delakis : *Structuration multimodale des vidéos de tennis en utilisant des modèles segmentaux*. Thèse de doctorat, Université de Rennes 1, 2006. 59

[DG05] S. Duffner et C. Garcia : A connexionist approach for robust and precise facial feature detection in complex scenes. *In IEEE International Symposium on Image and Signal Processing and Analysis*, pages 316–321, 2005. 24, 39

[DG06] S. Duffner et C. Garcia : A neural scheme for robust detection of transparent logos in tv programs. *International Conference on Artificial Neural Networks*, pages 14–23, 2006. 39

[DG08a] M. Delakis et C. Garcia : Text detection with convolutional neural networks. *In International Conference on Computer Vision Theory and Applications*, volume 2, pages 290–294, 2008. 39

[DG08b] S. Duffner et C. Garcia : Robust face alignment using convolutional neural networks. *In International Conference on Computer Vision Theory and Applications*, 2008. 122

[DK05] K.B. Duan et S. Keerthi : Which is the best multiclass SVM method ? An empirical study. *Multiple Classifier Systems*, 3541:732–760, 2005. 51

[DKRD03] R. Dahyot, A. Kokaram, N. Rea et H. Denman : Joint audio visual retrieval for tennis broadcasts. *In IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages III–561, 2003. 25

[DM96] D. DeCarlo et D. Metaxas : The integration of optical flow and deformable models with applications to human face shape and motion estimation. *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 231–238, 1996. 24

[DM11] M. Dahmane et J. Meunier : Emotion recognition using dynamic grid-based HoG features. *In IEEE International Conference on Automatic Face & Gesture Recognition*, pages 884–888, 2011. 132

[DRCB05] P. Dollár, V. Rabaud, G. Cottrell et S. Belongie : Behavior recognition via sparse spatio-temporal features. *In IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72, 2005. xv, 14, 15, 18, 21, 52, 131

[Duf07] S. Duffner : *Face Image Analysis with Convolutional Neural Networks*. Thèse de doctorat, Albert-Ludwigs-University Freiburg, 2007. 39

[DWC09] T. Dean, R. Washington et G. Corrado : Recursive sparse spatiotemporal coding. *In IEEE International Symposium on Multimedia*, pages 645–650. IEEE, 2009. 41

[EGS11] K. Elagouni, C. Garcia et P. Sébillot : A comprehensive neural-based approach for text recognition in videos using natural language processing. *In ACM International Conference on Multimedia Retrieval*, page 23, 2011. 39

[Elm90] J.L. Elman : Finding structure in time. *Cognitive science*, 14(2):179–211, 1990. 55

[EP97] I.A. Essa et A.P. Pentland : Coding, analysis, interpretation, and recognition of facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):757–763, 1997. 23

[ES02] D. Eck et J. Schmidhuber : Learning the long-term structure of the blues. *International Conference on Artificial Neural Networks*, 2415:796–796, 2002. 59

[ETM03] A. Ekin, A.M. Tekalp et R. Mehrotra : Automatic soccer video analysis and summarization. *IEEE Transactions on Image Processing*, 12(7):796–807, 2003. 26

[Fas06] I. R. Fasel : *Learning Real-Time Object Detectors : Probabilistic Generative Approaches*. Thèse de doctorat, Department of Cognitive Science, University of California, San Diego, USA, 2006. 24

[FBL09] S. Fidler, M. Boben et A. Leonardis : *Object Categorization : Computer and Human Vision Perspectives, chapitre : Learning Hierarchical Compositional Representations of Object Structure*. Cambridge University Press, 2009. 30

[FCA+09] A. Frome, G. Cheung, A. Abdulkader, M. Zennaro, B. Wu, A. Bissacco, H. Adam, H. Neven et L. Vincent : Large-scale privacy protection in google street view. *In IEEE International Conference on Computer Vision*, pages 2373–2380, 2009. 39

[FGS07] S. Fernández, A. Graves et J. Schmidhuber : An application of recurrent neural networks to discriminative keyword spotting. *International Conference on Artificial Neural Networks*, 4669:220–229, 2007. 59

[FHP04] X. Feng, A. Hadid et M. Pietikäinen : A coarse-to-fine classification scheme for facial expression recognition. *Image Analysis and Recognition*, pages 668–675, 2004. 20

[Fis81] M. Fischler : Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 69

[FL03] B. Fasel et J. Luettin : Automatic facial expression analysis : a survey. *Pattern Recognition*, 36(1):259–275, 2003. 22

[Föl90] P. Földiák : Forming sparse representations by local anti-hebbian learning. *Biological cybernetics*, 64(2):165–170, 1990. 102

[Fuk80] K. Fukushima : Neocognitron : A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980. 37

[FXWG10] J. Fan, W. Xu, Y. Wu et Y. Gong : Human tracking using convolutional neural networks. *IEEE Transactions on Neural Networks*, 21(10):1610–1623, 2010. 39

[GBTG02] S.B. Gokturk, J.Y. Bouguet, C. Tomasi et B. Girod : Model-based face tracking for view-independent facial expression recognition. *In IEEE International Conference on Automatic Face and Gesture Recognition*, pages 287–293, 2002. 24

[GCHC10] Z. Gao, M.Y. Chen, A. Hauptmann et A. Cai : Comparing evaluation protocols on the kth dataset. *In Human Behavior Understanding*, volume 6219, pages 88–100. Springer, 2010. 119, 120, 124, 128, 130, 131

[GD04] C. Garcia et M. Delakis : Convolutional Face Finder : A neural architecture for fast and robust face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1408–1423, 2004. 39, 89, 122

[GD05] G. Guo et C.R. Dyer : Learning from examples in the small sample case : face expression recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 35(3):477–488, 2005. 23

[Ger01] F. Gers : *Long Short-Term Memory in Recurrent Neural Networks*. Thèse de doctorat, Ecole Polytechnique Fédérale de Lausanne, 2001. xvi, 58, 59, 73, 80

[GFGS06] A. Graves, S. Fernández, F. Gomez et J. Schmidhuber : Connectionist Temporal Classification : labelling unsegmented sequence data with recurrent neural networks. *In International conference on Machine learning*, pages 369–376, 2006. 145

[GFL+08] A. Graves, S. Fernández, M. Liwicki, H. Bunke et J. Schmidhuber : Unconstrained online handwriting recognition with recurrent neural networks. *Advances in Neural Information Processing Systems*, 20:1–8, 2008. 59

[GHP00] A. Ganapathiraju, J. Hamaker et J. Picone : Hybrid SVM/HMM architectures for speech recognition. *In International Conference on Spoken Language Processing*, 2000. 52

[GJ05] H. Gu et Q. Ji : Information extraction from image sequences of real-world facial expressions. *Machine Vision and Applications*, 16(2):105–115, 2005. 25

[GLC95] Y. Gong, TS Lim et HC Chua : Automatic Parsing of TV Soccer Programs. *In IEEE International Conference on Multimedia Computing and Systems*, pages 167–174, 1995. 27

[GMAP05] A. Gunawardana, M. Mahajan, A. Acero et J.C. Platt : Hidden Conditional Random Fields for phone classification. *In Interspeech*, volume 2, 2005. 48, 49

[Gra08] A. Graves : *Supervised Sequence Labelling with Recurrent Neural Networks*. Thèse de doctorat, Technischen Universitat Munchen, Fakultat fur Informatik, 2008. 58, 73

[GRHS04] J. Goldberger, S. Roweis, G. Hinton et R. Salakhutdinov : Neighbourhood components analysis. *In Advances in Neural Information Processing Systems*, 2004. 29

[GS01] F.A. Gers et E. Schmidhuber : LSTM recurrent networks learn simple context-free and context-sensitive languages. *IEEE Transactions on Neural Networks*, 12(6):1333–1340, 2001. 59

[GS05] A. Graves et J. Schmidhuber : Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005. 59, 80

[GS09] A. Graves et J. Schmidhuber : Offline handwriting recognition with multidimensional recurrent neural networks. *Advances in Neural Information Processing Systems*, 21:545–552, 2009. 59, 80

[GSS03] F.A. Gers, N.N. Schraudolph et J. Schmidhuber : Learning precise timing with LSTM recurrent networks. *The Journal of Machine Learning Research*, 3:115–143, 2003. 59

[Gue02] A. Gueziec : Tracking pitches for broadcast television. *Computer*, 35(3):38–43, 2002. 27

[HAH01] X. Huang, A. Acero et H.W. Hon : *Spoken Language Processing : A Guide to Theory, Algorithm, and System Development*. Prentice Hall PTR New Jersey, 2001. 46

[Hay99] S. Haykin : *Neural Networks : A Comprehensive Foundation*. Pearson Education, 1999. 34

[HBF01] S. Hochreiter, Y. Bengio et P. Frasconi : Gradient flow in recurrent nets : the difficulty of learning long-term dependencies. *A Field Guide to Dynamical Recurrent Neural Networks*, 2001. 57

[HCL06] R. Hadsell, S. Chopra et Y. LeCun : Dimensionality reduction by learning an invariant mapping. *In IEEE conference on computer vision and pattern recognition*, volume 2, pages 1735–1742, 2006. 30

[Hes10] R. Hess : An open-source SIFT Library. *In International conference on Multimedia*, pages 1493–1496, 2010. 74

[HHO07] S. Hochreiter, M. Heusel et K. Obermayer : Fast model-based protein homology detection without alignment. *Bioinformatics*, 23(14):1728–1736, 2007. 59

[Hin02] G.E. Hinton : Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002. 30, 31, 32

[Hoc91] S. Hochreiter : Studies on dynamic neural networks. Mémoire de D.E.A., Institut fur Informatik, Technische Universitat, Munchen, 1991. 57

[Hop82] J.J. Hopfield : Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554, 1982. 34

[HOT06] G.E. Hinton, S. Osindero et Y.W. Teh : A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006. 31, 146

[HP06] M. Heikkila et M. Pietikainen : A texture-based method for modeling the background and detecting moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):657–662, 2006. 20

[HPA04] A. Hadid, M. Pietikainen et T. Ahonen : A discriminative feature space for detecting and recognizing faces. *In IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–797, 2004. 20

[HS88] C. Harris et M. Stephens : A combined corner and edge detector. *In Alvey vision conference*, volume 15, page 50, 1988. 13

[HS97] S. Hochreiter et J. Schmidhuber : Long Short-Term Memory. *Neural computation*, 9(8):1735–1780, 1997. 44, 57, 58

[HS03] D.P. Huijsmans et N. Sebe : Content-based indexing performance : size normalized precision, recall, generality evaluation. *In International Conference on Image Processing*, volume 3, pages III–733, 2003. 20

[HS06] G.E. Hinton et R.R. Salakhutdinov : Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. 31

[HW62] D.H. Hubel et T.N. Wiesel : Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106, 1962. 22, 36, 37

[HZ06] W.W. Hager et H. Zhang : Algorithm 851 : CG_DESCENT, a conjugate gradient method with guaranteed descent. *ACM Transactions on Mathematical Software*, 32(1):113–137, 2006. 73

[HZZH06] L. He, C. Zou, L. Zhao et D. Hu : An enhanced LBP feature based on facial expression recognition. *In IEEE International Conference of the Engineering in Medicine and Biology Society*, pages 3300–3303, 2006. 21

[ICD08] N. Ikizler, R.G. Cinbis et P. Duygulu : Human action recognition with line and flow histograms. *In International Conference on Pattern Recognition*, pages 1 –4, 2008. 131

[Jae01] H. Jaeger : The echo state approach to analysing and training recurrent neural networks-with an erratum note. Rapport technique, GMD Forschungszentrum Informationstechnik, Sankt Augustin, 2001. 55

[JKRL09] K. Jarrett, K. Kavukcuoglu, M.A. Ranzato et Y. LeCun : What is the best multi-stage architecture for object recognition ? *In IEEE International Conference on Computer Vision*, pages 2146 –2153, 2009. 88, 120

[Jol86] I. Jolliffe : *Principal component analysis*, volume 487. Springer-Verlag, 1986. 30

[Jor86] M. I. Jordan : Attractor dynamics and parallelism in a connectionist sequential machine. *International Conference on Cognitive Science*, pages 531–546, 1986. 55

[Jor02] A. Jordan : On discriminative vs. generative classifiers : A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, 14:841, 2002. 46

[JSWP07] H. Jhuang, T. Serre, L. Wolf et T. Poggio : A biologically inspired system for action recognition. *In IEEE International Conference on Computer Vision*, pages 1–8, 2007. 40, 120, 131

[JXYY10] S. Ji, W. Xu, M. Yang et K. Yu : 3D Convolutional Neural Networks for Human Action Recognition. *In International Conference on Machine Learning*, pages 495–502, 2010. 40, 42, 82, 94, 120, 123, 124, 131

[KB03] T. Kadir et M. Brady : Scale saliency : A novel approach to salient feature and scale selection. *In International Conference on Visual Information Engineering.*, pages 25–28, 2003. 16

[KLY07] H.J. Kim, J. Lee et H.S. Yang : Human action recognition using a modified convolutional neural network. *In Advances in Neural Networks*, volume 4492 de *Lecture Notes in Computer Science*, pages 715–723. Springer Berlin / Heidelberg, 2007. xvi, 39, 40, 42, 82, 94, 120, 131

[KMS+08] A. Klaser, M. Marszałek, C. Schmid *et al.* : A spatio-temporal descriptor based on 3d-gradients. *In British Machine Vision Conference*, 2008. 17

[Koh88] T. Kohonen : Self-organization and associative memory. *Springer Series in Information Science*, 8, 1988. 34

[KP07] I. Kotsia et I. Pitas : Facial expression recognition in image sequences using geometric deformation features and support vector machines. *IEEE Transactions on Image Processing*, 16(1):172–187, 2007. 24

[KS12] Z.A. Khan et W. Sohn : A model for abnormal activity recognition and alert generation system for elderly care by Hidden Conditional Random Fields using R-Transform and generalized discriminant analysis features. *Telemedicine and e-Health*, 2012. 48

[LAC07] S. Lucey, A.B. Ashraf et J. Cohn : Investigating spontaneous facial action recognition through aam representations of the face. *Face recognition*, pages 275–286, 2007. 25

[LAKG98] M. Lyons, S. Akamatsu, M. Kamachi et J. Gyoba : Coding facial expressions with gabor wavelets. *In IEEE International Conference on Automatic Face and Gesture Recognition*, pages 200–205, 1998. xv, 22, 23

[LBBH98] Y. LeCun, L. Bottou, Y. Bengio et P. Haffner : Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. xvi, 37, 38, 39, 82, 91, 140

[LBD+90] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard et L.D. Jackel : Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems*, pages 396–404, 1990. 37, 39

[LBL07] G.C. Littlewort, M.S. Bartlett et K. Lee : Faces of pain : automated measurement of spontaneousallfacial expressions of genuine and posed pain. *In International conference on Multimodal interfaces*, pages 15–21. ACM, 2007. 23

[LBRN07] H. Lee, A. Battle, R. Raina et A.Y. Ng : Efficient sparse coding algorithms. *Advances in neural information processing systems*, 19:801, 2007. 41

[LCS+08] A.L.M. Levada, D.C. Correa, D. Salvadeo, J.H. Saito et N. Mascarenhas : Novel approaches for face recognition : template-matching using dynamic time warping and LSTM Neural Network Supervised Classification. *In IEEE International Conference on Systems, Signals and Image Processing*, pages 241–244, 2008. 59

[LEC+04] C.S. Leslie, E. Eskin, A. Cohen, J. Weston et W.S. Noble : Mismatch string kernels for discriminative protein classification. *Bioinformatics*, 20(4):467–476, 2004. 53

[LFCY06] S. Liao, W. Fan, A.C.S. Chung et D.Y. Yeung : Facial expression recognition using advanced local binary patterns, tsallis entropies and global appearance features. *In IEEE International Conference on Image Processing*, pages 665–668, 2006. 21

[LGRN09] H. Lee, R. Grosse, R. Ranganath et A.Y. Ng : Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. *In International Conference on Machine Learning*, pages 609–616. ACM, 2009. 34

[LHTG96] T. Lin, B.G. Horne, P. Tino et C.L. Giles : Learning long-term dependencies in NARX recurrent neural networks. *IEEE Transactions on Neural Networks*, 7(6):1329–1338, 1996. 57

[Lin98] T. Lindeberg : Feature detection with automatic scale selection. *International journal of computer vision*, 30(2):79–116, 1998. 15

[LJ08] F. Liu et Y. Jia : Human action recognition using manifold learning and Hidden Conditional Random Fields. *In International Conference for Young Computer Scientists*, pages 693–698, 2008. 48, 131

[LK81] B.D. Lucas et T. Kanade : An iterative image registration technique with an application to stereo vision. *In International joint conference on Artificial intelligence*, 1981. 24

[LKF10] Y. LeCun, K. Kavukcuoglu et C. Farabet : Convolutional networks and applications in vision. *In IEEE International Symposium on Circuits and Systems*, pages 253–256, 2010. 39, 82, 140

[LL03] I. Laptev et T. Lindeberg : Space-time interest points. *In International Conference on Computer Vision*, volume 16, pages 432–439, 2003. xv, xx, 13, 14, 133

[LL08] S.M. Lajevardi et M. Lech : Averaged gabor filter features for facial expression recognition. *In Computing : Techniques and Applications*, pages 71–76. IEEE, 2008. 23

[LLK10] A.A. Liu, K. Li et T. Kanade : Mitosis sequence detection using Hidden Conditional Random Fields. *In IEEE International Symposium on Biomedical Imaging : From Nano to Macro*, pages 580–583, 2010. 52

[LLS09] J. Liu, J. Luo et M. Shah : Recognizing realistic actions from videos. *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 1996 –2003, 2009. 143

[LMB+06] Y. LeCun, U. Muller, J. Ben, E. Cosatto et B. Flepp : Off-road obstacle avoidance through end-to-end learning. *Advances in neural information processing systems*, 18:739, 2006. 39

[LMP01] J. Lafferty, A. McCallum et F.C.N. Pereira : Conditional Random Fields : Probabilistic models for segmenting and labeling sequence data. *International Conference on Machine Learning*, pages 282–289, 2001. xvi, 44, 46, 47

[LMP04] R. Leonardi, P. Migliorati et M. Prandini : Semantic indexing of soccer audio-visual sequences : a multimodal approach based on controlled markov chains. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(5):634–643, 2004. 25

[LMSR08] I. Laptev, M. Marszalek, C. Schmid et B. Rozenfeld : Learning realistic human actions from movies. *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 17, 131

[LN07] F. Lv et R. Nevatia : Single view human action recognition using key pose matching and viterbi path searching. *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 19

[Low04] D.G. Lowe : Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 13, 16, 68

[LS99] D. Lee et H. Seung : Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999. 30, 98

[LSF05] A. Lucieer, A. Stein et P. Fisher : Multivariate texture-based segmentation of remotely sensed imagery for extraction of objects and their uncertainty. *International Journal of Remote Sensing*, 26(14):2917–2936, 2005. 20

[LSST+02] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini et C. Watkins : Text classification using string kernels. *The Journal of Machine Learning Research*, 2:419–444, 2002. 53

[LSTI02] W. Lee, C.C. Sekhar, K. Takeda et F. Itakura : Recognition of continuous speech segments of monophone units using Support Vector Machines. *In International Conference on Spoken Language Processing*, 2002. 52

[Lu12] Q. Lu : Temporal action localization in videos with recurrent neural networks. Mémoire de D.E.A., Université René Descartes (Paris V), 2012. xviii, 145, 146

[LW99] C.J. Lee et S.D. Wang : Fingerprint feature extraction using gabor filters. *Electronics Letters*, 35(4):288–290, 1999. 21

[LWH90] K.J. Lang, A.H. Waibel et G.E. Hinton : A time-delay neural network architecture for isolated word recognition. *Neural networks*, 3(1):23–43, 1990. 55

[LWW+11] G. Littlewort, J. Whitehill, T.F. Wu, N. Butko, P. Ruvolo, J. Movellan et M. Bartlett : The motion in emotion - A CERT based approach to the FERA emotion challenge. *In IEEE International Conference on Automatic Face & Gesture Recognition*, pages 897–902, 2011. 132

[LYYL12] X. Lu, Y. Yuan, P. Yan et X. Li : Robust visual tracking with discriminative sparse learning. *Pattern Recognition*, 2012. 99

[LZYN11] Q.V. Le, W.Y. Zou, S.Y. Yeung et A.Y. Ng : Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 3361–3368, 2011. 41, 131

[MBPS09] J. Mairal, F. Bach, J. Ponce et G. Sapiro : Online dictionary learning for sparse coding. *In International Conference on Machine Learning*, pages 689–696, 2009. 99, 100

[MC01] D.P. Mandic et J. Chambers : *Recurrent Neural Networks for Prediction : Learning Algorithms, Architectures and Stability*. John Wiley & Sons, Inc. New York, USA, 2001. 55

[Mer09] J. Mercer : Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London*, pages 415–446, 1909. 53

[MGW+08] H. Mayer, F. Gomez, D. Wierstra, I. Nagy, A. Knoll et J. Schmidhuber : A system for robotic heart surgery that learns to tie knots using recurrent neural networks. *Advanced Robotics*, 22(13-14):1521–1537, 2008. 59

[MHV03] P.J. Moreno, P. Ho et N. Vasconcelos : A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications. *Advances in Neural Information Processing Systems*, 16:1385–1393, 2003. 53

[MI00] H. Miyamori et S. Iisaku : Video annotation for content-based retrieval using human behavior analysis and domain knowledge. *In IEEE International Conference on Automatic Face and Gesture Recognition*, pages 320–325, 2000. 27

[MJ10] L. Medsker et L. Jain : *Recurrent neural networks : design and applications*. CRC press, 2010. 54

[ML11] X. Mei et H. Ling : Robust visual tracking and vehicle classification via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2259–2272, 2011. 99

[MLS09] M. Marszalek, I. Laptev et C. Schmid : Actions in context. *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 2929–2936, 2009. 143

[Moz93] M.C. Mozer : Induction of multiscale temporal structure. *Advances in neural information processing systems*, pages 275–275, 1993. 57

[MRG07] F. Mamalet, S. Roux et C. Garcia : Real-time video convolutional face finder on embedded platforms. *Journal on Embedded Systems*, 1:22, 2007. 122

[MRPBB11] H. Meng, B. Romera-Paredes et N. Bianchi-Berthouze : Emotion recognition by two view svm_2k classifier on dynamic facial expression features. *In IEEE International Conference on Automatic Face & Gesture Recognition*, pages 854–859, 2011. 132

[NB06] M. Neuhaus et H. Bunke : Edit distance-based kernel functions for structural pattern classification. *Pattern Recognition*, 39(10):1852–1863, 2006. 53

[NDA09] H. Naomi, L. Daire et K. Anil : On Parsing Visual Sequences with the Hidden Markov Model. *Journal on Image and Video Processing*, 2009. 46

[NDL+05] F. Ning, D. Delhomme, Y. LeCun, F. Piano, L. Bottou et P.E. Barbano : Toward automatic phenotyping of developing embryos from videos. *IEEE Transactions on Image Processing*, 14(9):1360–1371, 2005. 39, 42, 82

[NP95] S.J. Nowlan et J.C. Platt : A convolutional neural network hand tracker. *Advances in Neural Information Processing Systems*, pages 901–908, 1995. 39

[NPZ02] C.W. Ngo, T.C. Pong et H.J. Zhang : On Clustering and Retrieval of Video Shots Through Temporal Slices Analysis. *IEEE Transactions on Multimedia*, 4(4), 2002. 28

[NTF09] F. Nasse, C. Thurau et G. Fink : Face detection using gpu-based convolutional neural networks. *In Computer Analysis of Images and Patterns*, pages 83–90, 2009. 39

[NW70] S.B. Needleman et C.D. Wunsch : A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970. 53, 72

[NWFF08] J.C. Niebles, H. Wang et L. Fei-Fei : Unsupervised learning of human action categories using spatial-temporal words. *International journal of computer vision*, 79:299–318, 2008. xv, 15, 18, 131

[OCM07] M. Osadchy, Y.L. Cun et M.L. Miller : Synergistic face detection and pose estimation with energy-based models. *The Journal of Machine Learning Research*, 8:1197–1215, 2007. 39

[OF97] B.A. Olshausen et D.J. Field : Sparse coding with an overcomplete basis set : A strategy employed by VI ? *Vision research*, 37(23):3311–3326, 1997. 30, 98, 99, 102

[OLFM07] A. Oliver, X. Lladó, J. Freixenet et J. Martí : False positive reduction in mammographic mass detection using local binary patterns. *Medical Image Computing and Computer-Assisted Intervention*, pages 286–293, 2007. 20

[OPH96] T. Ojala, M. Pietikäinen et D. Harwood : A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996. 20

[OPP05] A. Oikonomopoulos, I. Patras et M. Pantic : Spatiotemporal saliency for human action recognition. *In IEEE International Conference on Multimedia and Expo*, page 4, 2005. 16

[PB07] M. Pantic et M.S. Bartlett : Machine analysis of facial expressions. *Face Recognition*, pages 377–416, 2007. 23

[PJC98] G.S. Pingali, Y. Jean et I. Carlbom : Real time tracking for enhanced tennis broadcasts. *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 260–265, 1998. 27

[PJZ01] M. Petkovic, W. Jonker et Z. Zivkovic : Recognizing strokes in tennis videos using Hidden Markov Models. *In International Conference on Visualization, Imaging and Image Processing*, 2001. 46

[PNH86] D. Plaut, S. Nowlan et G. E. Hinton : Experiments on learning by back propagation. Rapport technique TR CMU-CS-86-126, Carnegie Mellon University, Department of Computer Science, 1986. 36

[POGES03] J.A. Pérez-Ortiz, F.A. Gers, D. Eck et J. Schmidhuber : Kalman filters improve LSTM network performance in problems unsolvable by traditional recurrent nets. *Neural Networks*, 16(2):241–250, 2003. 59

[Por88] A.B. Poritz : Hidden Markov Models : A guided tour. *In IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 7–13, 1988. 45

[PP05] M. Pantic et I. Patras : Detecting facial actions and their temporal segments in nearly frontal-view face image sequences. *In IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 3358–3363, 2005. 25

[QWM+07] A. Quattoni, S. Wang, L.-P. Morency, M. Collins et T. Darrell : Hidden Conditional Random Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1848–1852, 2007. xvi, 44, 48, 49

[RA09] M.S. Ryoo et J.K. Aggarwal : Spatio-temporal relationship match : Video structure comparison for recognition of complex human activities. *In IEEE International Conference on Computer Vision*, pages 1593 –1600, 2009. 143

[Rab89] L.R. Rabiner : A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. xvi, 45, 47

[RAS08] M.D. Rodriguez, J. Ahmed et M. Shah : Action mach a spatio-temporal maximum average correlation height filter for action recognition. *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 131, 143

[RF87] A.J. Robinson et F. Fallside : The utility driven dynamic error propagation network. Rapport technique CUED/F-INFENG/TR. 1, Cambridge University, Engineering Department, 1987. 55

[RGA00] Y. Rui, A. Gupta et A. Acero : Automatically extracting highlights for tv baseball programs. *In ACM international conference on Multimedia*, pages 105–115, 2000. 25

[RHBL07] M.A. Ranzato, F.J. Huang, Y.L. Boureau et Y. Lecun : Unsupervised learning of invariant feature hierarchies with applications to object recognition. *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. xvii, xx, 30, 98, 99, 100, 102, 103, 104, 106, 107, 108, 113, 114, 115, 135, 136, 141, 146

[RHW86] D.E. Rumelhart, G.E. Hinton et R.J. Williams : Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986. 30, 34, 36, 98

[Ros57] F. Rosemblat : The perceptron : A perceiving and recognizing automation. Rapport technique 47, Report 85-460-1, Cornell Aeronautical Laboratory, Ithaca, New York., 1957. 34, 35

[RPCL06] M.A. Ranzato, C. Poultney, S. Chopra et Y. LeCun : Efficient learning of sparse representations with an energy-based model. *Advances in neural information processing systems*, 19:1137–1144, 2006. 30, 98, 99, 100, 102, 106, 107, 108, 113, 114, 141, 146

[SAS07] P. Scovanner, S. Ali et M. Shah : A 3-dimensional sift descriptor and its application to action recognition. *In International conference on Multimedia*, pages 357–360, 2007. 17

[SCH09] X. Sun, M. Chen et A. Hauptmann : Action recognition via local descriptors and holistic features. *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 58 –65, 2009. 131

[SDNFF08] S. Savarese, A. DelPozo, J.C. Niebles et L. Fei-Fei : Spatial-temporal correlatons for unsupervised action classification. *In IEEE Workshop on Motion and video Computing*, pages 1–8, 2008. 18

[SG07] Z. Saidane et C. Garcia : Automatic scene text recognition using a convolutional neural network. *In International Workshop on Camera-Based Document Analysis and Recognition*, pages 100–106, 2007. 39

[SGE02] J. Schmidhuber, F. Gers et D. Eck : Learning nonregular languages : A comparison of simple recurrent networks and LSTM. *Neural Computation*, 14(9):2039–2041, 2002. 59

[SGM05] C. Shan, S. Gong et P.W. McOwan : Robust facial expression recognition using local binary patterns. *In IEEE International Conference on Image Processing*, volume 2, pages II–370, 2005. 20

[SGM09] C. Shan, S. Gong et P.W. McOwan : Facial expression recognition based on local binary patterns : A comprehensive study. *Image and Vision Computing*, 27(6):803–816, 2009. 21

[SH07a] R. Salakhutdinov et G. Hinton : Learning a nonlinear embedding by preserving class neighbourhood structure. *In Artificial Intelligence and Statistics*, volume 11, 2007. 29

[SH07b] I. Sutskever et G.E. Hinton : Learning multilevel distributed representations for high-dimensional sequences. *In International Conference on Artificial Intelligence and Statistics*, pages 544–551, 2007. xv, 31, 33

[Sim91] P.K. Simpson : Fuzzy min-max neural networks. *In IEEE International Joint Conference on Neural Networks*, pages 1658–1669, 1991. 39

[SJ09] Y.H. Sung et D. Jurafsky : Hidden conditional random fields for phone recognition. *In IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 107–112, 2009. 48, 49

[SLC04] C. Schuldt, I. Laptev et B. Caputo : Recognizing human actions : A local SVM approach. *In IEEE International Conference on Pattern Recognition*, volume 3, pages 32–36, 2004. xvii, xviii, 18, 52, 86, 88, 92, 98, 111, 113, 118, 119, 141

[SM86] G. Salton et M.J. McGill : *Introduction to modern information retrieval*. McGraw-Hill, Inc., 1986. 17, 93

[Smo86] P. Smolensky : Information processing in dynamical systems : foundations of harmony theory. *In Parallel distributed processing : explorations in the microstructure of cognition*, volume 1, pages 194–281. MIT Press, 1986. xv, 30, 31

[SO05] D.A. Sadlier et N.E. O'Connor : Event detection in field sports video using audio-visual features and a support vector machine. *Transactions on Circuits and Systems for Video Technology*, 15(10):1225–1233, 2005. 52

[SP97] M. Schuster et K.K. Paliwal : Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997. 56

[Sug07] M. Sugiyama : Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. *The Journal of Machine Learning Research*, 8:1027–1061, 2007. 29

[SVG08] K. Schindler et L. Van Gool : Action snippets : How many frames does human action recognition require ? *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 18, 131

[SWG05] J. Schmidhuber, D. Wierstra et F. Gomez : Evolino : Hybrid neuroevolution/optimal linear search for sequence prediction. *In International Joint Conference on Artificial Intelligence*, 2005. 59, 80

[TCP04] D. Tjondronegoro, Y.P.P. Chen et B. Pham : Highlights for more complete sports video summarization. *MultiMedia*, 11(4):22–37, 2004. 25

[Tea80] M. R. Teague : Image analysis via the general theory of moments. *Journal of the Optical Society of America*, 70(8):920–930, 1980. 90

[TFLB10] G. Taylor, R. Fergus, Y. LeCun et C. Bregler : Convolutional learning of spatio-temporal features. *European Conference on Computer Vision*, pages 140–153, 2010. xvi, 33, 34, 131

[THR07] G.W. Taylor, G.E. Hinton et S.T. Roweis : Modeling human motion using binary latent variables. *Advances in neural information processing systems*, 19:1345, 2007. 33

[TKC01] Y.I. Tian, T. Kanade et J.F. Cohn : Recognizing action units for facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):97–115, 2001. 24

[TLD+05] X. Tong, Q. Liu, L. Duan, H. Lu, C. Xu et Q. Tian : A unified framework for semantic shot representation of sports video. *In ACM International Workshop on Multimedia Information Retrieval*, page 134, 2005. 27

[TLL+11] U. Tariq, K.H. Lin, Z. Li, X. Zhou, Z. Wang, V. Le, T.S. Huang, X. Lv et T.X. Han : Emotion recognition from an ensemble of features. *In IEEE International Conference on Automatic Face & Gesture Recognition*, pages 872–877, 2011. 132

[TSKR00] Y.P. Tan, D.D. Saur, S.R. Kulkami et P.J. Ramadge : Rapid estimation of camera motion from compressed video with application to video annotation. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(1):133–146, 2000. 28

[TWL+10] A.P. Ta, C. Wolf, G. Lavoue, A. Baskurt et J. Jolion : Pairwise features for human action recognition. *In International Conference on Pattern Recognition*, pages 3224–3227, 2010. 131

[TWOH03] Y.W. Teh, M. Welling, S. Osindero et G.E. Hinton : Energy-based models for sparse overcomplete representations. *The Journal of Machine Learning Research*, 4:1235–1260, 2003. 99, 102

[Vap98] V. Vapnik : *Statistical Learning Theory*. John Wiley & Sons, 1998. 44, 50, 51

[VJ01] P. Viola et M. Jones : Robust real-time face detection. *In International Journal of Computer Vision*, 2001. 24

[VJM+11] M.F. Valstar, B. Jiang, M. Mehu, M. Pantic et K. Scherer : The first facial expression recognition and analysis challenge. *In IEEE International Conference on Automatic Face & Gesture Recognition*, pages 921–926, 2011. xviii, 21, 111, 121, 122, 132, 141

[VP05] D. Vukadinovic et M. Pantic : Fully automatic facial feature point detection using gabor feature based boosted classifiers. *In IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 1692–1698, 2005. xv, 24

[VP06] M. Valstar et M. Pantic : Fully automatic facial action unit detection and temporal analysis. *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 149–149, 2006. 25

[VPP04] M. Valstar, M. Pantic et I. Patras : Motion history for facial action detection in video. *In IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 635–640, 2004. 23

[WBR+11] T. Wu, N.J. Butko, P. Ruvolo, J. Whitehill, M.S. Bartlett et J.R. Movellan : Action unit recognition transfer across datasets. *In IEEE International Conference on Automatic Face & Gesture Recognition*, pages 889–896, 2011. 23

[WC07] S.F. Wong et R. Cipolla : Extracting spatiotemporal interest points using global information. *In IEEE International Conference on Computer Vision*, pages 1–8, 2007. 17, 131

[WEK+09] M. Wöllmer, F. Eyben, J. Keshet, A. Graves, B. Schuller et G. Rigoll : Robust discriminative keyword spotting for emotionally colored spontaneous speech using bidirectional LSTM networks. *In IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3949–3952. IEEE, 2009. 59

[WJC02] C. Wolf, J. Jolion et F. Chassaing : Text localization, enhancement and binarization in multimedia documents. *In International Conference on Pattern Recognition*, pages 1037–1040, 2002. 71

[WJKB00] C. Wolf, J.-M. Jolion, W. Kropatsch et H. Bischof : Content based image retrieval using interest points and texture features. *In IEEE International Conference on Pattern Recognition*, volume 4, pages 234–237, 2000. 21

[WKES12] M. Wöllmer, M. Kaiser, F. Eyben et B. Schuller : LSTM-Modeling of continuous emotions in an audiovisual affect recognition framework. *Image and Vision Computing*, 2012. 59

[WM08] Y. Wang et G. Mori : Learning a discriminative hidden part model for human action recognition. *In Advances in Neural Information Processing Systems*, 2008. 48

[WML+12] C. Wolf, J. Mille, E. Lombardi, O. Celiktutan, M. Jiu, M. Baccouche, E. Dellandrea, C.-E. Bichot, C. Garcia et Sankur B. : The LIRIS Human activities dataset and the ICPR 2012 human activities recognition and localization competition. Rapport technique LIRIS-2012-004, LIRIS Laboratory, 2012. 143

[WO06] J. Whitehill et C.W. Omlin : Haar features for facs au recognition. *In IEEE International Conference on Automatic Face and Gesture Recognition*, page 5, 2006. 23

[WQM+06] S.B. Wang, A. Quattoni, L.P. Morency, D. Demirdjian et T. Darrell : Hidden conditional random fields for gesture recognition. *In Computer Vision and Pattern Recognition*, volume 2, pages 1521–1527. IEEE, 2006. 48, 49

[WRB06] D. Weinland, R. Ronfard et E. Boyer : Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2):249–257, 2006. 19

[WRB10] D. Weinland, R. Ronfard et E. Boyer : A survey of vision-based methods for action representation, segmentation and recognition. Rapport technique INRIA-00459653, Institut National de Recherche en Informatique et en Automatique, 2010. xv, 18

[WTVG08] G. Willems, T. Tuytelaars et L. Van Gool : An efficient dense and scale-invariant spatio-temporal interest point detector. *European Conference on Computer Vision*, pages 650–663, 2008. xv, 15, 16, 18, 131

[WUK+09] H. Wang, M.M. Ullah, A. Klaser, I. Laptev et C. Schmid : Evaluation of local spatio-temporal features for action recognition. *In British Machine Vision Conference*, 2009. 133

[WZ95] R.J. Williams et D. Zipser : Gradient-based learning algorithms for recurrent networks and their computational complexity. *Back-propagation : Theory, architectures and applications*, pages 433–486, 1995. 55

[XG08] T. Xiang et S. Gong : Activity based surveillance video content modelling. *Pattern Recognition*, 41(7):2309–2326, 2008. 46

[XXC+01] P. Xu, L. Xie, S.F. Chang, A. Divakaran, A. Vetro et H. Sun : Algorithms and system for segmentation and structure analysis in soccer video. *In IEEE International Conference on Multimedia and Expo*, pages 721–724, 2001. 26

[XXC+04] L. Xie, P. Xu, S.F. Chang, A. Divakaran et H. Sun : Structure analysis of soccer video with domain knowledge and hidden Markov models. *Pattern Recognition Letters*, 25(7):767–775, 2004. 26

[YB11] S. Yang et B. Bhanu : Facial expression recognition using emotion avatar image. *In IEEE International Conference on Automatic Face Gesture Recognition*, pages 866 –871, 2011. 132

[YDA09] D. Yu, L. Deng et A. Acero : Hidden conditional random field with distribution constraints for phone classification. *In Interspeech*, pages 676–679, 2009. 48, 49

[YS05] A. Yilmaz et M. Shah : Actions sketch : A novel action representation. *In IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 984–989, 2005. 40

[ZC01] D. Zhong et S.F. Chang : Structure analysis of sports video using domain models. *In IEEE International Conference on Multimedia and Expo*, pages 713–716, 2001. 26

[ZG10] J. Zhang et S. Gong : Action categorization with modified hidden conditional random field. *Pattern Recognition*, 43(1):197–203, 2010. 48, 49

[ZGPBM05] D. Zhang, D. Gatica-Perez, S. Bengio et I. McCowan : Semi-supervised adapted hmms for unusual event detection. *In IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 611–618, 2005. 46

[ZJ05] Y. Zhang et Q. Ji : Active and dynamic information fusion for facial expression understanding from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):699–714, 2005. 25

[ZMLS07] J. Zhang, M. Marszałek, S. Lazebnik et C. Schmid : Local features and kernels for classification of texture and object categories : A comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, 2007. 51

[ZP07] G. Zhao et M. Pietikainen : Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):915–928, 2007. 21

[ZVK00] W. Zhou, A. Vellaikal et C.C.J. Kuo : Rule-based video classification system for basketball video indexing. *In ACM workshops on Multimedia*, pages 213–216, 2000. 28

[ZWIL00] D. Zhang, A. Wong, M. Indrawan et G. Lu : Content-based image retrieval using gabor texture features. *In IEEE Pacific-Rim Conference on Multimedia*. University of Sydney, Australia, 2000. 21