Description de contenu vidéo : mouvements et élasticité temporelle

Katy Blanc

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Description de Contenu Vidéo

Mouvements et Élasticité Temporelle

Katy BLANC

Laboratoire I3S

Présentée en vue de l'obtention du grade de docteur en Informatique **d'**Université Côte d'Azur
Dirigée par : Diane LINGRAND
Co-encadrée par : Frédéric PRECIOSO
Soutenue le : 17 décembre 2018

Devant le jury, composé de :
François BREMOND, Professeur, INRIA, *Examinateur*
Matthieu CORD, Professeur, LIP6, *Rapporteur*
Thomas MENGUY, Directeur général, WILDMOKA, *Examinateur*
Bernard MERIALDO, Directeur de recherches, EURECOM, *Examinateur*
Nicolas THOME, Professeur, CNAM, *Rapporteur*

i3S sophia antipolis

Description of Video Content : Motion and Temporal Elasticity

Abstract :

Video recognition gain in performance during the last years, especially due to the improvement in the deep learning performances on images. However the jump in recognition rate on images does not directly impact the recognition rate on videos. This limitation is certainly due to this added dimension, the time, on which a robust description is still hard to extract. The recurrent neural networks introduce temporality but they have a limited memory. State of the art methods for video description usually handle time as a spatial dimension and the combination of video description methods reach the current best accuracies. However the temporal dimension has its own elasticity, different from the spatial dimensions. Indeed, the temporal dimension of a video can be locally deformed : a partial dilatation produces a visual slow down during the video, without changing the understanding, in contrast with a spatial dilatation on an image which will modify the proportions of the shown objects. We can thus expect to improve the video content classification by creating an invariant description to these speed changes.

This thesis focus on the question of a robust video description considering the elasticity of the temporal dimension under three different angles.

First, we have locally and explicitly described the motion content. Singularities are detected in the optical flow, then tracked along the time axis and organized in chain to describe video part. We have used this description on sport content.

Then we have extracted global and implicit description thanks to tensor decompositions. Tensor enables to consider a video as a multi-dimensional data table. The extracted description are evaluated in a classification task.

Finally, we have studied speed normalization method thanks to Dynamical Time Warping methods on series. We have showed that this normalization improve the classification rates.

Keywords : Video, Recognition, Temporal Dimension, Tensor Decomposition, Dynamical Time Warping

Description de Contenu Vidéo : Mouvements et Élasticité Temporelle

Résumé : La reconnaissance en vidéo atteint de meilleures performances ces dernières années, notamment grâce à l'amélioration des réseaux de neurones profonds sur les images. Pourtant l'explosion des taux de reconnaissance en images ne s'est pas directement répercuté sur les taux en reconnaissance vidéo. Cela est dû à cette dimension supplémentaire qu'est le temps et dont il est encore difficile d'extraire une description robuste. Les réseaux de neurones récurrents introduisent une temporalité mais ils ont une mémoire limitée dans le temps. Les méthodes de description vidéo de l'état de l'art gèrent généralement le temps comme une dimension spatiale supplémentaire et la combinaison de plusieurs méthodes de description vidéo apportent les meilleures performances actuelles. Or la dimension temporelle possède une élasticité propre, différente des dimensions spatiales. En effet, la dimension temporelle peut être déformée localement : une dilatation partielle provoquera un ralentissement visuel de la vidéo sans en changer la compréhension, à l'inverse d'une dilatation spatiale sur une image qui modifierait les proportions des objets. On peut donc espérer améliorer encore la classification de contenu vidéo par la conception d'une description invariante aux changements de vitesse.

Cette thèse porte sur la problématique d'une description robuste de vidéo en considérant l'élasticité de la dimension temporelle sous trois angles différents.

Dans un premier temps, nous avons décrit localement et explicitement les informations de mouvements. Des singularités sont détectées sur le flot optique, puis traquées et agrégées dans une chaîne pour décrire des portions de vidéos. Nous avons utilisé cette description sur du contenu sportif.

Puis nous avons extrait des descriptions globales implicites grâce aux décompositions tensorielles. Les tenseurs permettent de considérer une vidéo comme un tableau de données multi-dimensionnelles. Les descriptions extraites sont évaluées dans une tache de classification.

Pour finir, nous avons étudié les méthodes de normalisation de la dimension temporelle. Nous avons utilisé les méthodes de déformations temporelles dynamiques des séquences. Nous avons montré que cette normalisation aide à une meilleure classification.

Mots-clés : Vidéo, Temporalité, Élasticité, Tenseur, Déformations Temporelles Dynamiques

Remerciements

Comme le disait Hans Christian Andersen "La reconnaissance est la mémoire du coeur".

J'aimerais alors fixer dans ma mémoire et dans mon manuscrit, à travers mes remerciements, les merveilleuses personnalités qui ont embelli ces années de thèse.

J'aimerais remercier en premier lieu mes encadrants, Diane Lingrand et Frédéric Precioso, pour m'avoir fait confiance sur ce sujet de thèse et m'avoir intégrée à leur équipe. Tout au long de la thèse, ils m'ont initiée au monde de la recherche scientifique en partageant la richesse de leurs connaissances du domaine avec enthousiasme, lucidité et sincérité. Ces moments furent particulièrement enrichissants autant sur le plan professionnel que sur le plan humain.

Je tiens à remercier mes deux rapporteurs, Matthieu Cord, Professeur au laboratoire Lip6, et le Professeur Nicolas Thome du CNAM, d'avoir donné de leur précieux temps pour évaluer ma thèse. Merci au directeur de recherches d'Eurecom, Bernard Merialdo, au Professeur François Bremond de l'Inria et à Thomas Menguy, directeur général de Wildmoka, qui ont accepté de faire partie de mon jury de soutenance.

Mes remerciements s'adressent de même à Cristian Livadiotti et à nouveau à Thomas Menguy, les co-fondateurs de Wildmoka, qui nous ont exposé les problématiques actuelles et leurs expériences sur les applications vidéos dans le contexte industriel. Ces nouveaux enjeux concrets m'ont permis de discerner le potentiel de la vidéo dans le monde numérique moderne.

Je remercie Arnaud Revel, Geoffrey Portelli et, de nouveau, Matthieu Cord pour m'avoir apporté leur point de vue et leur pertinence scientifique sur des facettes particulières de ma thèse.

J'aimerais aussi remercier mes collègues enseignants qui ont su communiquer leur plaisir de l'enseignement, leur organisation et leurs conseils pédagogiques, rendant l'atmosphère propice à l'intégration d'intervenant et au partage de leur méthodologie, en particulier merci à Stéphane Lavirotte, Vincent Granet, Dino Lopez, Hélène Renard et Diane Lingrand.

Je voudrais également remercier chaleureusement Igor Litovsky pour sa sympathie et sa bienveillance. Il m'a encouragé dans mon projet de thèse, m'a présenté aux personnes compétentes et m'a fournit les recommandations nécessaires pour le mener à bien.

Je remercie pareillement toute l'équipe Sparks pour leur accueil, en particulier merci à Anne-Marie, Marco et Stéphane pour avoir pris régulièrement la température. Je remercie l'équipe administrative du laboratoire I3S, en particulier Magali qui allie à la perfection réactivité, efficacité et amabilité.

Je tiens à remercier les collègues que j'ai côtoyés au cours de ces années, en particulier Améni, John, Mélanie, Benjamin, Gerald, Xhevahire et Thomas, pour les partages d'expériences nouvelles et leur bonne humeur. Un immense merci à Lucas et Stéphanie pour leur soutien constant et les duos au piano si agréables à mon oreille profane.

Je remercie vivement mes amies, Natacha, Virginie et Félicie, qui m'ont offert des moments de détentes et qui ont relu mon manuscrit.

Je remercie mon père et ma belle-mère, Pierre et Safia, qui m'ont encouragée et m'ont chaleureusement reçue à de nombreux déjeuners improvisés. Merci à mon mari, Brice, pour son amour, sa bienveillance, son écoute et son soutien inconditionnel durant ces dernières années.

Table des matières

Table des figures

Liste des tableaux

CHAPITRE 1

Introduction

Contents

1.1 La vidéo et sa dimension temporelle

Une vidéo est un ensemble d'images successives qui nous permettent de visualiser une scène dynamique. L'interprétation de cette succession d'images en mouvement par notre vision est due à la persistance rétinienne qui permet de garder en mémoire ce que l'on voit durant une courte durée. L'image suivante étant très peu différente, la persistance rétinienne permet de lier cette image à la précédente et leurs différences sont interprétées comme un mouvement entre ces deux temps. Cette faculté du cerveau permettant de relier automatiquement de façon cohérente deux images statiques consécutives est nommée l'Effet Phi. Cet effet est contraint à un minimum de 24 images par seconde, pour ne pas percevoir la fixation des images. Nous savons que la vision primaire consiste à analyser les orientations, les formes, les reliefs (perceptibles uniquement en vidéo 3D) et les mouvements. Notre vision analyse aussi bien les mouvements continus que saccadés. De plus, il est bien connu que le mouvement attire plus l'attention [44].

De nombreuses études effectuées sur la localisation de l'attention d'un utilisateur face à une vidéo, grâce aux appareils d'oculométrie, montrent que les coins et les orientations influent le regard ; mais les caractéristiques temporelles, comme une lumière vacillante ou un mouvement, offre une meilleure prédiction de l'intérêt et du regard d'un spectateur [71]. D'ailleurs, plus le mouvement est important, plus l'attention est captée.

Par conséquent, biologiquement parlant, le mouvement focalise notre attention dans une vidéo et, comme nous arrivons à comprendre une vidéo, cela nous donne un indice pour la création d'outils d'analyse vidéo. De plus, l'évolution temporelle

nous fournit des indices qu'aucune description spatiale ne fournit. En effet, sans cette analyse, on ne peut reconnaître ni un ralenti, cet effet de production vidéo qui permet de mettre en avant un événement important, ni différentier un avion qui décolle d'un avion qui atterrit [24]. De plus, la vitesse d'un coup de pied dans un ballon nous donne des informations concernant la force fournie par le joueur. Enfin, le fait qu'une action est répétée plusieurs fois permet également d'analyser des situations. On ne peut parler de mouvements sans parler de temps, d'évolution temporelle et donc de dimension temporelle de la vidéo. L'analyse temporelle est donc utile et nécessaire.

Ainsi, dans cette thèse, nous voulons nous concentrer sur cette dimension temporelle de la vidéo.

1.2 État de l'art

L'analyse vidéo est faite dans de nombreuses et diverses applications puisque les vidéos disponibles, notamment en ligne, sont de plus en plus nombreuses et diversifiées. Nous pouvons classer ces analyses vidéos par leur objectif, en deux catégories : l'extraction d'information de la vidéo et la génération de vidéo.

Par exemple, nous avons des applications vidéo utilisant l'extraction de contenu comme la reconnaissance d'actions humaines, la surveillance qu'elle soit publique (détection d'agressions, analyse de mouvements de foule,...) ou privée (protection des personnes âgées, surveillance d'animaux, connaître la consommation des clients dans un magasin, ...). Les applications qui peuvent utiliser la génération de vidéo sont la création de résumé, de ralentis, la production vidéo comme l'ajout d'effets spéciaux.

1.2.1 Extraire l'information

Dans cette partie, nous nous intéresserons à la reconnaissance, qui inclura la classification et la régression puisque l'objectif est le même, à savoir extraire de l'information sémantique d'une vidéo. Nous évoquerons également la détection, car elle est toujours un moyen d'extraire une information, mais cette fois avec un détail sur sa position dans la vidéo.

Plus précisément, nous nous intéressons ici aux techniques d'analyse d'évolution temporelle qui sont généralisées à l'ensemble des vidéos : le but étant d'analyser le temps dans sa globalité. En effet, une grande partie des applications de l'analyse vidéo sont basées sur l'étude d'actions humaines, et donc les méthodes sont souvent spécialisées pour cette tache. Par exemple, certaines méthodes utilisent des détections de squelettes pour la reconnaissance d'actions humaines [19] ou des tubes d'actions qui sont formés par des fenêtres de détection autour de détections de corps humains et qui servent alors de base pour la description et la reconnaissance [33, 53]. Ces méthodes seront alors mises de côté dans notre analyse de l'état de l'art.

L'analyse vidéo a commencé par l'adaptation des descriptions en "sac de mots" d'une image à la vidéo [28, 58, 73, 74] à partir de descripteurs classiques faits main ou à partir des descriptions extraites de réseaux de neurones convolutionnels. Bien qu'efficace en tant que modélisation d'une application directe, cette technique perd toute structure temporelle. Puis le travail fondateur de Laptev *et al.* a permis de détecter et décrire des points d'intérêt d'une vidéo en étendant le détecteur de Harris à une dimension supplémentaire : les points d'intérêts spatio-temporels (STIP) [57]. En effet, la méthode de Harris consiste à repérer les changements spatiaux locaux (les coins), et les STIP ajoutent la contrainte de changements temporels locaux (les mouvements). La restriction de la grande quantité d'information à certains points d'intérêt est le grand avantage de cette méthode. Cependant, elle représente le temps comme l'espace alors que leurs caractéristiques sont différentes et l'analyse de l'évolution temporelle est très locale. Le travail de Wang *et al.* sur les descripteurs denses de trajectoires (Improved Dense Trajectories IDT) [106] contre l'aspect local en proposant d'une part une description dense spatialement et un suivi dans le temps de ces éléments. Leurs descriptions sont alors constituées d'un descripteur spatial local HOG, d'un descripteur spatial du flot optique HOF et d'un descripteur des changements spatiaux du flot optique (MBH) qui sont moyennés tout au long de ce suivi, auquel on ajoute les vecteurs de déplacement représentant ce suivi. Rappelons que le flot optique est un champ de vecteurs décrivant une carte des déplacements entre deux images. Cette méthode de description est encore beaucoup utilisée dans la description puisqu'elle ne dépend pas d'un ensemble d'apprentissage ; de plus certaines méthodes d'apprentissage profond obtiennent une amélioration de la classification quand elles sont associées à une décision prise à partir d'une description par IDT [29, 96]. Par conséquent, cette description apporte des informations non-perceptibles par des architectures de neurones, malgré que les descriptions par réseaux de neurones opèrent mieux qu'une description seule par IDT.

Dans le domaine de la vision par ordinateur, les dernières années ont été marquées par un bon dans les performances sur les images grâce à l'arrivée de l'apprentissage profond [56], elle-même dû à l'évolution du matériel informatique et à l'abondance des contenus multimédia permettant d'entraîner ces modèles, que ce soit pour la classification ou la génération d'images. Alors, dans le domaine de l'analyse vidéo, la tendance est à réutiliser ce qui fonctionne pour les images en les généralisant à une nouvelle dimension. Ainsi, les dimensions spatiales et la dimension temporelle d'une vidéo sont traitées de façon semblable. Ainsi, des ensembles d'apprentissage plus consistants sont construits et l'analyse vidéo s'ouvre de plus en plus à l'apprentissage profond. Bien que le développement d'architectures de représentation d'images ait rapidement convergé vers une tendance, il n'y a toujours pas clairement d'architecture en tête pour la reconnaissance vidéo. On peut classer les types d'architectures à partir de trois questions :

— Les noyaux de convolutions sont-ils en deux ou trois dimensions ?

— L'entrée du réseaux est-elle une vidéo RGB ou la vidéo et ses images de flots

optiques pré-calculées ?

— Les informations spatiales sont-elles agrégées temporellement par des techniques de fusion simple (moyenne, maximum, ...) ou par une analyse récurrente ?

Ainsi, on peut présenter les principaux types d'architectures vidéo comme sur l'illustration 1.1.

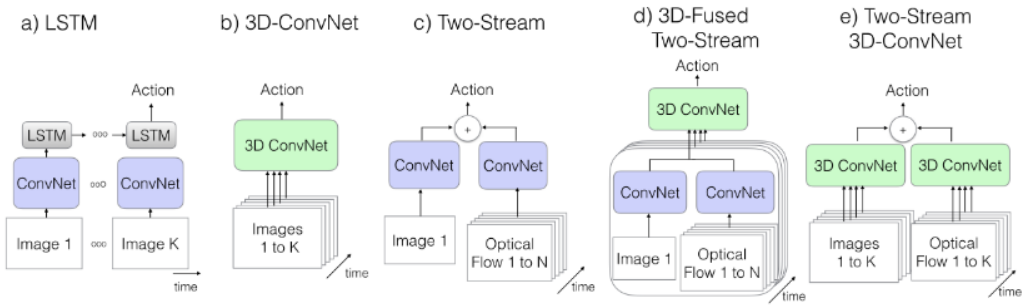


FIGURE 1.1 – Architectures vidéo les plus fréquentes. K est le nombre total d'images dans une vidéo, tandis que N est choisi pour un sous-ensemble d'images consécutives de la vidéo (illustration provenant de [13]).

Comme nous l'avons remarqué, les hautes performances en classification d'images appellent à une réutilisation de ces réseaux pour la vidéo. Chaque image peut donc être décrite indépendamment par ces réseaux et ces prédictions sont accumulées sur la vidéo entière [47], ce qui peut être sous forme de "sacs de mots" [58, 73].

Cependant, une approche plus efficace est d'ajouter une couche de récurrence aux modèles [72, 21]. Ainsi dans les réseaux de neurones récurrents (RNN), la décision au temps t est prise en fonction de l'état de la donnée au temps t et de son état caché au temps t-1. Ces méthodes permettent d'accéder à toutes les images en se libérant des contraintes d'une fenêtre glissante et de la modélisation haut-niveau. Cependant, les RNN ne permettent pas de détecter des mouvements complexes (donc des caractéristiques bas niveau) qui peuvent être critiques dans l'analyse puisqu'elles se basent sur des caractéristiques locales temporellement. Aussi ils sont peu utilisées en pratique, car ils sont difficiles à entraîner [77] et ont une attention limitée sur la durée [89]. De plus, ce pas de un est limitant quand il s'agit de détecter un changement survenant sur un ensemble d'images. Les réseaux de convolution temporelle [82, 52], présentés comme une alternative aux LSTM, encodent et décodent les descriptions en entrée grâce à des convolutions et des agrégations au niveau de la dimension temporelle. De la même façon, les auteurs de l'encodage linéaire temporel (TLE) [20] présentent une nouvelle méthode d'agrégation des descriptions d'images par des modèles bilinéaires ou des couches *fully-connected*. Ces réseaux parcourent entièrement la dimension temporelle et à différentes échelles, pour lier des événements éloignés temporellement. Cependant, il reste difficile à entraîner, comme les réseaux de neurones récurrents, dû à la redondance de la dimension temporelle.

Ensuite, les réseaux de convolutions avec des noyaux en trois dimensions semblent une extension naturelle aux réseaux de neurones en reconnaissance visuelle [96]. D'ailleurs, ils ont été massivement utilisés [41, 95, 100]. Cette méthode crée une représentation hiérarchique spatio-temporelle de la donnée. Comme tous les noyaux ont une dimension supplémentaire, le nombre de paramètres à optimiser est beaucoup plus important, ce qui rend ces réseaux difficiles à entraîner. En l'état, cette architecture ne peut tirer directement avantage d'une initialisation provenant d'un réseau de neurones pré-entraîné sur une grande base d'images comme ImageNet. Ainsi, ces architectures sont souvent peu profondes et entraînées de zéro. Par conséquent, bien que pratique, cette méthode n'atteint les résultats de l'état de l'art que combinée avec la description par IDT.

Enfin, les modèles profonds qui obtiennent les meilleures performances en classification sont les modèles à deux flux (2-stream networks) : un pour l'analyse des images RGB et l'analyse des flots optiques [88]. L'image et le flot optique au temps t sont décrits dans deux réseaux de neurones parallèles. Cette méthode est très pratique, car elle fournit de bonnes performances sur les bases de références. De plus, elle demande peu de ré-entraînement sur la nouvelle tache, car des réseaux de neurones entraînés sur de grandes bases servent d'initialisation pour ces réseaux parallèles. Au cours de son entraînement sur la nouvelle base de flot optique, le réseau s'adaptera rapidement, car l'entrée, bien que décrivant un mouvement, reste spatiale et en deux dimensions. Cette stratégie peut être combinée avec le choix de noyaux 2D ou 3D ou une combinaison des deux ou encore l'utilisation de couches de récurrence pour l'agrégation de décision (cf illu. 1.1). Par conséquent, une combinaison de différents types d'architecture est souvent privilégiée. Les noyaux 3D capturent les motifs de spatio-temporels locaux et les flots optiques apportent une sorte de récurrence, de continuité par leur calcul, notamment grâce aux flots optiques déformés (warped optical flow) [108]. De plus, les réseaux entraînés sur les flots optiques sont moins sujets au sur-apprentissage. Cela peut être dû au fait que le flot optique décrit un mouvement indépendamment de la couleur et la texture initiale, même si la forme de l'objet reste tout de même liée à son mouvement.

La tendance est alors à améliorer l'entraînement de ces réseaux 2-streams pour être de plus en plus performant. Feichtenhofer *et al.* proposent l'ajout de liaisons résiduelles symétriques entre ces deux réseaux permettant alors une communication entre la détection spatiale et la détection de mouvement [29]. Carreira *et al.* proposent une architecture 2-stream appelée Convolution 3D insufflée (I3D) [13] où ils partent d'une architecture image entraînée sur ImageNet et ils étendent les noyaux 2D à une dimension supplémentaire grâce aux noyaux entraînés de sorte à prendre les mêmes décisions puis relancer l'entraînement sur une tâche de classification d'une vidéo. Ainsi, ils peuvent utiliser une architecture type C3D en bénéficiant de l'avantage des entraînements 2D, et ainsi avoir des architectures plus profondes. Ils combinent ce réseau avec une approche double flux qui apporte selon lui une récurrence manquante. Cette combinaison des différents types d'architecture constitue aujourd'hui l'état de l'art dans la classification vidéo, testée

sur les bases de référence UCF101 et HMDB-51. Wang *et al.* proposent le Temporal Segment Network (TSN), un réseaux à deux flux avec un échantillonage uniforme sur la dimension temporelle sur chaque vidéo [108]. Cette stratégie réduit fortement la redondance et donc le coût de calcul, et permet alors un apprentissage sur des bases de données bien plus importante. Ce réseaux a permis de remporter le challenge ActivityNet 2016. Cependant, ce réseau ne prend pas en considération les variations de vitesse d'exécution. Le parcours de la dimension temporelle est saccadé et le mouvement ne peut être totalement décrit. D'ailleurs, les classes les moins reconnues de ce challenge sont "se laver le visage" et "boire un café", alors que les classes les plus reconnues sont "montée de chameau" et "pêcher sur la glace". On remarque cet échantillonnage, bien que pratique pour les réseaux à deux flux, privilégie les descriptions d'éléments visuels caractéristiques plutôt qu'une description de mouvements.

En définitive, la description vidéo et ainsi la relation entre l'espace et le temps est encore mystérieuse. Un atelier mettant en avant des idées originales sur cette relation a eu beaucoup de succès à la dernière conférence CVPR 2017 : *Brave news ideas for motion representation*. De plus en plus d'articles présentent des idées créatives nous invitant à réfléchir autrement. Les mécanismes d'attention [62, 46] ciblent la décision grâce à l'utilisation d'un deuxième réseau prédisant la localisation de l'intérêt dans la vidéo. On peut également citer les méthodes ordonnant les images dans les vidéos et qui se servent de cet ordre pour représenter la vidéo [30], la représentation d'une action comme une transformation d'un état avant à un état après [109] ou enfin les visualisations cherchant à représenter une séquence d'images par une seule image, appelés les réseaux d'images dynamiques (cf illu. 1.2).

Pour conclure, il est nécessaire d'allier une description spatiale à une description temporelle indépendante. En effet, les méthodes les plus performantes utilisent la stratégie 2-stream qui analyse les images et les flots optiques. Edison et al. ont même montré l'intérêt d'analyser également des flots d'accélération [25]. Bien que la relation spatio-temporelle soit étudiée de façon diverse, il n'y a pas de méthodes phares dans cette direction. D'ailleurs, les techniques les plus performantes utilisent une combinaison de plusieurs méthodes. Cela nous montre que cette dimension doit être encore étudiée de sorte à apporter le meilleur modèle possible. Dans cette partie, nous avons étudié les méthodes d'extraction de l'information permettant cette analyse temporelle. Dans la prochaine partie, nous présenterons comment cette analyse est faite dans l'objectif de générer des vidéos qui auront ainsi une cohérence temporelle.

1.2.2 La génération

La génération de vidéos est un secteur de recherche récent et il n'est pas encore au niveau de la génération d'images. Pour la génération d'image, on construit une image d'un vecteur tiré aléatoirement et projeté dans l'espace des images réelles, appelé aussi *manifolds* ou variétés, grâce aux *generative adversarial networks* (GAN).

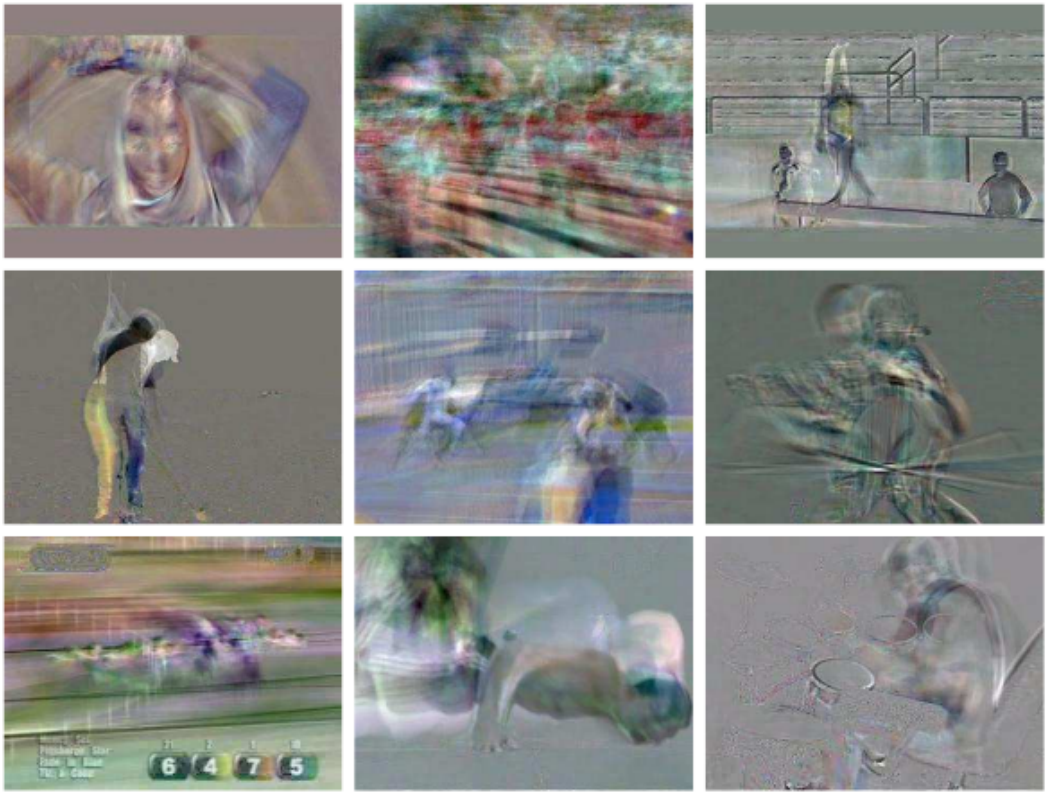


FIGURE 1.2 – Exemples d'images dynamiques (illustration provenant de [8]).

Il existe deux types de générations d'images qui se diffèrent par leur l'objectif de construction.

Interpolation vidéo Tout d'abord, il y a l'interpolation vidéo dans le but de la ralentir, d'obtenir une vue intermédiaire de l'action ou de compresser une vidéo [66, 93]. Ces tâches ont en commun que la sortie de l'algorithme est une image qui sera intégrée ou qui remplacera des images de la vidéo pour en constituer une nouvelle. La prédiction de futures images est un type de génération vidéo qui est souvent une étape intermédiaire à un autre objectif. D'ailleurs, prédire les images futures dans une séquence vidéo est récemment perçu comme un moyen propice à l'apprentissage non-supervisé de contenu vidéo puisque prédire les changements d'une image à l'autre implique que le modèle apprenne une riche description de la séquence d'images précédentes et ces descriptions peuvent être réutilisées pour de la classification [69, 90, 83]. On peut grouper les méthodes pour ces types de générations en deux stratégies : soit les images en sortie sont construites par interpolation en copiant des pixels existants ou par combinaison linéaire de pixels existant ; soit les images sont construites par hallucination de pixels. On parle d'hallucination de pixels quand ce pixel est la sortie directe d'une couche de neurones.

Dans le premier cas, les méthodes interpolent les images à partir des images

d'entrées et de leurs flots optiques, mais cette approche souffre du manque de supervision de la construction du flot optique. D'ailleurs, Szeliski *et al.* [93] présentent la tâche de prédiction comme un bon moyen d'évaluer et donc de superviser les constructions de flots optiques. Liu *et al.* lient donc la construction du flot optique à la prédiction en faisant suivre la couche d'estimation du flot par une interpolation bilinéaire construisant une prédiction de la frame suivante, dans le réseau Deep Voxel Flow (DVF) [65]. Jiang *et al.* [43] proposent Super SlowMo, un réseau d'interpolation en deux parties successives : l'une permettant de prédire des potentielles images intermédiaires, l'autre permettant d'interpoler les images intermédiaires en utilisant les images originales et les images hallucinées de la précédente partie.

Dans le deuxième cas, les réseaux de neurones sont sans interpolation, mais assez complexes pour modéliser au mieux la temporalité des vidéos et être capable d'halluciner les images intermédiaires. Liang *et al.* [63] proposent un réseau de neurones convolutionnel qui encode une séquence d'images de sorte d'une part à halluciner la future image et d'autre part à estimer un flot permettant d'estimer la future image par interpolation. Le système est alors dual et corrige les deux réseaux par backpropagation à partir des deux sorties : le réseau estimant le flot est corrigé par rapport à l'exactitude d'une prédiction d'image suivante à partir de ce flot ; et l'hallucination de la future image sert à produire un flot optique qui sert à corriger le réseau d'hallucination. Le problème de ce type de méthode est que les images produites sont floues, ce qui est un problème récurrent des GAN puisque la distribution naturelle des images suit une distribution multimodale alors que les fonctions utilisées supposent souvent une distribution gaussienne. Ce problème est souvent contré par la création d'un autre réseau dit "adversaire" qui est un réseau discriminant : il indique si l'image en entrée est naturelle ou pas. Ainsi, dans le réseau de Liang *et al.*, il y a deux discriminants, un pour la construction du flot, l'autre pour les images. Cette régularisation de flot améliore la construction de l'image voulue par une interpolation classique provenant d'un flot. À l'inverse de la méthode Super SlowMo qui s'appuie sur une hallucination de pixel pour ensuite faire de l'interpolation ; la méthode Dual Motion GAN part d'une construction de flot et d'une interpolation pour halluciner ses pixels finaux. Enfin la récente méthode de Byeon *et al.* [12] présente un simple LSTM pour halluciner les pixels, mais en modifiant les opérations de couches : ils proposent de réutiliser des unités permettant des couches *fully-connected* mais de façon spatialement hiérarchique. Dans [7], les auteurs proposent une prédiction des contours au lieu des images directement pour palier à ce flou récurrent chez les approches génératives.

Ces deux types d'approches opposent une construction explicite d'une image en intégrant l'information du mouvement de flot optique à l'image précédente, à une construction implicite faite par l'apprentissage du réseau. Les résultats de la première méthode contiennent souvent des artefacts et des dédoublements dans les images, car l'interpolation se base sur le flot qui est souvent imparfait. Tandis que la deuxième méthode produit des images floues malgré l'amélioration grâce à l'ajout de discriminateurs.

Génération directe de vidéo Tout récemment, de nouveaux problèmes de génération de vidéos ont émergé : générer entièrement une vidéo visuellement satisfaisante. Cette tache, très proche de la création d'images, s'illustre dans un problème de génération pure, où l'on crée une vidéo à partir de vecteur aléatoire ; ce qui ramène à l'estimation du *manifold*, l'espace où se trouvent les vidéos. On cherche donc à tirer aléatoirement une vidéo dans cet espace. Les auteurs de VGAN [103] simplifient la tache en supposant un arrière-plan statique : alors, à partir d'un même vecteur aléatoire, un réseau prédit une image qui sera l'arrière-plan et un autre réseau ayant des noyaux 3D prédit les pixels en mouvement. On a alors un réseau à double flux et un discriminateur au niveau de la vidéo indique s'il s'agit d'une vraie vidéo ou non. Cette méthode intéressante subit le désavantage des noyaux 3D qui sont difficiles à entraîner et demandent beaucoup d'espace mémoire. Cela implique que les vidéos générées sont très courtes (1 s). La méthode TGAN [86] crée d'abord l'évolution temporelle par un générateur temporel qui produit une séquence de vecteurs latents, puis chaque vecteur est l'entrée d'un générateur d'images. La séquence des images produites forme la vidéo qui sera évaluée par un dernier réseau. En suivant l'idée que l'on peut séparer le mouvement du contenu d'une vidéo, MocoGAN [99] génère une vidéo en supposant que son contenu est une donnée statique (pour les vidéos courtes) et son mouvement une donnée qui change au cours du temps. Sensiblement comme TGAN, chaque élément de la séquence décrivant la dynamique est associé au contenu pour générer une image et cette suite d'images forme alors la vidéo finale. Deux discriminants permettent d'améliorer la génération : un qui vérifie les images du générateur et un autre qui vérifie la vidéo globale. MocoGAN est évalué par la qualité des vidéos générées évidemment, mais également dans la génération de vidéos ayant le même contenu et des dynamiques différentes et inversement.

Les résultats sont en général peu satisfaisants visuellement, ralentis par les contraintes de mémoire, d'apprentissage et la difficulté de modélisation entre l'évolution temporelle et le contenu. Pour contrer ces difficultés, on se place souvent dans un contexte plus simple ; comme MocoGAN qui n'est évalué que sur des vidéos de visages ou VGAN sur des vidéos à arrière-plan statique. Ainsi d'autres tâches plus simples de génération de vidéo émergent comme le transfert de la dynamique d'une vidéo source à une image cible [4] ou alors la génération de cinémagraphes [126]. Le cinémagraphe est un nouveau type de photographie utilisé surtout en publicité, qui consiste en une image animée d'un léger mouvement répétitif : par exemple, un portrait où la personne resterait fixe, mais clignerait des yeux régulièrement, ou encore, un arbre dont une seule branche bougerait avec le vent. Sans aller plus en détails dans ces méthodes, j'aimerais souligner que ces propositions de génération utilisent des réseaux de neurones complexes. Comme on l'a vu pour la classification, ces réseaux peuvent être à plusieurs flux, composés de noyau 2D et/ou 3D ou encore de couches récurrentes. Ainsi, la méthode MocoGAN propose un réseau alliant RNN et discriminateurs CNN 2D et 3D.

La recherche dans la génération vidéo est très récente, mais on voit que la tendance est de reproduire les descriptions qui fonctionnent en classification, comme

on essaye d'appliquer les techniques efficaces sur les images à la vidéo, ce qui en soi est une bonne stratégie comme on l'a vu pour étendre les descriptions supervisées à des descriptions non-supervisées. Cependant, la génération de vidéo, bien que moins performante que la reconnaissance, propose des idées originales sur la dimension temporelle. Par exemple, en génération vidéo, on aborde l'idée de séparer complètement les mouvements du contenu visuel pour de la génération de nouvelles vidéos. Il est évident que l'objectif de génération d'un objet numérique implique souvent une compréhension globale de l'espace des vidéos et c'est donc une tâche difficile comparée à une compréhension d'une vidéo permet une reconnaissance. Dans le but de caractériser et décrire la relation entre contenu et mouvement, la génération vidéo apporte plus de sophistication à la représentation vidéo. Bien que la génération demande une modélisation complexe pour modéliser la diversité des vidéos, la reconnaissance vidéo permet d'évaluer la représentation d'une vidéo particulière en se focalisant sur la classe à trouver. Puisque, dans cette thèse, nous nous intéressons à une description adaptée aux évolutions temporelles, nous considérerons les taches de classification et plus généralement les tâches d'extraction d'information plutôt que les tâches de génération.

1.3 Difficultés et contributions

L'avancée de la recherche dans l'analyse temporelle est une tâche complexe pour diverses raisons.

Tout d'abord, la différence majeure entre l'espace et le temps se trouve dans la redondance. En effet, une vidéo est une séquence d'images et deux images consécutives contiennent peu d'information distinctes. Ainsi, dans une vidéo de 30 images, il y a moins d'informations que dans 30 images distinctes, mais tout autant de pixels et donc tout autant d'information numérique. Cette redondance rend l'analyse vidéo d'une part coûteuse en temps de calcul et d'autre part, cette redondance rend l'information recherchée à la fois rare et diffuse dans la vidéo. Cette redondance entraîne comme nous l'avons vu dans la section précédente, un échantillonnage pour alléger cette dimension ou un parcours itératif pour l'analyser à un plus haut niveau, où les descriptions seront moins détaillées et donc moins exigeante en place mémoire.

La deuxième particularité de la dimension temporelle se trouve dans la variation de ces évolutions temporelles pour une classe d'action donnée. En effet, le temps est une dimension complexe comparée à la spatialité, car il possède une caractéristique qui lui est propre que l'on nommera l'élasticité temporelle. Lors d'un mouvement, la vitesse d'exécution varie entre deux actions et varie même à l'intérieur même de l'action. Cette variation de vitesse peut être comparée à la variation d'échelle en spatial : une dilation uniforme globale temporelle ralentit une vidéo et une dilation uniforme globale sur les deux dimensions spatiales agrandit une image. À la différence des dimensions spatiales, la dimension temporelle peut être déformée localement : une dilatation partielle provoquera un ralentissement

partiel visuel de la vidéo sans en changer la compréhension, à l'inverse d'une dilatation spatiale qui modifierait les proportions des objets présents. À l'exception d'une perspective importante dans l'image, les objets quelconques n'apparaissent pas à grande échelle en bas et à petite échelle en haut. Pour un exemple visuel, cela transformerait notre vision d'un corps humain en un corps avec une tête aplatie par exemple, ou encore, dans le cas d'une déformation spatiale locale sur les deux dimensions, une pomme pourrait prendre les proportions d'une poire. La nature de l'objet serait modifiée, à l'exception des objets élastiques, d'où le choix du terme "élasticité temporelle" pour nommer cette robustesse à ce type de déformation.

Comme nous l'avons vu, la dimension temporelle est complexe en elle-même, mais d'autant plus que la relation entre la dimension temporelle et les dimensions spatiales est complexe. En effet, on a vu que certaines approches séparent le contenu qui est présent et fixe tout au long de la vidéo (cette personne, cet endroit, cet objet ...) et la dimension temporelle illustrant les mouvements, l'évolution de ce contenu, une séquence d'évolution donc. Ainsi, le sous-espace des contenus modélise l'apparence vidéo indépendante du mouvement tandis que le sous-espace des mouvements modélise l'apparence qui est dépendante du mouvement dans les vidéos. La difficulté réside dans le fait de décrire le sous-espace des mouvements qui est donc différent de l'espace des contenus, mais qui est dépendant de lui. En effet, un mouvement est le changement de localisation d'une apparence. Il est donc difficile de s'intéresser aux mouvements en mettant totalement de côté le contenu.

Des contraintes techniques feront également obstacle à l'objectif de cette thèse. La grande quantité d'information dans un seul échantillon impose que peu d'échantillons peuvent être vus simultanément dans leur ensemble. On pourra donc envisager des représentations intermédiaires pour pallier à cette grande quantité de pixels, en gardant à l'esprit la perte d'information engendrée.

Pour conclure, cette thèse devra faire face à différents challenges. Ainsi, prenant en compte les contraintes de la dimension temporelle dans la vidéo, nous devrons :

- Établir un protocole permettant de sélectionner les informations pertinentes, et ainsi limiter la quantité et la redondance d'information à travers la représentation

- La représentation temporelle choisie doit prendre en compte l'élasticité temporelle en gérant différentes vitesses d'exécution

- L'étude devra prendre en compte le contenu de la vidéo de paire avec le mouvement pour classifier l'information

- Le modèle proposé devra se généraliser à plusieurs exemples contenant diverses informations temporelles

Les principaux intérêts relatifs aux tâches de détection et de classification sur une base de vidéo sont :

- Définir une description implicite ou explicite de l'évolution temporelle

- Identifier les méthodes de classification les plus adaptées aux descriptions fournies

- Évaluer et étudier les résultats dans divers contextes.

1.4 Objectifs

Dans le contexte actuel d'analyse vidéo où l'on se concentre sur les méthodes pertinentes sur les images et où on cherche à s'ouvrir à de nouvelles approches sur l'analyse du mouvement, l'objectif de la thèse est de mettre en avant l'apport de l'information temporelle et des moyens pour l'extraire. Cet apport sera observé dans une tâche de détection et de classification, dans divers contextes contraints de sorte à concentrer l'étude de ces contextes sur l'étude de l'évolution dans la dimension du temps. De plus, nous nous intéresserons également à cette caractéristique particulière de cette dimension, l'élasticité. Puisque cette élasticité crée de la variation intra-classe, nous chercherons à l'étudier et à réduire cette variation pour faciliter la classification. Nous montrons ainsi que l'information temporelle est utile et maîtriser sa description permettrait de réduire le bruit apporté par l'élasticité et de réduire la redondance en repérant les points clés de cette élasticité.

1.5 Structure

Dans le chapitre 2, nous présentons une analyse des mouvements à partir de l'approche classique et explicite de détections et de descriptions modélisées à partir de point clés. Pour se concentrer sur le mouvement au maximum bien qu'il soit dépendant du contenu, nous utilisons les cartes de flots optiques sur lesquelles nous repérons et suivons des singularités. Ce suivi de singularités forme notre descripteur, les singlets, à partir desquels nous interprétons la vidéo. Cette analyse a été réalisée sur des vidéos sportives, plus précisément des vidéos de football, puisque c'est un domaine où justement le mouvement apporte la majorité de l'information. Des classifications et des détections de diverses natures seront expérimentés pour démontrer l'apport de l'information de mouvement à travers cette description proposée.

Ensuite, dans le chapitre 3, nous faisons l'approche inverse et, au lieu d'isoler le mouvement du contenu spatial, nous abordons l'analyse de la relation entre temps et espace de façon implicite. Pour se faire, nous présentons la représentation d'une donnée multi-dimensionnelle sous forme d'un tenseur et les outils mathématiques existants pour étudier ses tableaux complexes multi-dimensionnels. Ainsi à travers ces cubes vidéos, nous voulons extraire les informations communes à une classe pour obtenir une description permettant d'extraire au minimum l'information utile pour reconnaître l'action. Nous choisissons des actions à classifier dont l'information temporelle est primordiale de sorte qu'une bonne reconnaissance soit nécessairement due à une bonne collecte des informations temporelles. Ainsi, on présente un moyen d'extraire et décrire implicitement le contenu lié à une classe. Comme nos classes sont différentiable par leurs mouvements, les informations extraites concerneront les mouvements spécifiques à la classe.

Enfin, dans le chapitre 4, nous nous concentrons sur l'élasticité temporelle. Nous montrons tout d'abord les techniques existantes permettant de contrer cette élasticité temporelle et d'uniformiser les échantillons d'une même classe. Ainsi, on peut diminuer les variations intra-classes et nous montrons que cela permet une classification plus performante.

CHAPITRE 2

Les Singlets

Contents

2.1 Introduction

Dans ce chapitre, nous nous intéressons à l'étude du mouvement à travers le flot optique et nous voulons décrire explicitement l'évolution temporelle. Nous présentons un nouveau descripteur de mouvement basé sur l'extraction de singularités dans le domaine des mouvements dans la section 2.3, qui sont des motifs spécifiques détectés dans le flot optique. Dans la section 2.3.2, nous construisons une description unifiée du flot optique qui nous permet de décrire des aspects différents de la sémantique de la vidéo. Nous appelons ce descripteur **Singlet** lequel correspond au mouvement des singularités à différentes résolutions suivi tout au long de la vidéo. Comme contexte d'abstraction vidéo, nous nous concentrons sur l'application de notre nouveau descripteur de mouvement pour l'analyse sportive, plus spécifiquement des matchs de football. C'est, en effet, un domaine visuel où le mouvement est particulièrement porteur d'information. Dans la section 2.4, des informations diverses comme les zooms et les pics d'activité sont extraits par notre descripteur, puis nous collectons ces informations qui sont des indices pour pouvoir construire un résumé vidéo automatique. Enfin, nous évaluons l'efficacité des singlets dans des contextes variés : matchs de football de la coupe du monde 2014 et la finale du championnat du monde de handball 2015.

2.2 Travaux connexes

Dans la masse de vidéos disponibles aujourd'hui, la demande d'analyse est plus importante dans quatre domaines de vidéos : les réunions, les films, les journaux télévisés et les sports [113]. La masse grandissante de données vidéo disponibles est particulièrement bien illustrée dans le domaine sportif : les chaînes de télévision possèdent et emmagasinent des matchs de compétition de football depuis des années. De plus, dernièrement, la quantité de vidéos sportives disponibles en ligne a grandement augmentée dû à la légalisation du pari sportif. Les algorithmes pour mettre en avant ces vidéos sont alors grandement attendus et les travaux de recherche à ce sujet ont émergé.

Dans l'abstraction vidéo et en particulier dans l'abstraction de vidéo de football, les auteurs n'utilisent généralement pas les représentations de l'état de l'art, les iDT ou les descripteurs de réseaux profonds [106, 56], qui sont construits pour la classification et pas forcément adaptés pour l'abstraction des mouvements. À partir d'un seul match, il y a déjà une grande quantité d'information à extraire dans le but de résumer le match : le nombre de passes, de buts, les statistiques des joueurs .... Par exemple, en se concentrant sur les sessions de jeu et les pauses, on peut filtrer les informations vidéo [26]. Pour détecter les moments de jeu et d'arrêts, Xu *et al.* [115] utilisent des règles heuristiques basées sur les classifications des vues (plan large, plan moyen et plan rapproché) tandis que Xie *et al.* [112] utilisent un modèle structuré et stochastique à travers les modèles de Markov cachés (HMM). Également par une approche stochastique, Leonardi *et al.* [61] proposent des chaînes de Markov contrôlées pour détecter des buts.

En général, la première approche dans l'analyse de vidéo de football est de segmenter la vidéo en plans-séquences et d'extraire les descripteurs. Des descripteurs bas niveaux globaux, comme un histogramme de couleurs par exemple, sont extraits des images et sont souvent associés à des descripteurs faits mains, comme l'orientation des lignes du terrain, le suivi de la balle ou les couleurs des maillots des joueurs. Gong *et al.* [35] utilisent ces caractéristiques pour classifier les événements comme les tirs ou les corners.

Un match de football est plutôt long et ne contient que peu d'actions intéressantes, c'est pourquoi détecter les moments saillants est l'activité de recherche la plus dynamique dans le domaine des vidéos de football. Hanjalic *et al.* [37] utilisent d'autres descripteurs faits main comme l'excitation globale calculée à partir de la longueur des plans-séquences, les activités audio et de mouvements pour caractériser les extraits de jeu. Ces caractéristiques sont choisies et leur impact dans la prise de décision est hiérarchisé par un apprentissage sur la représentation des plans-séquences pour reconnaître les événements : Duan *et al.* [22] utilisent la longueur des plans-séquences et des cartes de texture pour entraîner un SVM tandis que Wu *et al.* [111] utilisent une estimation du mouvement global pour entraîner un réseau de neurones. Sadlier *et al.* [84] se concentrent sur les caractéristiques audio, et des descriptions bas niveaux permettant d'entraîner un SVM. Leur méthode est présentée comme générique pour les sports sur gazon puisqu'ils ne se

concentrent que sur les sports de terrain and donc se basent sur la connaissance de la couleur du terrain (herbe), des lignes de délimitations du terrain et d'autres caractéristiques communes à ces sports de terrain. Yow *et al.* [118] construisent une représentation panoramique d'un plan-séquence et sélectionnent des plans-séquences importants en utilisant des heuristiques.

Les modèles de machines à états finis sont également utilisés pour déterminer si un événement est saillant ou non. Assfalg *et al.* [3] utilisent des machines à états finis basées sur les mouvements de la balle, les positions des joueurs et les maillots., tandis que Tabii *et al.* [94] utilisent des machines à états finis basées sur la segmentation du terrain de jeu, la détection des plans-séquences et la classification des vues. Cependant, ces méthodes requièrent des règles manuelles et précises. Ye *et al.* [117] entraînent un SVM à apprentissage incrémental à partir de descriptions haut niveaux contenant les descripteurs des moments successifs comme les types de vues, la position des lignes et les descripteurs des plans-séquences pour sélectionner les temps forts.

Après avoir utilisé un réseau de neurones artificiel pour détecter l'apparence des logos et du score et après avoir utilisé la méthode de Hough et la méthode des plus proches voisins (K-means) pour détecter l'ouverture du but, Zawbaa *et al.* [119] classifient les buts, les attaques et d'autres événements avec un SVM entraîné sur la représentation de l'ouverture du but. Dans un article récent [81], les plans-séquences et les images clés associées sont triés en utilisant des détections de visages et de peau, un détecteur de sifflet et les indications d'utilisateurs.

Toutes ces méthodes utilisent beaucoup plus de caractéristiques visuelles pour créer des résumés et détecter les activités saillantes d'un match. Au mieux une fonction d'énergie sur ces caractéristiques visuelles (longueur des plans-séquences par exemple) est utilisée pour mesurer l'activité. Or, on sait que le mouvement apporte de l'information utile et nécessaire et réduire l'activité à des critères visuels simplifie et dégrade l'information. Nous présentons alors notre représentation générique pour décrire le mouvement et montrer comment cette représentation nous permet de construire une abstraction de la vidéo.

2.3 Analyse des mouvements en vidéo

Notre méthode d'analyse de mouvement est inspirée du travail de Kihl *et al.* [49] qui extrait des singularités du mouvement dans le domaine des fluides mécaniques. S'inspirer d'un travail sur la description du mouvement des fluides est pour nous un moyen d'étudier le mouvement en s'éloignant un peu plus de l'apparence de ce qui est en mouvement. Une singularité est un point d'annulation similaire aux racines pour les polynômes. Il peut être vu comme un point clé dans le flot optique. Le flot optique est projeté sur l'espace des fonctions polynomiales bi-variables de sorte à repérer ces points d'annulation de son approximation polynomiale. En partant de la définition de ces singularités, nous avons construit un nouveau descripteur local de mouvement dans les vidéos.

2.3.1 Projection polynomiale du flot optique et Singularités

Les deux composantes horizontales et verticales du flot optique U et V de chaque pixel (x_1, x_2) sont calculées en utilisant la méthode de Gunnar Farneback [27]. U et V sont projetés sur la base de Legendre de degré d pour obtenir la meilleure approximation dans l'espace des flots optiques polynomiaux. Ensuite, ils sont exprimés dans la base canonique.

$$P_U(x,y) = \sum_{k=0}^{K}\sum_{l=0}^{L} u_{k,l}.x^k.y^l ; P_V(x,y) = \sum_{k=0}^{K}\sum_{l=0}^{L} v_{k,l}.x^k.y^l \tag{2.1}$$

avec $K+L \leq d$. Comme dans [49], nous restreignons les approximations au degré $d = 1$.

$$\begin{pmatrix} U \\ V \end{pmatrix} \simeq \mathbf{A}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \mathbf{b} = \begin{pmatrix} a_{11}x_1 + a_{12}x_2 + b_1 \\ a_{21}x_1 + a_{22}x_2 + b_2 \end{pmatrix} \tag{2.2}$$

Selon $\mathbf{A}$ et $\mathbf{b}$, les singularités du flot optique apparaissent à la position $(x_1\ x_2)^T = -\mathbf{A}^{-1}\mathbf{b}$. Le type de la singularité dépend de $tr(A)$ et de $\Delta(\mathbf{A})$.

$$\Delta(\mathbf{A}) = (tr(\mathbf{A}))^2 - 4\ \det(\mathbf{A}) \tag{2.3}$$

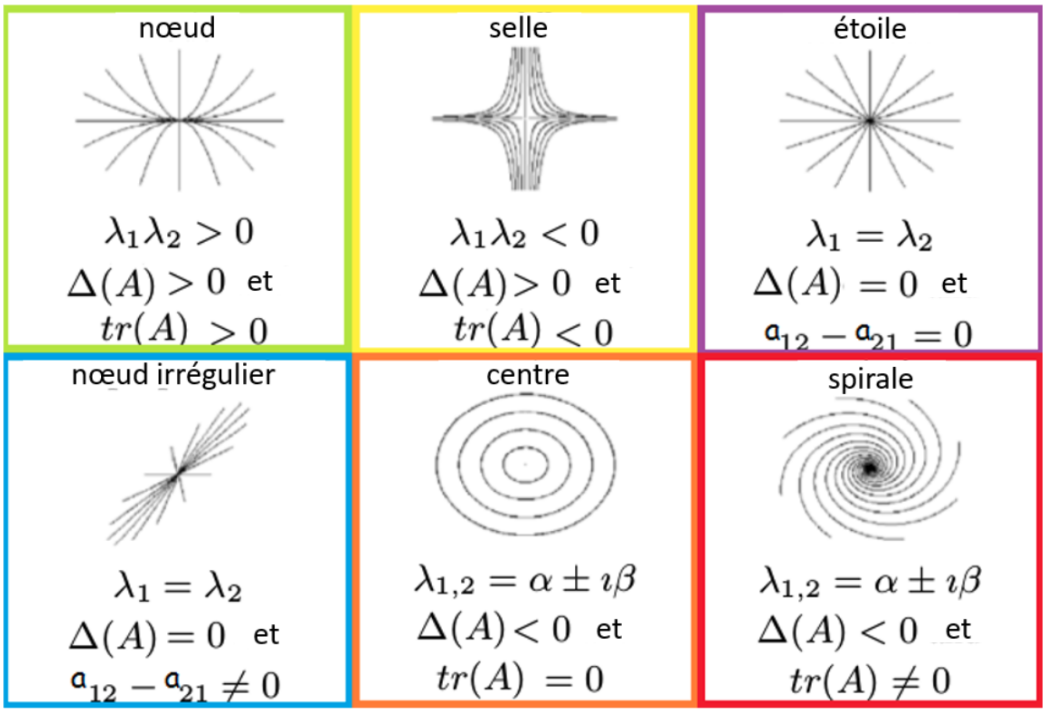


FIGURE 2.1 – Classification des singularités selon les valeurs de $\mathbf{A}$ (illustration construite à partir d'une illustration provenant de [49]).

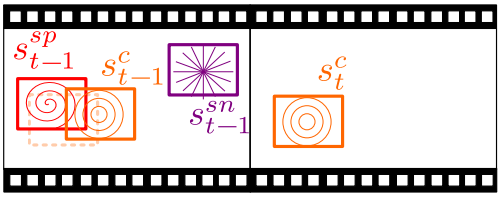


FIGURE 2.2 – Deux flots optiques successifs : Cherchant une correspondance pour la singularité s^c_t parmi les singularités du flot optique précédent. Concernant le ratio de chevauchement, s^{sp}_{t-1} et s^c_{t-1} sont candidates alors que s^{sn}_{t-1} ne l'est pas. La singularité correspondante est la singularité la plus proche en fonction de la position et du type : s^{sp}_{t-1}. Les types de singularités sp, sn et c correspondent respectivement à spirale, étoile et centre.

Nous présentons les différentes configurations de singularités dans l'illustration 2.1 : noeud, selle, étoile, noeud irrégulier, centre et spirale.

La méthode permet de détecter au maximum une singularité sur l'approximation. Pour étudier les cas où deux singularités sont présentes, il faudrait complexifier l'approximation en augmentant le degré d. Les singularités sont détectées en utilisant une fenêtre glissante à différentes échelles (de $0.1h$ à $0.5h$ par pas de $0.1h$ où h est la plus petite dimension de la vidéo, généralement la hauteur). Ainsi, plusieurs singularités peuvent être détectées à la même position, mais à différentes échelles. Ces singularités multiples n'ont pas été fusionnées, comme pour les descripteurs SIFT, puisqu'elles peuvent être de différents types ou de différentes intensités et donc apporter des visions du mouvement à différentes échelles et complémentaires.

Par définition, une singularité correspond à un point d'annulation de l'équation 2.3. Donc aucune singularité ne sera détectée en cas de translation pure. Une singularité n'est pas détectée dans des situations où $\mathbf{A}$ n'est pas inversible. À l'inverse, plus le déterminant est élevé, plus la singularité est significative : un seuil sur $\det(\mathbf{A})$ permet donc de filtrer les singularités du bruit et de garder celles qui sont les plus informatives. Sans seuillage, toutes les singularités seront conservées, celles avec un très faible mouvement et qui sont potentiellement produites par du bruit de compression ou autre. Ce seuillage permet ainsi de contrôler la sensibilité du détecteur de singularités.

Par la suite, nous extrayons ces singularités globalement ou localement à travers diverses tâches.

2.3.2 Représentation Spatio-Temporelle

Comme introduit plus tôt, les singularités peuvent être suivies le long d'une vidéo pour décrire un plan-séquence. Pour T images, $T-1$ flots optiques sont calculés. Sur chacun de ces flots optiques, nous extrayons les singularités. Dans le but de construire des chaînes de singularités, appelées *Singlets*, nous les avons suivies dans le sens inverse au temps. En se basant sur l'hypothèse qu'une singu-

larité a un faible déplacement entre deux images et pour réduire le coût de calcul, nous mettons en correspondance seulement les singularités dans un proche voisinage. Donc, pour chaque singularité $sing_s$ dans le flot optique f_t, les singularités candidates dans le flot optique f_{t-1} sont restreintes à un voisinage proche $V(sing_s)$. Deux singularités sont considérées dans le même voisinage si leurs fenêtres glissantes ont un ratio de chevauchement approprié (figure 2.2) comme défini ci-dessous :

$$V(sing_s) = \left\{ sing; \frac{area(W(sing) \cap W(sing_s))}{area(W(sing) \cup W(sing_s))} > \alpha \right\} \tag{2.4}$$

où $W(sing_s)$ est la fenêtre glissante de $sing_s$.

Pour mettre en relation les singularités le long de la vidéo, le meilleur candidat dans un voisinage proche est sélectionné comme celui qui minimise la distance entre singularités décrite ci-dessous :

$$d(\begin{pmatrix} \mathbf{A} \\ x \\ y \end{pmatrix}, \begin{pmatrix} \mathbf{A}' \\ x' \\ y' \end{pmatrix}) = ||A - A'||_F + \lambda|| \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x' \\ y' \end{pmatrix} ||_2 \tag{2.5}$$

où λ est le poids qui permet d'équilibrer l'apport d'information entre la distance entre les positions et la similarité entre les valeurs des coefficients des singularités. Dans chaque flot optique affine f_t, une singularité $sing_s$ est décrite par ces 6 coefficients, 4 dans $\mathbf{A}$ et 2 dans $\mathbf{b}$, et la distance entre les singularités prend en compte ces caractéristiques.

— Puisque $\mathbf{b}$ contient l'information au sujet de la position du centre de la singularité à l'intérieur de sa fenêtre $W(sing_s)$, nous convertissons ce centre normalisé en position de pixel dans la vidéo (x, y).

— Puisque $\mathbf{A}$ contient l'information au sujet du type de la singularité (figure 2.1), comparer les coefficients de $\mathbf{A}$ et $\mathbf{A}'$ permet de comparer les champs vectoriels des singularités et donc leur type et leur aspect.

L'algorithme complet pour extraire les singularités et les traquer comme une singlet est décrit dans l'algorithme 1. Un exemple de singlet est présenté dans l'illustration 2.3.

Extraire les singlets tout au long d'une vidéo et analyser leur forme, position et leur longueur nous fournit une description robuste du flot optique de cet extrait. Dans la prochaine partie, nous détaillons nos applications des descriptions des singlets sur une vidéo de football pour résumer des matchs de sport.

2.4 Abstraction vidéo

Dans cette section, nous allons construire des méthodes d'abstraction vidéo pour extraire des caractéristiques de plus haut-niveau comme des zooms ou de

pour chaque *image du plan-sequence* **faire**
  obtenir l'image f_n;
  si *l'image précédente existe* **alors**
    Construire le flot optique$_n$=(U,V) de f_{n-1} et f_n;
    pour chaque *fenêtre glissante* **faire**
      Rogner le flot optique à l'intérieur de la fenêtre;
      Projeter chaque composante U et V en faisant le produit scalaire de chaque composante avec chaque élément $P_{i,j}$ de la base de Legendre;
      changement de base pour obtenir **A** et **b**;
      détecter si il y une singularité et ses caractéristiques;
    fin
  fin
  pour chaque *singularité détecté sing$_s$ dans le flot$_n$* **faire**
    pour chaque *singularité du flot$_{n-1}$ à l'intérieur V(sing$_s$)* **faire**
      trouver celui qui avec la meilleur distance entre leur **A** et les positions des singularités;
    fin
    si *la meilleur distance est en-dessous de 2* **alors**
      si *le meilleur candidat est à la fin d'une singlet* **alors** ajouter sing$_s$ dans cette singlet;
      sinon créer une singlet avec les deux singularités.;
      si *pas de correspondance trouvée à l'intérieur du flot$_n$* **alors**
        chercher une correspondance dans le flot précédant jusqu'à un historique de 5 flots
      fin
    fin
  fin
fin

Algorithme 1 : Algorithme d'extraction des singlets

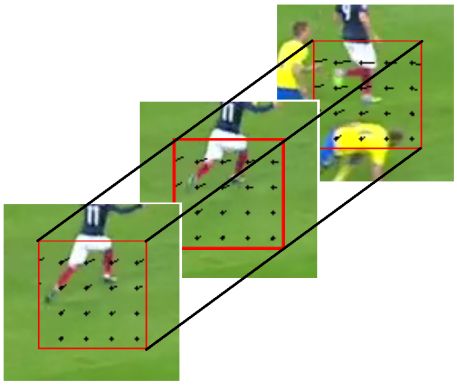


FIGURE 2.3 – Singlets : illustration du suivi des singularités extraites du flot optique sur 3 images consécutives dans un match de foot. C'est une singularité spirale comme cela peut être vu dans le flot.

l'agitation à partir de nos singularités et de nos singlets. Ces méthodes sont évaluées dans ces tâches d'extraction en comparaison à l'état de l'art. Nous montrons ainsi l'intérêt de nos descripteurs dans l'extraction d'information concernant le mouvement.

Ces caractéristiques haut-niveaux ont été choisies dans le but de repérer les moments saillants du match. Un moment saillant est défini ici comme un événement, un moment assez important pour être dans le résumé du match : soit une faute, un corner, un but ou simplement un tir.

2.4.1 Les données

Comme indiqué précédemment, une grande quantité de vidéos sportives sont présentes en ligne et stockées en archives. Cependant, il n'y a pas encore de bases de données construites et partagées en vu d'une tâche de classification dans le domaine de l'analyse de sport. En conséquence, nous avons enregistré nos propres vidéos provenant de chaînes HDTV. Nous avons annoté manuellement les zooms et les ralentis de 4 matchs de football de la Coupe du Monde FIFA 2014 (Germany vs Portugal, Nigeria vs Argentine, France vs Honduras, Switzerland vs France) et la finale de la Coupe du Monde 2015 de handball au Qatar. Chaque vidéo a été normalisée à 25 fps.

Pour évaluer notre détection de moments saillants, nous avons extrait la vérité terrain pour les 4 matchs sélectionnés (Germany vs Portugal, Nigeria vs Argentine, France vs Honduras, Switzerland vs France) à partir de listes de moments saillants sélectionnés sur le site officiel de la FIFA [17]. Comme on peut le remarquer sur ces listes(cf illustration 2.4), la description de chaque événement saillant est donnée à une minute donnée du match et elle décrit brièvement l'action. Nous avons annoté manuellement les moments saillants de la description de la FIFA avec le numéro de l'image de début et de fin pour obtenir un niveau de description à l'image.

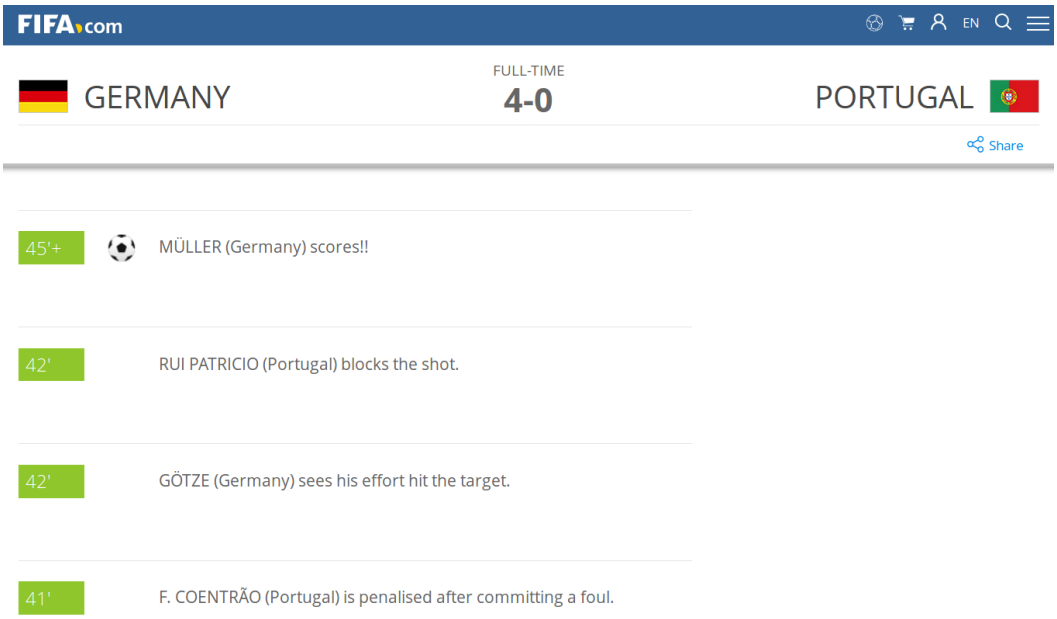


FIGURE 2.4 – Extrait de la liste de moments saillants sur le site officiel de la FIFA. On remarque que les deux actions de la 42^{ieme} minute, un tir cadré suivi d'un arrêt du gardien, désigne un même moment saillant.

Nous avons amélioré l'annotation de cette base pour la rendre conforme à un travail de vision par ordinateur. Tout d'abord, deux instances d'une même action peuvent être parfois annotées positivement et parfois négativement, saillant ou non-saillant. Or, dans notre cadre d'étude, nous voulons nous évaluer dans un cadre de label certain pour évaluer notre description de mouvement. Il est à noté que le domaine de la classification à partir de label probabiliste pourrait apporter de la sophistication dans un apprentissage à partir de données subjectives, comme il en est ici le cas. Dans notre cadre d'étude, nous avons complété la liste des moments saillants en ajoutant tous les corners et tous les coups d'envoi pour ne manquer aucune action. Deuxièmement, un enchaînement rapide d'actions saillantes est considéré comme une action saillante. Nous avons ainsi fusionné les moments saillants qui sont listés deux fois ou très proches temporellement : par exemple, un but depuis un coup d'envoi listé comme une action de coup d'envoi en premier lieu puis par une action de but. Nous appelons la vérité-terrain formée par cette liste complétée de moments saillants : vérité terrain étendue. Dans le tableau 2.1, nous reportons la quantité de moments saillants dans chaque match et dans chaque base.

Dans cette base de données, il y a plus de 7 heures de vidéos, précisément 696002 images. Dans l'intérêt d'une recherche reproductible, toutes nos méta-données et notre implémentation sont disponibles en ligne [1].

1. http://www.i3s.unice.fr/~kblanc/

Match	Base de données FIFA	Base de données Étendue
Germany vs Portugal	30	27
Nigeria vs Argentine	51	35
France vs Honduras	54	32
Switzerland vs France	40	26

TABLE 2.1 – Nombre de moments saillants dans chaque match selon la vérité terrain de tous les moments saillants sur le site officiel de la FIFA par rapport à notre vérité terrain étendue.

2.4.2 Détection de zooms

Dans un match de foot provenant d'une chaîne de télévision, les zooms, et en particulier les zooms en avant, sont d'efficaces indicateurs de temps forts et moments saillants du match, puisqu'ils représentent une réaction naturelle du cameraman face à une action importante.

Pour évaluer notre méthode, nous utilisons une vidéo de la moitié d'un match de football, *i.e.* plus de 128 500 images. Nous avons annoté seulement les zooms persistants lesquels sont importants en intensité et en temps. Nous avons exactement 5 659 flots optiques positifs et 122 841 flots optiques négatifs. Pour comparer chaque méthode, nous utilisons des courbes ROC (illustration 2.5 et 2.6) lesquelles illustrent bien les performances même dans le cas de classes non équilibrées.

Les deux singularités *étoile* et *noeud irrégulier* (illustration 2.1) représentent des zooms. Deux conditions sont nécessaires à cette détection : un fort déterminant $\det(\mathbf{A})$ et un faible $\Delta(\mathbf{A})$ (cf eq. 2.3). $det(A)$ correspond à l'intensité du flot optique. Dans l'illustration 2.5, nous évaluons l'influence du seuil sur $\det(\mathbf{A})$ par rapport à la performance de la détection de zoom : aucun seuil sur $\det(\mathbf{A})$ implique que chaque mouvement global est un candidat au zoom, même les plus légers, et, comme on peut le deviner, cette stratégie donne peu de résultats ; avec un seuil sur $\det(\mathbf{A})$, les résultats sont stables. Puisque les projections sont toutes calculées à partir de flots optiques et dans le but d'obtenir une détection avec une cohérence temporelle proche de la perception humaine, nous ajoutons une fenêtre chronologique pour moyenner ces $\Delta(\mathbf{A})$ au cours du temps. Après une analyse sur la taille de la fenêtre de 5 à 100 images, les meilleurs résultats sont obtenus avec un historique de 30 images, ce qui correspond environ à une seconde. Nous obtenons ainsi les meilleurs résultats avec la moyenne chronologique, un seuil de 0.2 sur $\det(\mathbf{A})$ et un seuil de 4 sur $\Delta(\mathbf{A})$ (illustration 2.5).

Généralement, un zoom est détecté par la méthode d'Estimation de Mouvement Global (GME) [117]. Pour cette méthode, nous utilisons la méthode RGMC [85] pour calculer l'homographie h_t qui modélise le mouvement de la caméra pour chaque image t.

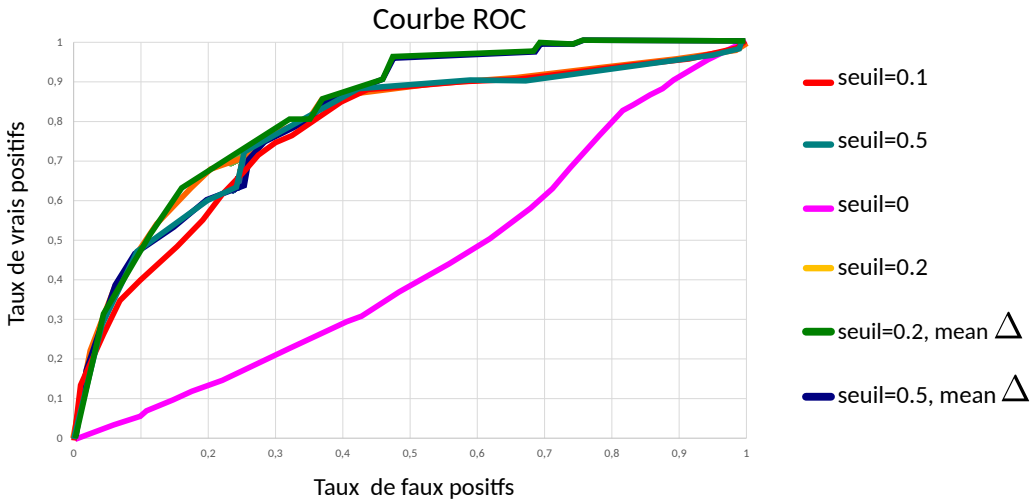


FIGURE 2.5 – Courbes ROC de notre méthode pour les différents seuils sur $\det(\mathbf{A})$. Le meilleur résultat est obtenu avec un seuil de 0.2 sur $\det(\mathbf{A})$ et un $\Delta(\mathbf{A})$ moyenné sur 30 images.

$$x' = m_0 x + m_1 y + m_2$$
$$y' = m_3 x + m_4 y + m_5$$

Selon Qian *et al.* [79], un zoom est détecté si m_0 et m_4 sont égaux. Nous mettons alors un seuil sur leur différence. La meilleure valeur de seuil trouvée pour la méthode GME est 0.0004.

Duan *et al.* [23] quantifient les vecteurs de mouvement pour produire deux histogrammes, un sur les angles et un sur les longueurs. Un zoom est détecté si il y assez de vecteurs avec une norme faible et assez de vecteurs avec des angles compris dans $[15; 75] \cup [105; 165] \cup [195; 255] \cup [285; 345]$. Ainsi, cette méthode demande deux seuils qui ne sont pas fournis. Sur nos données, la détection donne les meilleurs résultats sans le filtre sur l'histogramme des longueurs des vecteurs. Cela nous à mener à nous focaliser sur l'histogramme des angles. La meilleure valeur de seuil pour la méthode Duan sur l'histogramme des angles est de 42% du nombre total de pixels.

Nous comparons notre méthode et ces approches selon la meilleure configuration pour chacune. Toutefois, grâce aux illustrations 2.5 et 2.6 conjointement, notons que notre méthode n'est pas très sensible à la configuration et que la plupart des configurations fournissent une meilleure détection que GME et celle de Duan.

Dans le tableau 2.2, nous présentons les taux de reconnaissance de chaque méthode avec la meilleure configuration. Puisque notre base de données est déséquilibrée avec 5% d'exemples positifs, les valeurs de précision sont faibles. Elles indiquent toutefois la proportion de faux zooms détectés par le classifieur.

Notre méthode de détection de zoom possède trois avantages. Le premier avantage est la localisation du centre du zoom qui est un indicateur de la position où se

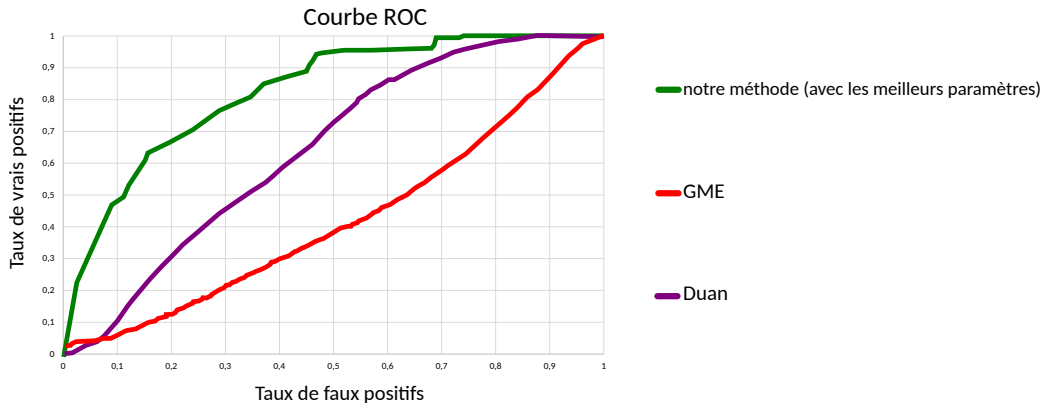


FIGURE 2.6 – Courbes ROC pour chaque méthode pour la détection de zoom : notre méthode vs la méthode de Duan vs la méthode GME.

Méthode	Précision	Rappel	Exactitude
GME	3.68 %	68.4 %	19.79 %
Duan	8.92 %	50.62 %	75.06 %
ours	19.45 %	63.47 %	86.82 %

TABLE 2.2 – Précision, rappel et taux de reconnaissance pour la détection de zoom.

pour chaque *image dans le plan-séquence* **faire**
  obtenir l'image f_n;
  si *l'image précédente existe* **alors**
    construire le flot optique $flot_n$=(U,V) à partir de f_{n-1} et f_n;
    projeter chaque composante U et V en faisant le produit scalaire de chaque composante avec chaque élément $P_{i,j}$ de la base de Legendre;
    changement de base pour obtenir $\mathbf{A}$ et $\mathbf{b}$;
    si $|\det(\mathbf{A})| > 0.2$ *et* $|\Delta(\mathbf{A})| < 4$ **alors**
      zoom détecté
      position du centre du zoom à $-\mathbf{A}^{-1}\mathbf{b}$ **si** $trace(\mathbf{A}) < 0$ **alors**
        zoom-arrière détecté;
      sinon zoom-avant détecté;
    fin
  fin
fin

Algorithme 2 : Détection de zoom

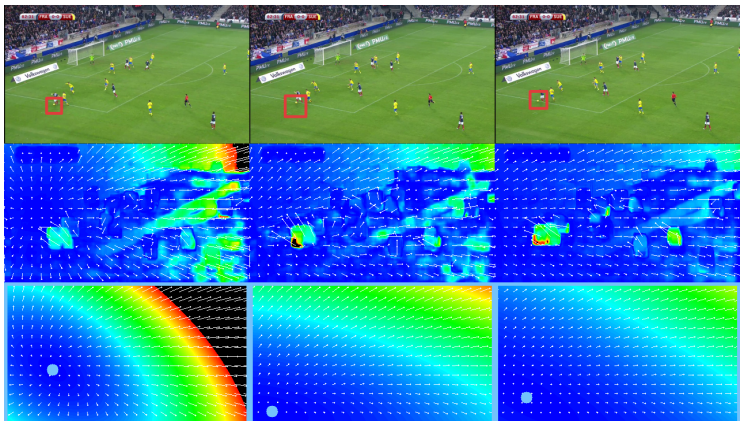


FIGURE 2.7 – Un zoom détecté dans un match de football. De gauche à droite : 3 images consécutives. De haut au bas : images originales (la balle est mise en avant par un carré rouge), flots optiques originaux et approximations polynomiales des flots optiques. Sur la dernière ligne, le centre de la singularité est représenté par un point bleu clair et il correspond à la balle comme vous pouvez le voir sur la première ligne.

passe l'action dans l'image. Le second avantage, et l'avantage principal de notre méthode, est que les zooms sont détectés même si le centre du zoom ne se situe pas au centre de l'image. Comme on peut le remarquer sur l'illustration 2.7, le centre du zoom (représenté comme un point bleu clair sur la troisième ligne) est très loin du centre de l'image et il est tout de même détecté. À notre connaissance, un tel résultat ne peut être atteint seulement avec notre approche. Enfin, le dernier avantage est de facilement différencier les zooms avant des zooms arrière. Dans le cas de ces singularités, *étoile* et *noeud irrégulier*, les valeurs propres de $\mathbf{A}$ sont égales. Nous vérifions donc simplement le signe d'une des valeurs propres pour différentier les zooms avant et les zooms arrières. Cette méthode est très efficace pour détecter les zooms n'importe où dans la vidéo.

Par conséquent, extraire une singularité globale sur le flot optique entier peut fournir des informations d'édition de la vidéo telles que les zooms. Un zoom peut également être détecté localement dans le cas où un objet grossirait dans une scène ou dans le cas où l'objet se rapprocherait de l'objectif de la caméra. Extraire des singularités locales apporte une sophistication dans la description et peut aider à mesurer la quantité de mouvement.

2.4.3 Agitation globale

Nous faisons l'hypothèse qu'un moment saillant dans les sports d'équipe se produit quand plusieurs joueurs sont en conflit pour obtenir le ballon ou quand des joueurs courent vers le but (phase d'attaque). Dans ces cas, il y a des singularités locales autour et sur ces zones de mouvement et ces singularités ont une

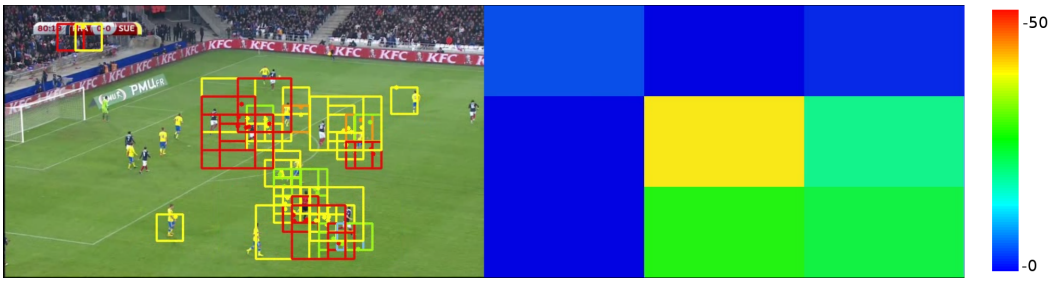


FIGURE 2.8 – Gauche : un tir avec toutes les singularités détectées. Chaque carré coloré correspond à une singularité, avec la couleur de leur type (illustration 2.1). Droite : leurs histogrammes spatio-temporels exprimés grâce à une échelle de chaleur indiquant la quantité de singularités par zone de l'espace découpé en 3x3 sous-images.

consistance temporelle.

Nous détectons les singularités dans chaque flot optique en utilisant une fenêtre glissante et nous construisons un histogramme spatial sur leur position. Notre but est de détecter les régions où il y a une agitation globale. Chaque image est découpée spatialement en 3 par 3, ce qui donne 9 régions uniformes (voir illustration 2.8), sur lesquelles nous comptabilisons les singularités. Nous obtenons alors un histogramme de singularités en fonction de leur position dans l'image.

Pour stabiliser les histogrammes spatiaux au cours du temps, nous les sommons à l'intérieur d'une fenêtre temporelle de 10 images. Nous mettons de côté les régions où le tableau de score est affiché et nous sommons les cases restantes. Nous choisissons cet indicateur pour sélectionner les moments pour un résumé et nous trions les événements en fonction de leur intensité calculée grâce à l'hypothèse formulée précédemment. Ces histogrammes sont particulièrement intéressants quand ils sont extraits lors d'une vue éloignée du match, en plan large.

D'autres signes comme les rediffusions d'action (ralentis) indiquent qu'une action importante vient juste de se produire.

2.4.4 Détection de ralentis

Quand une action rapide mérite d'être détaillée, les producteurs utilisent généralement les ralentis, souvent d'un nouveau point de vue pour voir tous les détails. Donc nous pouvons détecter les moments saillants du match en repérant ces ralentis. La plupart des solutions proposées pour ce problème de détection utilisent les habitudes de productions vidéos qui consistent généralement à mettre un logo avant et après chaque rediffusion. Pan *et al.* [76] utilisent un algorithme de modèle de Markov caché (HMM) pour détecter les ralentis à partir des différences entre les images et un HMM pour modéliser les états avant et après le ralenti, comme les effets d'édition pour les transitions entre plans. Zawbaa *et al.* [119] filtrent chaque image de la vidéo avec la couleur dominante du logo et un séparateur à large marge (SVM) entraîné prédit si le logo apparaît ou non dans l'image.

Base de données	Précision	Rappel	Reconnaissance
Entraînement	97.06 %	80.49 %	89.41 %
Test	76.32 %	87.88 %	79.36 %
Test sur le handball	100 %	20 %	60 %

TABLE 2.3 – Précision, rappel et reconnaissance pour la détection de ralenti sur les différentes parties de la base de données.

Ce type de méthode permet d'éviter le problème de caractérisation d'un mouvement lent, mais elle apporte le désavantage de devoir construire un ensemble d'apprentissage pour chaque logo.

Pour détecter un ralenti, nous nous concentrons sur l'intensité lente du mouvement. Notre méthode est basée sur une hypothèse simple qui est : dans un ralenti, un mouvement sera plus consistant au cours du temps. Une singlet décrit une évolution typique du mouvement jusqu'à ce qu'il change rapidement d'aspect ou de position. Ainsi, la longueur des singlets est proportionnelle à la longueur de l'action et la présence plusieurs longues singlets indique que le plan est un ralenti.

Nous calculons la longueur de chaque singlet et stockons leur longueur sous forme d'un histogramme de 100 classes. L'histogramme est ensuite considéré comme un vecteur de description de la lenteur de la vidéo. Pour détecter les ralentis dans la vidéo, nous entraînons un SVM avec un noyau gaussien après un blanchiment et une analyse en composantes principales (PCA) sur notre descripteur de mouvement lent. Pour entraîner notre SVM, nous utilisons une base de données de 82 vidéos contenant 41 ralentis et 41 vidéos à vitesse normale. Ces vidéos sont extraites de 3 matchs de football. Dans nos expériences, nous avons fixé $\alpha = 0.4$ pour la sélection du voisinage (cf eq 2.4), $\lambda = 0.02$ pour la distance de mise en correspondance des singularités (cf eq 2.5) et $C = 29$ et $\gamma = 4.10^3$ pour l'entraînement du SVM.

L' ensemble de test est composé de toutes les autres vidéos de ralentis et vidéos normales dans chacun des matchs, ce qui fait 33 plans ralentis et 33 plans non ralentis. Pour évaluer la capacité de généralisation de notre descripteur, nous ajoutons 5 ralentis et 5 vidéos normales extraites d'un match de handball.

Le tableau 2.3 présente nos résultats de reconnaissance sur chaque ralenti. Comme vous pouvez le voir, nous obtenons un fort de taux de reconnaissance pour la classification des ralentis de football. Avec la même logique d'extraction du descripteur et sans ré-entraîner le SVM, nous obtenons une détection de ralentis de haute précision sur les vidéos de handball. Puisque notre système a été entraîné sur des matchs de football et puisque nous n'avons pas changé de paramètres, selon les résultats sur le match de handball, il est, en effet, générique en dépit de l'effet.

Dans la prochaine section, nous décrivons comment les singlets sont utiles à travers ces détections pour extraire les moments saillants d'un match et construire un résumé.

2.4.5 Détection de moments saillants et Synthèse de match

Les singularités et les singlets représentent les régions en mouvement. Extraites globalement, les singularités représentent le mouvement de la caméra : nous utilisons l'étoile et le noeud irrégulier pour repérer les zooms dans la section 2.4.2. Extraites localement, elles représentent les mouvements des joueurs et de la balle : nous utilisons la quantité de singularités pour caractériser globalement l'agitation. Une singlet identifie l'évolution d'une singularité au cours du temps : nous utilisons leurs longueurs pour remarquer les ralentis par leur nature lente.

À partir de ces descriptions de singlet, nous construisons un résumé. Nous sélectionnons les meilleurs moments par une combinaison de plusieurs zooms avants et arrières, suivis par un pic dans l'histogramme spatio-temporel des singularités et ensuite une rediffusion de l'action ralentie.

Dans le but de combiner ces différents indices sur les moments saillants, nous construisons une frise pour chaque détection. Dans l'illustration 2.9, vous pouvez remarquer sur la gauche une vidéo de football et sur la droite son histogramme spatio-temporel de singlets associé. En dessous, de haut en bas, il y a 5 frises. La première frise correspond aux zooms extraits : rouge pour les zooms avant et bleu pour les zooms arrière. La seconde frise montre la quantité de changement de zooms durant une seconde en fonction d'une échelle de chaleur. La frise suivante montre la quantité de vecteurs forts également en utilisant une échelle de chaleur. Nous utilisons l'indicateur de lumière pour différentier les vues de près des vues en plan large. La quatrième frise indique la classification du SVM pour les ralentis dans les vues de près : rouge pour le ralenti et bleu pour la vidéo à vitesse normale. Enfin, la dernière frise indique la quantité de singularité : l'indicateur d'agitation globale.

Une **synthèse du match** est alors la concaténation des moments saillants détectés. Un moment saillant est détecté si à l'intérieur d'une fenêtre de 30 secondes, il y a :

— au moins deux changements de direction du zoom, et

— un pic d'activité plus haut que 1500 (au moins 1500 singularités) d'un plan large, et

— une rediffusion ralentie dans une vue proche.

Par conséquent, nous extrayons une liste de moments principaux du match avec ces règles. Sur ces quatre matchs, notre méthode obtient les résultats référencés dans le tableau 2.4.

Malheureusement, les auteurs de méthode de synthèse pour les matchs de football ne fournissent pas de sources de leur implémentation ni de binaire exécutable pour reproduire leurs résultats sur notre base de données et la ré-implémentation implique soit des spécifications d'utilisateurs humains, soit des seuils non fournis ou une collection de bases de données de logo. Cela rend la comparaison directe impossible pour la construction de synthèse.

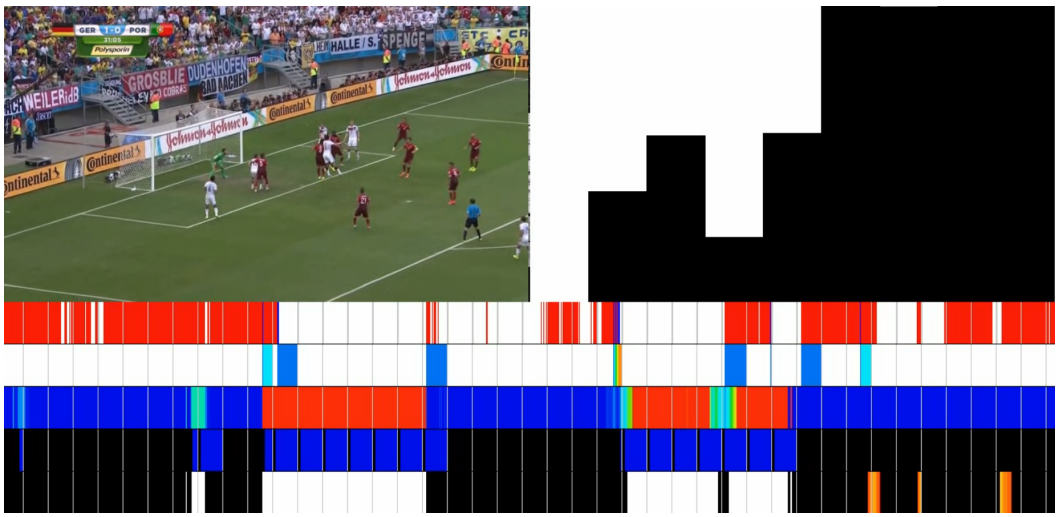


FIGURE 2.9 – En haut à gauche : image aléatoire d'un match de football. En haut à droite : son histogramme spatio-temporel. En bas : 5 frises qui décrivent les critères de détection de moments saillants (de haut en bas : zooms avant/arrière, le compteur de changement de zoom, la saturation, la détection de ralenti et le niveau d'activité).

Match	Base de donnée FIFA	Base de donnée étendue
Germany vs Portugal	80 %	88.9 %
Nigeria vs Argentine	53 %	77.2 %
France vs Honduras	53.7 %	90.7 %
Switzerland vs France	62.5 %	96.6 %
Moyenne	62.3 %	88.2 %

TABLE 2.4 – Taux de précision de la détection de moments saillant par notre méthode sur tous les moments saillants, en utilisant deux bases de données différentes (voir paragraphe 2.4.1).

FIGURE 2.10 – Image de l'extrait du match de handball.

Nous pouvons résumer un match de football en extrayant les zooms, en détection les ralentis et les moments saillants sans aucune hypothèse sur le mouvement des joueurs, juste en analysant le mouvement global de la vidéo. Pour calculer le résumé, les plans qui contiennent les pics d'activité détectés sont agrégés en une seule vidéo.

Puisque nous n'utilisons ni les caractéristiques du football ni les spécificités de production comme les logos, notre approche est générique. Pour confirmer cela, nous avons fixé tous les paramètres de notre méthode pour le football et entraîné le SVM sur un ensemble d'apprentissage constitué d'extraits de football. Nous avons ensuite extrait les moments saillants sur un extrait de match de handball, sans ré-entraînement, peaufinage ou adaptation. L'extrait est une partie de la finale de championnat du monde 2015 provenant d'une vidéo HDTV, le Qatar contre la France (illustration 2.10). Nous détectons sur cet extrait un moment saillant, suivi par une rediffusion ralentie, sur les trois moments qui sont présents dans cet extrait de la base de données.

Veuillez noter que le terrain de handball est mauve, et que les joueurs se déplacent différemment avec des zones de restriction différentes des règles du football.

2.5 Discussion

Dans cette partie, nous nous concentrons sur la détection de moments saillants comme les zooms, les ralentis et l'agitation globale des joueurs. Cependant, notre description apporte d'autres informations grâce aux coefficients de projection qui peuvent mener à une description sémantique du match.

Par exemple, le simple degré 0 de la base de champs polynomiaux (cf eq.2.1) peut fournir des informations intéressantes dans le processus de description sémantique : $u_{0,0}$ et $v_{0,0}$ donnent la translation sur l'axe vertical et horizontal respectivement. Ces coefficients de projection du flot optique global, comme calculé pour la détection de zoom 2.4.2, caractérise la translation de la caméra et ensuite peuvent déterminer la présence de la phase d'attaque et de contre-attaque.

La différence entre le flot initial et le flot approché polynomiale a été réduite en découpant le flot spatialement grâce à des fenêtres glissantes et en effectuant des approximations locales. Toutefois, ces projections peuvent approcher toute fonction bivariable en imposant la proximité de l'approximation en fonction du degré de fonction polynomiale bivariable. Dans l'implémentation fournie de ces méthodes et donc de la projection, nous offrons la possibilité de fixer le degré de projection à n'importe quel degré de projection polynomial, donc ces autres descriptions du mouvement peuvent être aisément calculées. Il y aura alors d'autres types de singularités présentes.

2.6 Conclusion

Nous avons présenté dans cette partie un nouveau descripteur multi-échelle robuste : les singlets. Elles correspondent au suivi des singularités de la projection du flot optique au long de la dimension temporelle de la vidéo.

Ce descripteur prouve son caractère informatif sur le mouvement en détectant des zooms (avant et arrière), des ralentis et des moments saillants durant un événement sportif sans aucun élément ad hoc (pas de logo, pas de couleur de sol particulière). Ces détections offrent alors la possibilité à construire un résumé de sport pertinent.

Ainsi en se concentrant uniquement sur les informations de mouvements et en proposant un descripteur explicite de l'évolution temporelle de la vidéo, nous avons montré l'utilité de ces informations dans l'analyse vidéo et la diversité de leurs applications par les différentes détections. Dans le prochain chapitre, nous nous intéresserons à la vidéo sous forme de cube de pixels avec l'objectif d'extraire implicitement les évolutions au cours de la vidéo et la relation entre les dimensions spatiales et la dimension temporelle.

CHAPITRE 3

Les Tenseurs

Contents

3.1 Motivations

Dans ce chapitre, nous voulons étudier les mouvements sans apporter de la connaissance humaine, sans modélisation. Pour cela, nous analysons de façon implicite les relations entre la dimension temporelle et les dimensions spatiales par l'utilisation des tenseurs. Dans cette section, nous exposons nos motivations à l'utilisation des tenseurs, l'intégration des tenseurs dans l'exploration de données et les domaines où les tenseurs ont montré leur efficacité.

3.1.1 La vidéo est un cube

Un tenseur est un tableau multi-dimensionnel à plusieurs entrées, dont *l'ordre* est le nombre de ses entrées. La vidéo est naturellement un tenseur d'ordre 3 comme l'image est une matrice, soit un tenseur d'ordre 2. À l'instar des matrices, les tenseurs, les opérations et les décompositions tensorielles sont des outils mathématiques simples et puissants qui peuvent être et sont employés à des fins très diverses.

Comme présenté dans le chapitre précédent, la vidéo est généralement étudiée comme une donnée 2D+T, c'est-à-dire comme une succession de données spatiales. Cette vision empêche d'analyser indépendamment les relations entre l'horizontalité et le temps et les relations entre la verticalité et le temps. Keimel *et al.* [48] ont montré que pour une évaluation de la qualité d'une vidéo, il est plus informatif d'analyser ces relations non-conventionnelles. La fusion des analyses de ces paires de dimensions apporte également plus d'informations. Lopez *et al.* [68] montrent que la fusion de descriptions SIFT sur chacune des paires de dimension, donc des descripteurs 2D extrais de tableaux de pixels à deux dimensions, est plus efficace que chercher des informations en trois dimensions directement par des STIP.

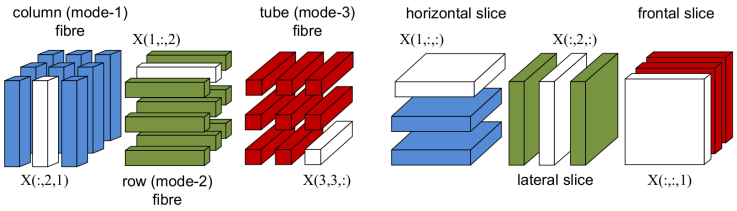


FIGURE 3.1 – Illustration des fibres et des tranches dans les tenseurs (illustration provenant de [48]).

Ainsi, ces informations entre deux dimensions quelconques permettent de décrire sur le contenu vidéo.

Les tenseurs sont analysés par dimension, par tranches (sous-tenseur d'ordre 2) et par fibres (sous-tenseur d'ordre 1), comme illustré dans la figure 6.1. Ces tranches et ces fibres sont associées à une dimension, appelées également un *mode*. La *vectorisation* d'un tenseur consiste à réassembler tous ses éléments pour en faire un vecteur. Comme un tenseur peut être vu comme un ensemble de fibres, alors la concaténation de ces fibres forment une vectorisation du tenseur en suivant l'un de ses modes.

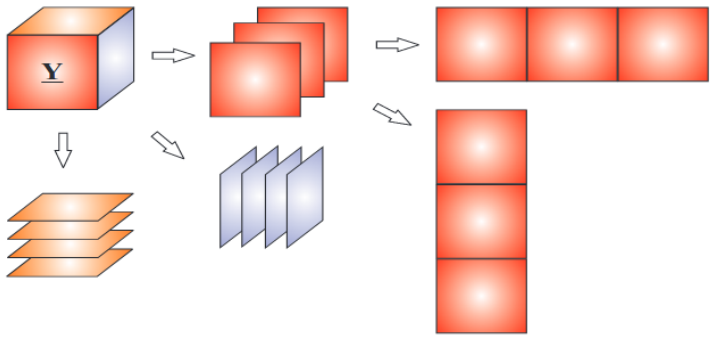


FIGURE 3.2 – Illustration des matricisations possibles d'un tenseur d'ordre 3 (illustration provenant de [15]).

De même, la *matricisation* de mode i d'un tenseur $\mathcal{X}$, appelée également son i-mode et notée $\mathcal{X}_{(i)}$, est la concaténation ces tranches selon un mode pour en faire une matrice (cf illu. 6.4). Cette matrice a la plupart du temps une dimension bien plus grande que l'autre.

L'apparition d'études montrant l'intérêt d'explorer différemment les dimensions vidéo ainsi que l'objet vidéo lui-même et l'apparition d'outils tensoriels permettant de visualiser un tableau 3D sous différentes formes ont été pour nous une motivation pour l'utilisation des tenseurs et des outils tensoriels pour l'analyse implicite d'une vidéo et ainsi des mouvements qui y sont présents.

3.1.2 La généralisation d'outils vectoriels et matriciels aux ordres supérieurs

Une matrice est un tenseur d'ordre 2 et les outils matriciels comme l'extraction en composantes principales (PCA) ou l'analyse canonique des corrélations (CCA) ont montrés leur efficacité dans le domaine de l'extraction de connaissance. Ainsi, la tendance est de généraliser ces outils aux ordres supérieurs.

Tout d'abord, la PCA étant une application de la décomposition en valeurs singulières (SVD), la SVD a été généralisée aux ordres supérieurs par De Lathauwer *et al.* par l'utilisation de la *décomposition de Tucker* : HOSVD [18]. Soit un tenseur $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ d'ordre N, sa décomposition de Tucker est définie par :

$$\begin{aligned} \mathcal{X} &= [\![\mathcal{G}; A^{(1)}, \ldots, A^{(N)}]\!] \\ &= \mathcal{G} \times_1 A^{(1)} \times_2 A^{(2)} \times_3 \cdots \times_N A^{(N)} \\ &= \sum_{i_1, i_2, \ldots, i_N} g_{i_1 i_2 \ldots i_N} . a_{i_1}^{(1)} \circ \ldots \circ a_{i_N}^{(N)} \end{aligned} \tag{3.1}$$

$A^{(i)} \in \mathbb{R}^{I_i \times R_i}$ sont appelées les matrices facteurs et $\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_N}$ le noyau décrivant les niveaux d'interactions entre les différentes composantes. L'opération $\circ$ est le produit vectoriel externe, *i.e.* pour deux vecteurs de taille quelconque x et y, $x \circ y = xy^T = A$ avec A une matrice telle que $A_{ij} = x_i y_j$.

Le *mode-n produit* d'un tenseur $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ avec une matrice $U \in \mathbb{R}^{J \times I_n}$ est un tenseur d'ordre N appartenant à $\mathbb{R}^{I_1 \times \ldots I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N}$ défini par

$$(\mathcal{X} \times_n U)_{i_1 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_N} \mathcal{X}_{i_1 \cdots i_n \cdots i_N} . U_{j i_n} \tag{3.2}$$

ce qui est équivalent, en utilisant les matricisations, à

$$\mathcal{Y} = \mathcal{X} \times_n U \Leftrightarrow \mathcal{Y}_{(n)} = U . \mathcal{X}_{(n)} \tag{3.3}$$

Au niveau des éléments de $\mathcal{X}$, cette décomposition s'exprime par :

$$\mathcal{X}_{i_1, i_2, \ldots, i_N} = \sum_{j_1, j_2, \ldots, j_N = 1}^{R_1 \times \cdots \times R_N} \mathcal{G}_{j_1 j_2 \ldots j_N} . A_{i_1 j_1}^{(1)} . \ldots . A_{i_N j_N}^{(N)} \tag{3.4}$$

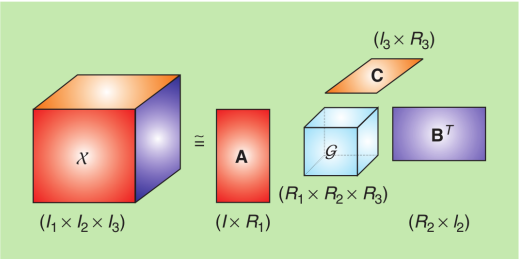


FIGURE 3.3 – Décomposition de Tucker d'une tenseur du troisième ordre. Les espaces des colonnes A,B et C représentent les sous-espaces de la données pour les trois modes. Le tenseur $\mathcal{G}$ représente les interactions complexes possibles parmi les composantes du tenseur (illustration de [14]).

La décomposition de Tucker est construite par algorithme grâce à ses versions matricielles écrites grâce aux i-modes du tenseur d'origine est exprimée comme suit :

$$\mathcal{X}_{(n)} = A^{(n)}\mathcal{G}_{(n)}(A^{(N)} \otimes \cdots \otimes A^{(n+1)} \otimes A^{(n-1)} \otimes \cdots \otimes A^{(1)})^T, \forall n \tag{3.5}$$

avec le produit de Kronecker noté par $\otimes$. *Le produit de Kronecker* entre deux matrices $A \in \mathbb{R}^{I\times J}$ et $B \in \mathbb{R}^{K\times L}$ résulte d' une matrice de $\mathbb{R}^{IK\times JL}$ définie par

$$A\otimes B = \begin{pmatrix} a_{11}B & a_{12}B\cdots a_{1J}B \\ a_{21}B & a_{22}B\cdots a_{2J}B \\ \vdots & \\ a_{I1}B & a_{I2}B\cdots a_{IJ}B \end{pmatrix} = [a_1\otimes b_1 \;\; a_1\otimes b_2 \cdots a_1\otimes b_L \;\; a_2\otimes b_1 \cdots a_J\otimes b_L] \tag{3.6}$$

Pour qu'une décomposition de Tucker soit une HOSVD, il faut que les matrices facteurs soient orthogonales et que le tenseur noyau $\mathcal{G}$ soit un tenseur tout-orthogonal et ordonné. Un tenseur est *tout orthogonal* si toutes les tranches de ce tenseur sont orthogonales par rapport au produit scalaire matriciel selon toutes les dimensions. Pour un tenseur $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times I_3}$ d'ordre 3, cela consiste à vérifier l'équation suivante :

$$\sum_{i_1,i_2} \mathcal{X}_{i_1,i_2,\alpha}.\mathcal{X}_{i_1,i_2,\beta} = \sum_{i_2,i_3} \mathcal{X}_{\alpha,i_2,i_3}.\mathcal{X}_{\beta,i_2,i_3} = \sum_{i_1,i_3} \mathcal{X}_{i_1,\alpha,i_3}.\mathcal{X}_{i_1,\beta,i_3} = 0 \;,\; \forall \alpha \neq \beta \tag{3.7}$$

Ainsi les tranches horizontales, i.e. selon le premier mode, sont mutuellement orthogonales, et dans le même temps, les tranches frontales et les tranches verticales sont également mutuellement orthogonales.

Un tenseur est dit *ordonné* si toutes ses tranches selon tous ses modes sont ordonnées par ordre décroissant selon leur norme de Frobenius.

Les dimensions de $\mathcal{G}$ sont le plus souvent imposées comme plus petites que celles de $\mathcal{X}$ pour visualiser $\mathcal{G}$ comme une version compressée de $\mathcal{X}$ car il permet de reconstruire le tenseur original à partir d'un nombre moins important d'éléments.

En résumé, la décomposition de Tucker associée à des contraintes d'orthogonalité permet d'effectuer une recherche de valeurs propres sur une donnée multi-dimensionnelle (HOSVD). Cette SVD d'ordre supérieur peut être employée pour trouver les directions principales d'un ensemble de données multi-dimensionnelles et ainsi de faire de la réduction de dimension telle une PCA, appelée alors TPCA (tensor principal component analysis) ou MPCA (multi-dimensional principal component analysis).

Par la suite, les méthodes de classification linéaire ont été également généralisées aux ordres supérieurs [55]. Ainsi Tao *et al.* proposent de généraliser la méthode de classification à large marge (SVM) : il faut classifier des tenseurs en trouvant la meilleure projection dans un espace vectoriel discriminant les données selon leur label. Ainsi, à partir d'un ensemble de N données $\mathcal{X}_i \in \mathbb{R}^{L_1,L_2,\ldots,L_M} (1 \leq i \leq N)$ et de leurs labels $y_i \in \{+1, -1\}$, on cherche la fonction de décision suivante

$$y(\mathcal{X}) = sign[\mathcal{X} \prod_{k=1}^{M} \times_k w_k + b] \tag{3.8}$$

Les vecteurs de projection w_k et le biais b sont déterminés en optimisant

$$\begin{aligned} &\min_{w_k,b,\varepsilon} J_{C-STM}(w_k, b, \varepsilon) = \frac{1}{2} \parallel \bigotimes_{k=1}^{M} w_k \parallel_F^2 + c \sum_{i=1}^{N} \varepsilon_i \\ &\text{tel que } \; y_i[\mathcal{X}_i \prod_{k=1}^{M} \times_k w_k + b] \geq 1 - \varepsilon_i, 1 \leq i \leq N \\ &\text{et } \varepsilon \geq 0 \end{aligned} \tag{3.9}$$

La méthode STM a ensuite été employée par Guo *et al.* [36], en combinaison avec une TPCA effectuée préalablement (cf illustration 3.4) pour réduire la taille de la donnée dans une tache de segmentation sur des photos satellites (maïs, bois, immeubles ...).

Le système dynamique linéaire (LDS) est aussi généralisé par Lee [60] en l'appliquant à des silhouettes et le système auto-régressif est généralisé par Zhang *et al.* [120]. Ces généralisations améliorent les outils linéaires dont elles s'inspirent mais n'apportent pas de progression importante comparée aux méthodes de classification non-linéaire qui constituent l'état de l'art actuel. Elles sont évaluées sur des bases de données de faibles variations spatiales et où les données sont de petites tailles ou alors préalablement réduites par TPCA.

Les outils tensoriels ont pourtant un impact sur l'état de l'art dans certains domaines ayant comme support le multimédia. Cependant, si les généralisations des outils linéaires aux ordres supérieurs sont intéressantes théoriquement, elles

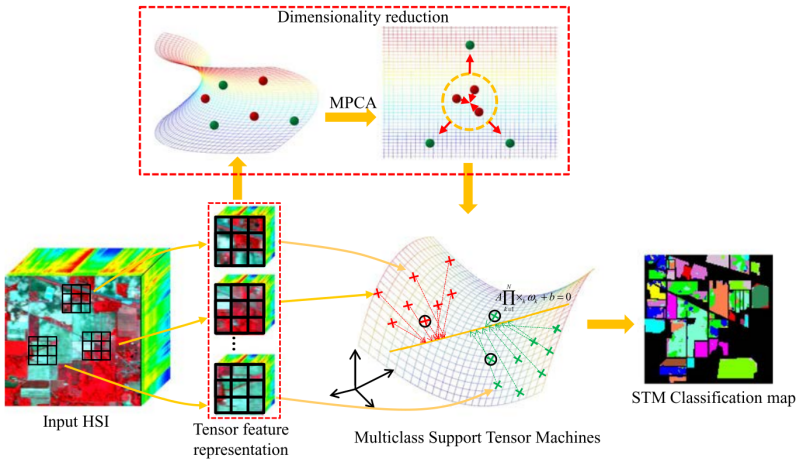


FIGURE 3.4 – Système de classification de données satellites par MPCA puis STM (illustration provenant de [36]).

sont coûteuses en temps de calcul et doivent être combinées à de la non-linéarité pour atteindre les performances actuelles de l'état de l'art.

3.1.3 Regain d'impact dans le multimédia : les réseaux de neurones profonds et la complétion

Les tenseurs et leurs décompositions sont appréciés pour l'analyse détaillée de leur structure interne et leur compression.

L'analyse de la structure interne d'un tenseur est notamment employée pour étudier et comparer différents réseaux de neurones : décrire la densité des connexions à l'intérieur d'un réseau, et décrire le pouvoir expressif d'un réseau de neurones, i.e. l'espace des fonctions modélisables [16, 70].

Cette analyse des réseaux de neurones profonds a pour but leur amélioration. Un désavantage des réseaux de neurones profonds est la capacité matérielle nécessaire. L'avènement de la capacité de calcul et de stockage de nos machines a permis l'entraînement de réseaux de neurones complexes. Pour la diffusion de ces outils de reconnaissance, il est souhaitable de pouvoir a posteriori réduire ces contraintes matérielles. Ainsi l'accélération des calculs à travers un réseau et la compression de l'espace de stockage des réseaux sont des améliorations vivement recherchées. Les tenseurs et les décompositions tensorielles sont des outils puissants pour cette compression et accélération. L'approche tensorielle de la compression de réseau est présentée par Lebedev *et al.* [59]. Dans cet article, les auteurs remplacent le tenseur stockant les poids d'une couche convolutionnelle par sa décomposition canonique polyadique (CPD).

La décomposition canonique polyadique (CP), également appelée CANDECOMP ou PARAFAC (Parallel Factor Analysis), est une variante de la décompo-

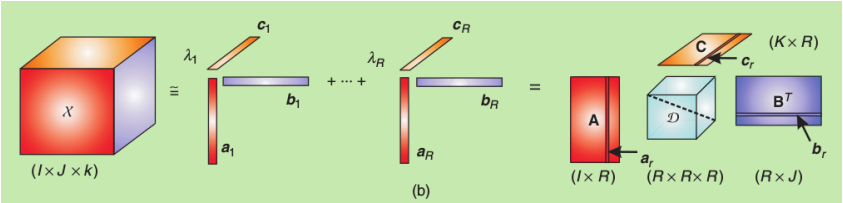


FIGURE 3.5 – Une décomposition CP d'un tenseur d'ordre 3 (illustration provenant de [14]).

sition de Tucker (cf illu. 3.5) où le tenseur noyau est super-diagonal. Soit $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, sa décomposition CP est définit par :

$$\begin{aligned} \mathcal{X} &= \sum_{r=1}^{R} \lambda_r \circ a_r^{(1)} \circ \ldots \circ a_r^{(N)} \\ &= [\![\lambda; A^{(1)}, \ldots, A^{(N)}]\!] \\ &= [\![\mathcal{D}; A^{(1)}, \ldots, A^{(N)}]\!] \end{aligned} \tag{3.10}$$

avec $\mathcal{D} = diag_N(\lambda)$ un tenseur super-diagonal d'ordre N.

La décomposition CP peut s'exprimer de façon matricielle grâce aux i-modes :

$$\begin{aligned} \mathcal{X}_{(n)} &= A^{(n)} \mathcal{D}_{(n)} (A^{(N)} \odot \cdots \odot A^{(n+1)} \odot A^{(n-1)} \odot \cdots \odot A^{(1)})^T, \forall n \\ vec(\mathcal{X}) &= [A^{(N)} \odot \cdots \odot A^{(1)}]\lambda \end{aligned} \tag{3.11}$$

Au niveau des éléments, la décomposition s'exprime par :

$$\mathcal{X}_{i_1, \cdots, i_n} = \sum_{r=1}^{R} \lambda_r . A^{(1)}_{i_1, r} . \cdots . A^{(N)}_{i_N, r} \tag{3.12}$$

Le tenseur initial est alors vu comme une somme de tenseur de rang 1 sachant qu'un tenseur de rang 1 est un tenseur égal à un produit extérieur de vecteurs. R est appelé le rang canonique du tenseur $\mathcal{X}$.

Ainsi, pour optimiser le nombre de paramètres et de calcul d'une couche convolutionnelle, Lebedev *et al.* proposent de remplacer le tenseur contenant les filtres de convolution par son expression CP. Une opération de convolution ayant comme tenseur d'entrée U de taille $X \times Y \times S$ et tenseur de sortie V de taille $(X - d + 1) \times (Y - d + 1) \times T$ s'exprime comme suit :

$$V(x, y, t) = \sum_{i=x-\delta}^{x+\delta} \sum_{i=x-\delta}^{x+\delta} \sum_{s=1}^{S} K(i - x + \delta, j - y + \delta, s, t) U(i, j, s) \tag{3.13}$$

Le tenseur de poids K d'ordre 4 est remplacé par sa décomposition CP de rang R :

$$K(i,j,s,t) = \sum_{r=1}^{R} K^x(i-x+\delta,r)K^y(j-y+\delta,r)K^s(s,r)K^t(t,r) \tag{3.14}$$

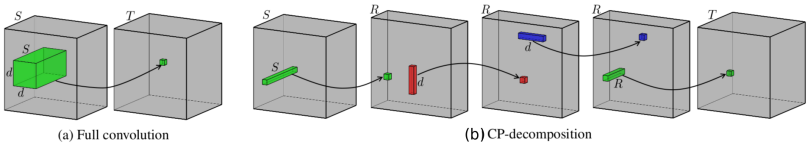


FIGURE 3.6 – Illustration d'une convolution classique dans un réseau de neurones (a) et la combinaison de 4 convolutions simples (illustration provenant de [59]).

Le tenseur de poids d'ordre 4 est représenté par un produit de 4 matrices, et ainsi la convolution multi-dimensionnelle est exprimée comme une succession de convolutions simples (cf illu. 3.6). On passe donc d'un tenseur à $d \times d \times S \times T$ éléments à $R(d+d+S+T)$ éléments dans sa décomposition ce qui réduit considérablement le stockage et le temps de calcul sans nuire aux performances.

Suite à cet article, Novikov *et al.* proposent d'utiliser la décomposition en "train", appelée décomposition TT, pour compresser les couches *"fully connected"*. Les couches fully-connected sont les couches contenant le plus de paramètres et donc celles que l'on cherche le plus à optimiser dans les réseaux de neurones.

La décomposition en train de tenseur est introduite par Oseledets *et al.* [75]. Soit $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, alors sa décomposition en train de tenseur s'écrit :

$$\begin{aligned} \mathcal{X}(i_1, \cdots, i_N) &= \sum_{r_0, r_1, \cdots, r_N} \mathcal{G}_1[i_1](r_0, r_1).\mathcal{G}_2[i_2](r_1, r_2). \cdots .\mathcal{G}_N[N](r_{N-1}, r_N) \\ &= \mathcal{G}_1[i_1]. \cdots .\mathcal{G}_N[i_N] \end{aligned} \tag{3.15}$$

avec les matrices $\mathcal{G}_j[i_j] \in \mathbb{R}^{R_{j-1} \times R_j}$ et R_j des dimensions supplémentaires choisies telles que $R_0 = R_N = 1$. On remarque ici que l'on a la formule uniquement par élément : chaque élément de $\mathcal{X}$ est un produit de matrices et de vecteurs. Cependant, comme tous les éléments de la ligne i_1 de $\mathcal{X}$ sont calculés à partir de la matrice $G_1[i_1]$, alors il y a bien une factorisation de $\mathcal{X}$ qui est faite. L'illustration 6.6 montre la factorisation par train de tenseur pour un tenseur d'ordre 5.

La décomposition TT est appréciée, car elle s'applique bien aux tenseurs aux dimensions déséquilibrées. En effet, pour calculer une décomposition de Tucker, on doit effectuer des décompositions en valeurs singulières sur les matricisations du tenseur (cf eq. 3.5). Alors que, pour calculer une décomposition TT, on effectue des SVD sur les mode-(1,...,k) matricisations du tenseur qui forment des matrices bien plus équilibrées. Le mode-(1,...,k) matricisation d'un tenseur d'ordre N est une

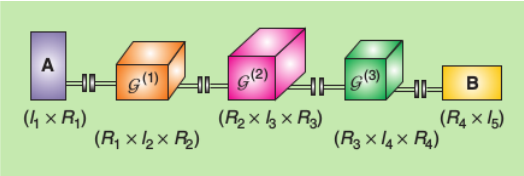


FIGURE 3.7 – La TTD d'un tenseur d'ordre 5 $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_5}$ composée de deux matrices wagons et de trois tenseurs wagons d'ordre 3. Les cinq wagons sont connectés par contractions tensorielles et l'on a $\mathcal{X}(i_1, i_2, i_3, i_4, i_5) = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_5=1}^{R_5} a_{i_1,r_1} . \mathcal{G}^{(1)}(r_1, i_2, r_2) . \mathcal{G}^{(2)}(r_2, i_3, r_3) . \mathcal{G}^{(3)}(r_3, i_4, r_5) . b(r_4, i_5)$ (illustration provenant de [14]).

matrice où l'on a regroupé les k premières dimensions pour constituer les lignes et les (N-k) autres pour les colonnes.

Une couche de neurones fully-connected s'exprime par $y = Wx + b$ avec $x \in \mathbb{R}^N$, $W \in \mathbb{R}^{M \times N}$, y et $b \in \mathbb{R}^M$. Pour la compression et l'accélération, Novikov *et al.* proposent tout d'abord de réorganiser tous les éléments vectoriels ou matriciels de cette équation en tenseur d'ordre d. Cette tensorisation se fait par une factorisation des indices, $M = \sum_{i=1}^{d} m_i$ et $N = \sum_{i=1}^{d} n_i$, et surtout deux fonctions de mise en correspondance μ et ν qui sont des bijections entre l'indice l et le vecteur d'indices $\mu(l)$ dans ce nouveau découpage. Ainsi, on peut créer $\mathcal{X}(\mu(l)) = x_l$ à partir d'un mapping μ entre les indices initiaux et les indices du nouveau découpage, et de même pour les tenseurs $\mathcal{Y}$ et $\mathcal{B}$. Pour $\mathcal{W}$, comme W est une matrice, il utilisera les sorties des deux mises en correspondance μ et ν comme indices. Donc $W(t, l) = \mathcal{W}(\mu(l), \nu(l))$.

Enfin, en exprimant $\mathcal{W}$ par sa décomposition TT, on obtient alors le calcul suivant :

$$\mathcal{Y}(i_1, \cdots, i_d) = \sum_{j_1, \cdots, j_d} G_1[i_1, j_1] \cdots G_d[i_d, j_d] \mathcal{X}(j_1, \cdots, j_d) + \mathcal{B}(i_1, \cdots, i_d). \quad (3.16)$$

Cette couche de réseau de neurones est alors intitulée une couche TT (TT-layer). Cette factorisation leur permet d'optimiser le réseau "Very Deep VGG" pour qu'il soit 7 fois plus rapide sans perte sur le taux de reconnaissance.

Ensuite, Garipov *et al.* [31] utilisent la décomposition TT sur les couches de convolution. Comme cette décomposition directe ne compresse pas assez le tenseur de convolution, ils le vectorisent pour la tensoriser avec un nouvel ordre et de nouvelles dimensions, et ils obtiennent alors une compression de la couche plus importante. Enfin, Yang *et al.*[116] proposent l'intégration de plusieurs couches TT pour remplacer toutes les couches fully-connected de réseaux de neurones récurrents, type GRU ou LSTM, et ils montrent que la diminution importante des

paramètres permet de mettre en entrée du RNN les images originales des vidéos. En effet, les RNN sont difficiles à entraîner comme évoqué dans la section 1.2.1, et les auteurs de ces méthodes choisissent souvent une description spatiale des frames pour entraîner le réseau. L'intégration de ces couches TT permet d'accélérer les calculs des réseaux RNN, de diminuer le temps d'apprentissage (moins de paramètres) mais il permet aussi des entrées plus importantes en taille, ce qui permettrait de développer l'utilisation des réseaux récurrents. Il s'avère également que, dans certains cas, l'introduction des décompositions permettent d'améliorer légèrement les résultats sur la globalité des exemples comme le ferait une régularisation.

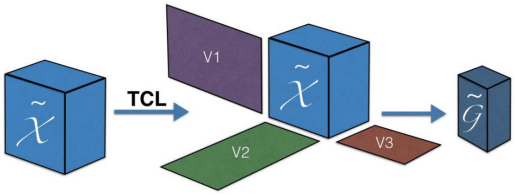


FIGURE 3.8 – Illustration d'une couche TCL sur un tenseur d'entrée d'ordre 3 (illustration provenant de [54]).

Kossaifi *et al.* [54] présentent une utilisation différente des décompositions tensorielles dans les réseaux de neurones. Ils présentent la *Tensor Contraction Layer* (TCL) qui est une couche s'inspirant de la décomposition de Tucker et de sa non-unicité. L'entrée de la couche est un tenseur et la sortie est le tenseur noyau de la décomposition (cf illu. 3.8). Ce sont les matrices facteurs qui sont donc optimisées lors de l'apprentissage pour obtenir les meilleurs tenseurs noyaux possibles pour la classification. Cette approche est assez différente des autres intégrations de décomposition tensorielle dans des réseaux de neurones profonds, car on n'utilise pas la décomposition pour réduire la description (le nombre d'éléments) d'un tenseur en approchant au mieux le tenseur initial mais on l'utilise comme une projection qui sélectionne des motifs par un apprentissage. On peut ramener cette différence à la différence entre une PCA qui cherche à réduire un ensemble de vecteurs et une projection qui est effectuée dans un but de classification ; ainsi la différence entre une description représentative et une description discriminative.

Ben-Younes *et al.* [5] utilisent également une décomposition de Tucker dans un réseau de neurones dans une application de réponse à une question à partir d'une image (Visual Question Answering). Dans ce domaine, l'intérêt se porte usuellement sur la fusion de la question et de l'image comme entrées d'un système de réponse. Ben-Younes *et al.* proposent d'utiliser le produit externe entre ces deux représentations créant ainsi un tenseur contenant les interactions entre ces deux représentations. Ce tenseur est alors décomposé grâce à la décomposition de Tucker, et le tenseur noyau résultant sera l'entrée de nouvelles couches de neurones

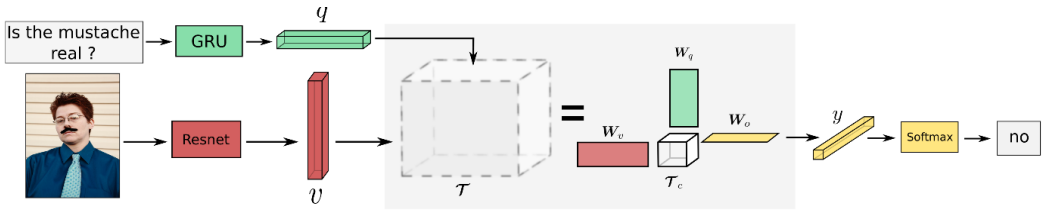


FIGURE 3.9 – Réseau de neurones MUTAN pour une application de réponse à une question à partir d'une image de [5]).

permettant de prédire une réponse. Contrairement à la TCL, les matrices facteurs ne sont pas apprises, mais, pour obtenir un tenseur noyau discriminant, les auteurs ont montré qu'il est efficace de contraindre le tenseur noyau à être creux. Il est à noté que le tenseur noyau doit être creux mais pas forcément super-diagonal comme lors d'une décomposition CP.

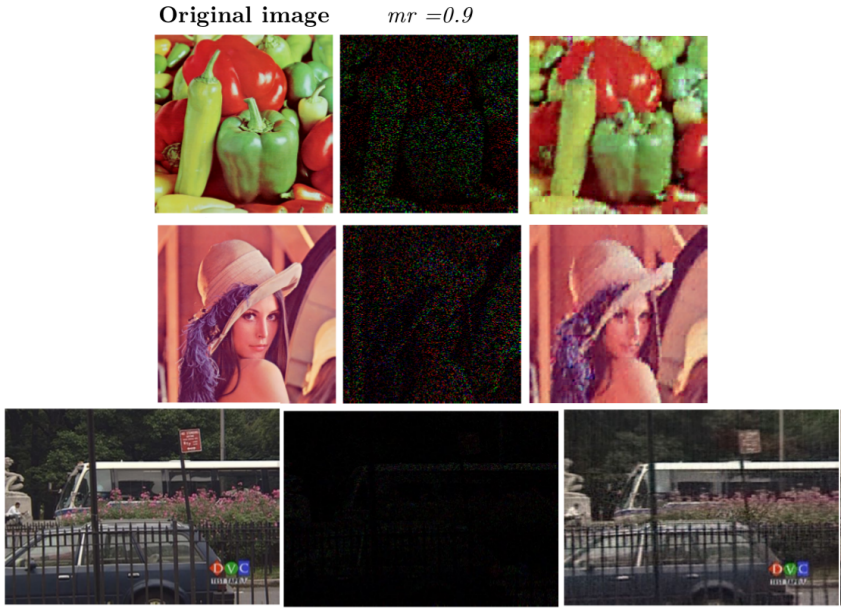


FIGURE 3.10 – Exemples de complétion d'images auxquelles il reste 10% des pixels avec la méthode TMAC-TT de Bengua *et al.*(illustrations construites à partir de [6])

Pour finir, un nouveau domaine multimédia où les tenseurs ont montré leur efficacité est la complétion de données, en particulier pour les images et les vidéos. L'objectif est de reconstruire un tenseur avec des données manquantes. Pour reconstruire des matrices, l'approche usuelle est de chercher la matrice contenant les valeurs connues et ayant la plus petite norme nucléaire. Pour l'appliquer à des données tensorielles, Liu *et al.* [64] généralisent la définition de la norme nucléaire

aux tenseurs comme étant la somme pondérée des normes de ses matricisations. Puis ils proposent 3 algorithmes de base à partir de 3 décompositions différentes. Tandis que Bengua *et al.* [6] utilisent une décomposition TT pour construire leurs deux algorithmes d'optimisation pour de la complétion de vidéo. Leur méthode permet de reconstruire une image ou une vidéo à partir de 10% des pixels (figure 3.10). Dans ce dernier article, il est intéressant de noter que les auteurs ont utilisé l'augmentation de tenseur à travers des changements d'indices comme Garipov *et al.*. Réarranger un tenseur pour augmenter son ordre sans modifier son nombre d'éléments offre certains avantages, notamment obtenir des matrices équilibrées en dimension ou accélérer les décompositions TT.

3.2 Les tenseurs pour la représentation et la classification

Dans la section précédente, nous avons exposé nos motivations pour l'utilisation des tenseurs. Nous avons vu que les tenseurs sont souvent couplés à des méthodes déjà performantes, comme les réseaux de neurones, dans le but de les améliorer. L'objectif de ce chapitre étant l'extraction implicite de contenu informatif sur une classe d'action, nous allons maintenant nous intéresser aux systèmes d'exploration de données qui sont exclusivement basés sur les tenseurs et les décompositions tensorielles. En fait, nous nous intéressons à la création et aux utilisations des décompositions ayant pour but d'extraire une représentation d'un ensemble de données.

3.2.1 TPCA et ses utilisations

La TPCA a été définie dans la sous-section 3.1.2 comme une application de la décomposition de Tucker (eq 3.1) contrainte pour être considérée comme une HOSVD. Dans la suite, nous allons présenter comment la TPCA a été employée dans un but de représentation et de classification sur du contenu image et vidéo.

Vasilescu *et al.* furent les premiers à populariser l'application de la TPCA sur du contenu multimédia grâce à leur analyse des matrices facteurs sur une base de données de visages : TensorFaces[102]. Dans cet article, les auteurs construisent un tenseur $\mathcal{D}$ stockant la base d'images de sorte qu'une dimension représente la donnée (les pixels) et les autres leurs attributs (le sujet, la luminosité, l'expression faciale et la position du sujet). Ce tenseur $\mathcal{D}$ est alors de taille $28 \times 5 \times 3 \times 3 \times 7945$, et donc un tenseur d'ordre 5. La décomposition TPCA est alors de la forme :

$$\mathcal{D} = \mathcal{Z} \times_1 U_{sujet} \times_2 U_{point\ de\ vue} \times_3 U_{illum} \times_4 U_{express} \times_5 U_{pixels} \tag{3.17}$$

Les auteurs ont choisi un tenseur noyau $\mathcal{Z}$ de même taille que $\mathcal{D}$. Par conséquent, les matrices facteurs sont toutes carrées. Ils ont ensuite analysé les matrices facteurs extraites et le tenseur noyau. Tout d'abord, dans cette décomposition, la

matrice facteur U_{pixels} représente la spatialité des données ; de cette façon, les colonnes de U_{pixels} contiennent les vecteurs propres si l'on avait effectué une PCA globale sur toutes les images de visage (3.11).

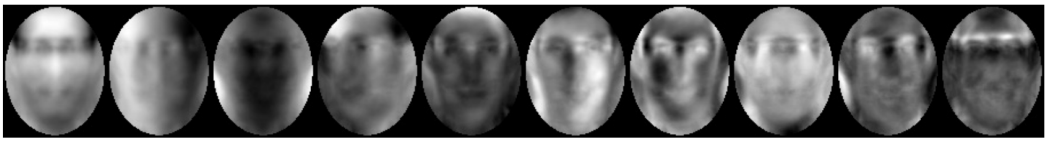


FIGURE 3.11 – Les 10 premières colonnes de U_{pixels} (illustration de [102]).

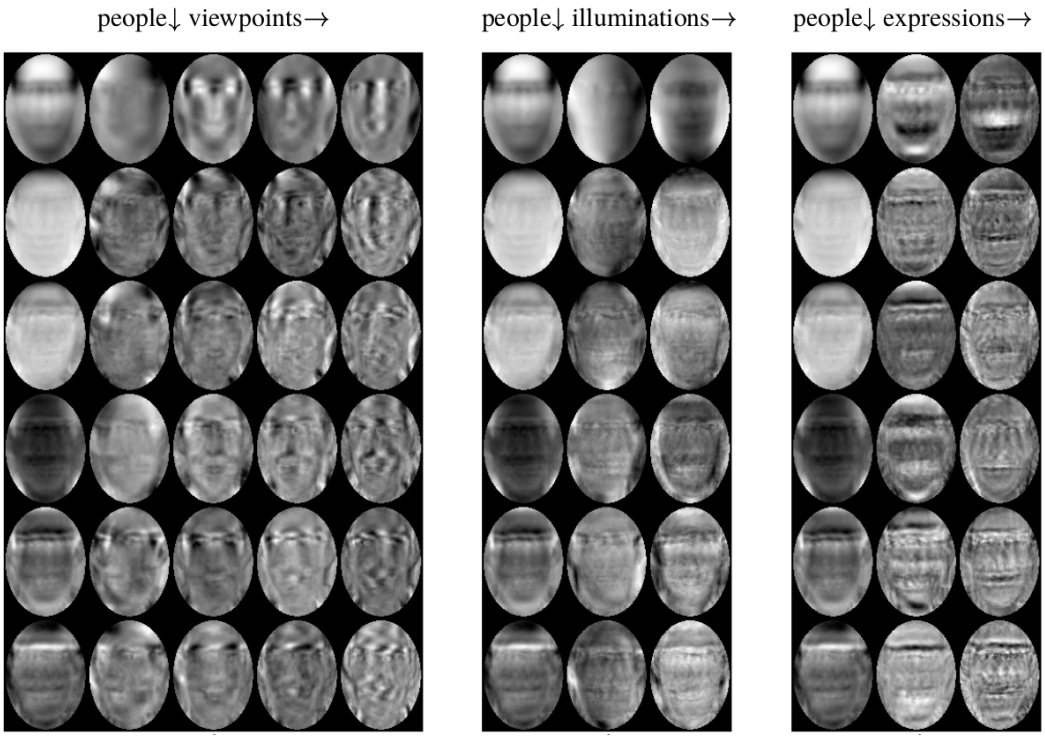


FIGURE 3.12 – Les premiers vecteurs propres contenus dans $Z \times_5 U_{pixels}$ selon différentes directions (illustration de [102]).

Ensuite, $Z \times_5 U_{pixels}$ est présenté comme le tenseur permettant d'analyser les vecteurs propres selon les modes en ajoutant la distribution spatiale grâce à U_{pixels}. Dans l'illustration 3.12, on peut visualiser ces différents vecteurs propres, selon les différents modes du tenseur.

Enfin, ils ont montré qu'en multipliant $Z \times_5 U_{pixels}$ avec le vecteur propre du point de vue 1, i.e. la première colonne de $U_{point\ de\ vue}$, on obtient alors tous les précédents vecteurs propres transformés pour être spécifiques au point de vue 1 (3.13). De même, on crée des vecteurs propres spécifiques au sujet 1 en multipliant

$Z \times_5 U_{pixels}$ avec la première colonne de U_{sujet}.

Enfin, dans la figure 3.14, vous pouvez visualiser les vecteurs propres en ayant fixé le point de vue, l'illumination et l'expression faciale, en somme tous les attributs excepté celui indiquant la personne sur la donnée.

Cette analyse nous montre que le noyau contient une représentation neutre des différents attributs, qui sont réintégrés par le produit tensoriel entre ce noyau et la colonne de la matrice facteur associée à cet attribut. Par contre, en analysant particulièrement l'illustration 3.13, on remarque que les matrices facteurs, et les vecteurs propres en général, sont dépendants des positions spatiales. Ainsi, même en fixant le sujet, on ne discerne pas forcément de motif représentatif du sujet en question, mais plutôt un peu de représentation du sujet dans toutes les positions visualisées. Ci-dessous, vous avez la représentation des vecteurs propres pour d'autres sujets (illustration 3.15).

Suite à cette analyse visuelle de la décomposition TPCA, Vasilescu *et al.* [102] proposent une méthode de classification basée sur l'interprétation des matrices facteurs comme les coefficients de décomposition de la TPCA. Ainsi, les auteurs considèrent le tenseur $\mathcal{B} = \mathcal{Z} \times_2 U_{point\ de\ vue} \times_3 U_{illum} \times_4 U_{express} \times_5 U_{pixels}$ comme le tenseur contenant les 45 vecteurs propres formant la base de chaque combinaison de conditions (luminosité, point de vue et expression). Ainsi, le visage de sujet p, vu avec la luminosité i, du point de vue v et avec l'expression e peut s'exprimer par : $I_{p,v,i,e} = \mathcal{B}_{v,i,e}.c_p$ où c_p est la colonne p de U_{sujet} et $\mathcal{B}_{v,i,e}$ est la partie du tenseur $\mathcal{B}$ correspondant aux caractéristiques v, i, e. $\mathcal{B}_{v,i,e}$ est donc de taille $28 \times 1 \times 1 \times 7945$: c'est une matrice. Ainsi, pour un nouvel échantillon I, on calcule toutes ses projections $c_{v,i,e}$ selon tous les arrangements des autres attributs et on choisit l'identité de la personne en cherchant le plus proche voisin parmi les c_p existants.

Cette première utilisation de la TPCA pour classifier soulève plusieurs questions. Tout d'abord au niveau des attributs, cette configuration de la donnée exige un et un seul échantillon pour chaque combinaison de conditions. On ne peut alors pas exploiter deux images de la même personne dans les mêmes conditions et donc on ne peut pas étudier la variance de cette sous-classe. À l'inverse, s'il manque une image dans certaines conditions, on a vu précédemment à travers la complétion que les décompositions tensorielles sont robustes quant aux données manquantes. De plus, dans les cas classiques de classification, la seule donnée disponible est le label et l'on a aucune donnée sur d'autres attributs. Ainsi, on aurait alors qu'une seule matrice B, et l'on en viendrait à un calcul de PCA classique. De plus, les auteurs vectorisent la donnée image en une seule dimension. Dans l'annexe de la thèse de Vasilescu [101], elle avance plusieurs arguments pour ce choix ; notamment que la spatialité des données est retrouvée grâce au produit avec la matrice facteur de position U_{pixel}.

Par la suite, de nombreux articles utilisent cette décomposition comme une projection dans un espace de représentations robustes, mais décident de classifier la donnée autrement : la TPCA est alors utilisée comme une projection de la donnée multi-dimensionnelle et c'est le noyau qui sera la représentation de la donnée. Cette utilisation de la PCA rejoint l'utilisation de la décomposition de Tucker

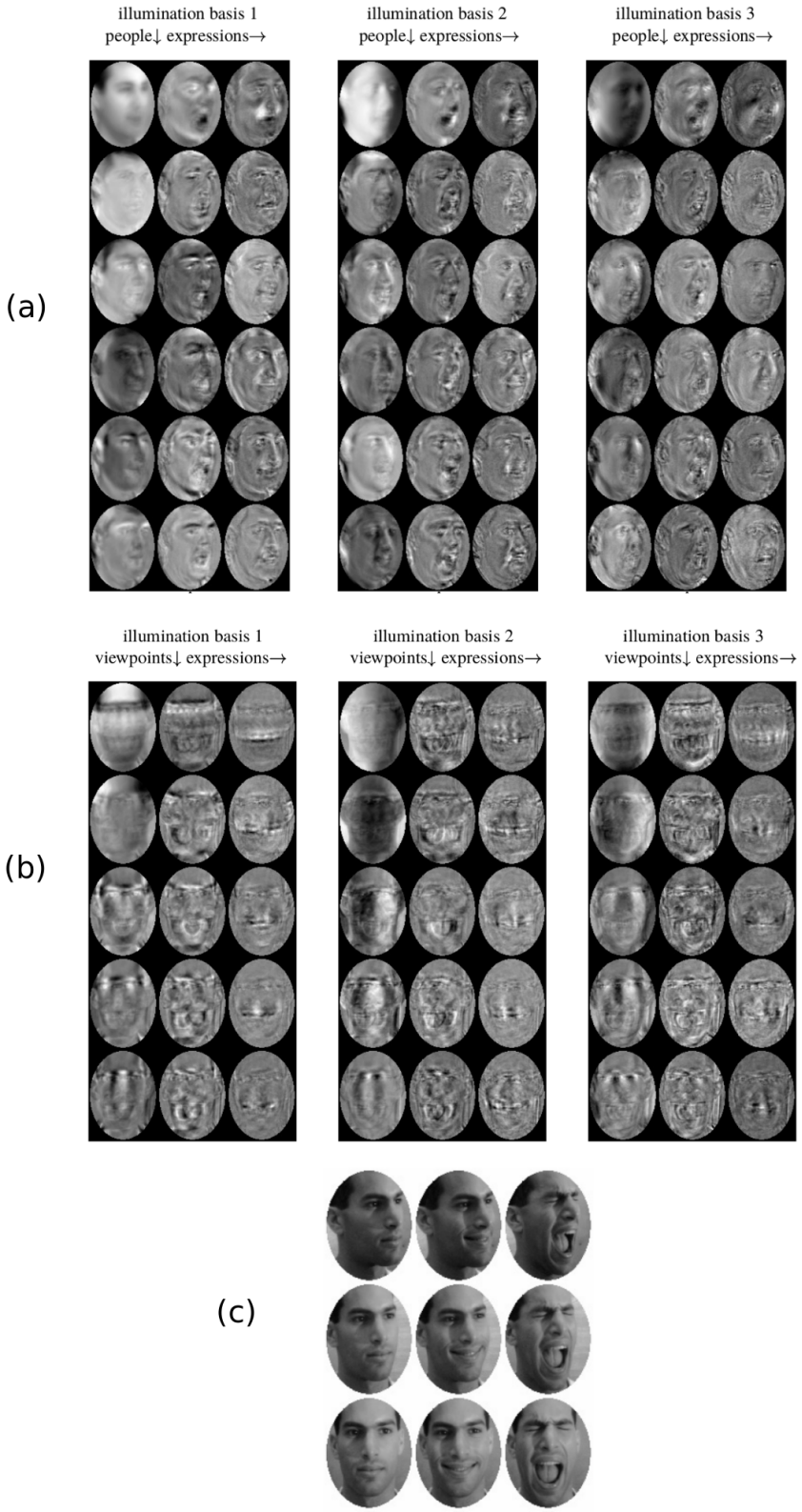


FIGURE 3.13 – (a)Les premiers vecteurs propres de la base de données selon le point de vue 1 (b) Les premiers vecteurs propres de la base de donnée selon le sujet 1 (c) Exemples d'image provenant du sujet 1

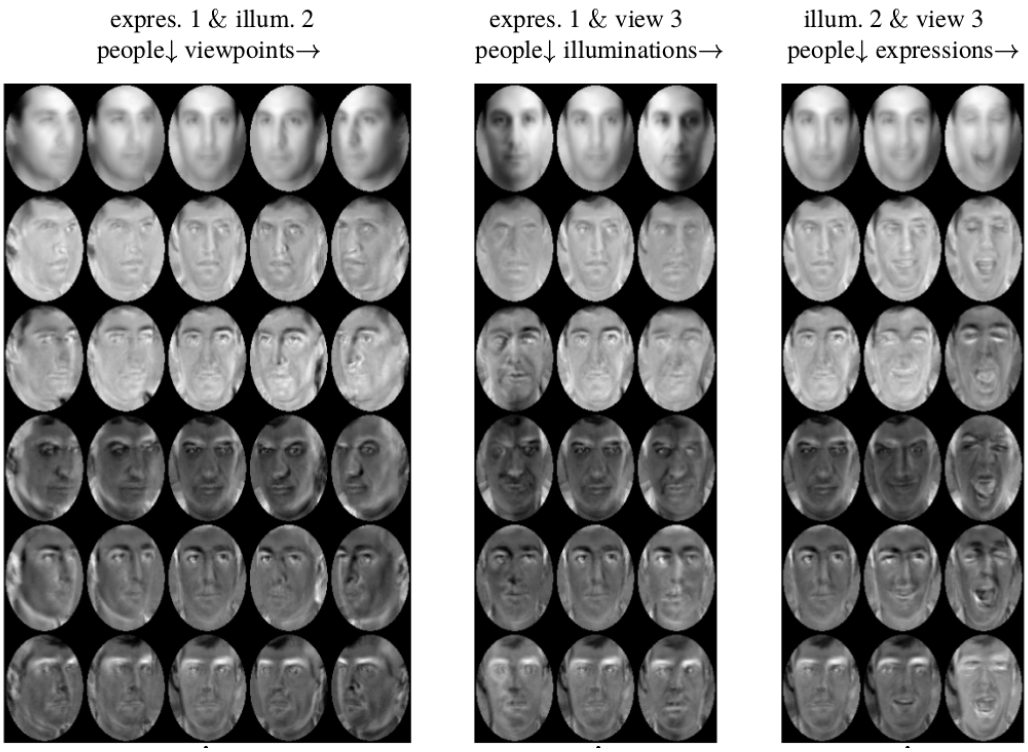


FIGURE 3.14 – Les premiers vecteurs propres en fixant tous les attributs sauf l'identifiant du sujet (illustration de [102]).

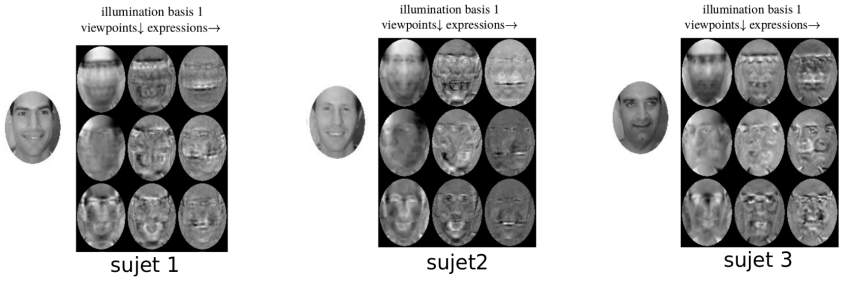


FIGURE 3.15 – Les premiers vecteurs propres selon le sujet 1, 2 et 3 en suivant les dimensions des points de vue et des expressions faciales (illustration de [102]).

comme la couche de neurones TCL.

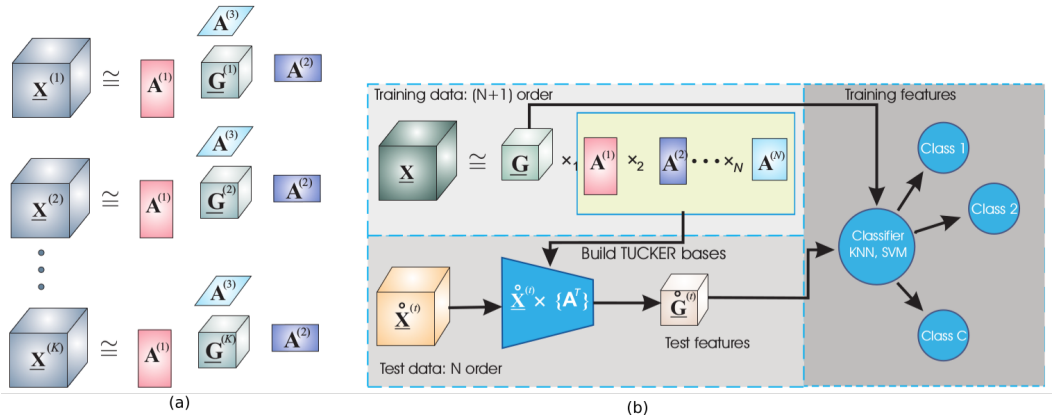


FIGURE 3.16 – (a) Illustration d'une extraction de représentation par TPCA. (b) Le système classique de classification par TPCA sur le tenseur contenant l'ensemble des données (illustration provenant de [78]).

La première approche est alors de contraindre la décomposition pour que les tenseurs noyaux soient discriminants selon leur classe. Cichocki *et al.* [78] présentent cette approche comme l'analyse discriminante d'ordre supérieure (HODA) en maximisant le ratio de Fisher :

$$\phi = \frac{\sum_{c=1}^{C} K_c \|\mathcal{G}_c - \mathcal{G}\|_F^2}{\sum_{k=1}^{K} \|\mathcal{G}^{(k)} - \mathcal{G}_{c_k}\|_F^2} \tag{3.18}$$

où $\mathcal{G}_c$ est la moyenne des tenseurs noyaux de la classe c contenant K_c éléments, c_k indique la classe de l'élément k et $\mathcal{G}$ est la moyenne de tous les tenseurs noyaux.

Jia *et al.* [42] utilisent une TPCA semblable pour reconnaître des actions à partir de silhouettes extraites de vidéos. Ils contraignent les tenseurs noyaux à être discriminants selon le label grâce à un critère de dispersion sur chaque classe proche du ratio de Fisher, mais exprimé sous forme de différences. Ce critère discriminant est ajouté à la fonction d'optimisation de la décomposition de Tucker. Puis les données de test sont représentées par leur tenseur noyau calculé à partir des matrices facteurs fixées de la TPCA et ces représentations sont classifiées par la méthode du plus proche voisin.

Su *et al.* [91] appliquent la TPCA pour classifier des vidéos de la base de données de vidéos de mains de Cambridge. Il utilise une TPCA pour réduire les dimensions spatiales, mais pas la dimension temporelle. Ils modifient la dimension temporelle pour réduire toutes les vidéos à la même longueur en supprimant les frames successives les plus semblables. Puis ces représentations sont classifiées par une recherche de plus proche voisin grâce à une distance géodésique sur les

espaces engendrés par les tenseurs noyau de chaque classe.

La méthode TPCA est toujours employée sur des bases contenant peu de données, elles-mêmes étant de petites tailles et très corrélées spatialement : des visages, des mains ... En effet, la décomposition de Tucker demande d'effectuer des SVD sur les matricisations de ce tenseur contenant toute la base. Il est donc préférable d'avoir un tenseur initial de taille raisonnable.

3.2.2 Régression d'ordre supérieur par les moindres carrés partiels HO-PLS

La prochaine décomposition est inspirée de la régression linéaire généralisée aux ordres supérieurs : *régression d'ordre supérieur par les moindres carrés partiels* (ou High order partial least square HOPLS). Elle a été introduite par Zhao *et al* [121] comme une méthode généralisée de régression multi-linéaire. L'objectif est de mettre en relation la décomposition de deux tenseurs ayant une dimension commune ; les tenseurs sont alors exprimés dans un sous-espace optimisant leur propre approximation en maximisant la covariance des éléments de décomposition. Cette technique est directement inspirée de la méthode d'approximation partielle des moindres carrées pour les relations entre matrices (illu. 3.17).

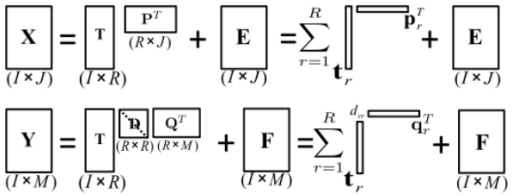


FIGURE 3.17 – Le modèle PLS : décomposition de données comme la somme de matrices de rang 1 (illustration provenant de [121]).

Soit $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ et $\mathcal{Y} \in \mathbb{R}^{J_1 \times \cdots \times J_M}$ deux tenseurs d'ordres différents ayant le même premier mode, donc $I_1 = J_1$. Alors $\mathcal{X}$ et $\mathcal{Y}$ sont décomposés tels que

$$\begin{aligned} \mathcal{X} &= \sum_{r=1}^{R} \mathcal{G}_r \times_1 t_r \times_2 P_r^{(1)} \times_3 \cdots \times_N P_r^{(N-1)} + \mathcal{E}_R \\ \mathcal{Y} &= \sum_{r=1}^{R} \mathcal{D}_r \times_1 t_r \times_2 Q_r^{(1)} \times_3 \cdots \times_M Q_r^{(M-1)} + \mathcal{F}_R \end{aligned} \tag{3.19}$$

avec R étant le nombre de vecteurs latents, $t_r \in \mathbb{R}_1^I$ le r^{ieme} vecteur latent, $P_r^{(n)} \in \mathbb{R}^{I_{n+1} \times L_{n+1}}$ et $Q_r^{(m)} \in \mathbb{R}^{J_{m+1} \times K_{m+1}}$ sont les matrices facteurs de mode n et m respectivement, et $\mathcal{G}_r \in \mathbb{R}^{1 \times L_1 \times \cdots \times L_N}$ et $\mathcal{D}_r \in \mathbb{R}^{1 \times K_1 \times \cdots \times K_M}$ sont les tenseurs noyau.

Dans le cas où l'on a deux représentations d'une même donnée, il est intéressant de les organiser sous forme de deux tenseurs ayant donc une dimension

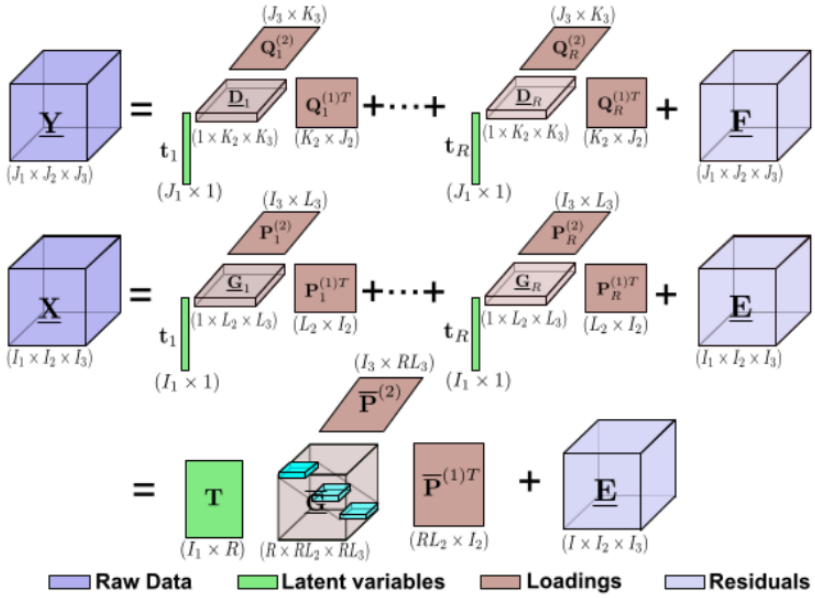


FIGURE 3.18 – Diagramme schématique de la méthode HOPLS : l'approximation d'un tenseur $\mathcal{X}$ par une somme de tenseurs de rang $(1, L_2, L_3)$ et l'approximation de $\mathcal{Y}$ suivant le même principe avec les composantes latentes communes T (illustration provenant de [121]).

en commun, le numéro de l'échantillon, et alors chercher à exprimer les relations entre les deux tenseurs et possiblement prédire une des représentations à partir de l'autre. Par exemple, si l'on voulait prédire des mouvements corporels à partir de données d'électroencéphalographies (ECoG), on commencerait par étudier les relations entre les données enregistrées par capteurs ECoG et par capteurs spatiaux 3D à partir d'enregistrements sur un ensemble de sujets.

Inspirée de la régression, elle est formulée initialement pour réunir deux représentations d'un même élément. Ainsi, dans une vision multi-modale de l'objet, deux descriptions partageront des éléments de leur décomposition. En classification, on cherche à réunir une représentation d'un élément et sa classe $\{-1, +1\}$. Beaucoup de problèmes de décomposition classiques peuvent être reformulés comme des problèmes de régression et généralisés grâce à cette décomposition : CCA, LDA... [10]

3.2.3 Extraction de représentation communes et individuelles CIFE

Inspirée par le travail de Lock *et al.* dans la séparation de sources aveugles [67], Zhou *et al.* ont présenté une version tensorielle de l'extraction de connaissance d'un ensemble de données. Cette version fut tout d'abord appliquée sur des données biomédicales (radios de poumons, IRM) [125] puis elle fut présentée sur des photos (visages, objets) [124]. Cette décomposition porte plusieurs noms : l'analyse

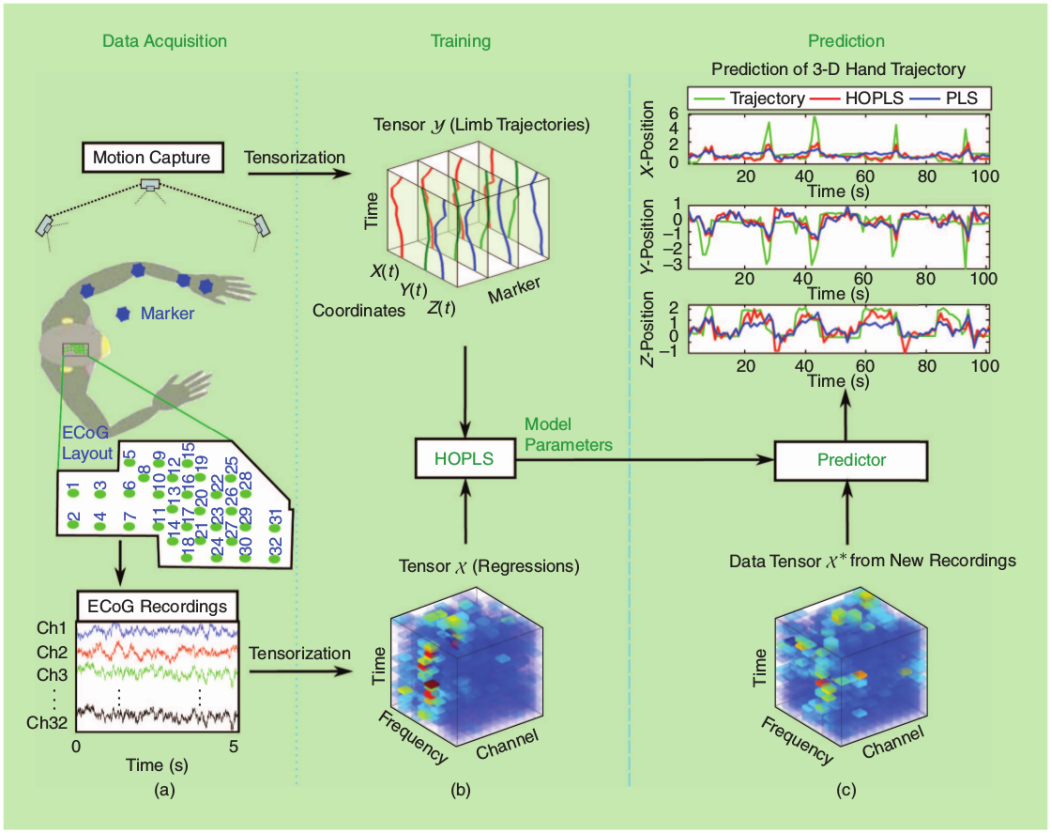


FIGURE 3.19 – Prédiction de mouvement de bras à partir de signaux électriques du cerveau (a) L'installation de l'expérience (b) Construction des tenseurs de données et entraînement du modèle HOPLS (c) Un nouveau tenseur de signaux et la prédiction de nouvelles trajectoires de bras en comparant HOPLS au PLS vectoriel (illustration provenant de [14]).

en composantes multi-ordres liées (LMWCA ou LWCA), l'extraction d'une orthogonale commune (COBE) qui est en fait une partie de la décomposition et enfin l'extraction ou l'analyse de représentations communes et individuelles (CIFE ou CIFA).

Cette méthode a été créée dans le but d'exploiter la nature liée de groupes de données et de leurs dimensions. Soit un ensemble de tenseurs $\mathcal{Y} = \{\mathcal{Y}_1, \mathcal{Y}_2, \cdots, \mathcal{Y}_N\}$ qui partagent un mode. Comme dans l'article [125], pour simplifier la notation, on suppose que le mode partagé est le premier et que les tenseurs $\mathcal{Y}_n$ sont d'ordre 3. On suppose donc $\mathcal{Y}_n \in \mathbb{R}^{D \times J_{n,2} \times J_{n,3}}$. Par la CIFE, on cherche une décomposition de chaque tenseur comme suit :

$$\mathcal{Y}_n = \mathcal{G}^{(n)} \times_1 A^{(1,n)} \times_2 A^{(2,n)} \times_3 A^{(3,n)} \tag{3.20}$$

où $A^{(1,n)} = [\overline{A}^{(1)} \breve{A}^{(1,n)}]$ se décompose donc en une partie de variables latentes

communes à tous les tenseurs $\mathcal{Y}_n$ et une partie de variables individuelles. Ainsi, on obtient :

$$\begin{aligned}\mathcal{Y}_n &= \overline{\mathcal{G}}^{(n)} \times_1 \overline{A}^{(1)} \times_2 A^{(2,n)} \times_3 A^{(3,n)} + \breve{\mathcal{G}}^{(n)} \times_1 \breve{A}^{(1,n)} \times_2 A^{(2,n)} \times_3 A^{(3,n)} \\ &= \overline{\mathcal{Y}}^{(n)} + \breve{\mathcal{Y}}^{(n)}, \quad \forall n\end{aligned} \tag{3.21}$$

En se concentrant sur la matricisation de mode 1 des tenseurs $\mathcal{Y}_n$, que l'on notera ici $\mathcal{Y}_{(n,1)}$, l'équation devient alors

$$\begin{aligned}\mathcal{Y}_{(n,1)}^T &= \overline{B}_n^{(1)} \overline{A}^{(1)T} + \breve{B}_n^{(1)} \breve{A}_n^{(1)T} \\ &\forall n = 1, \cdots, n \quad \text{avec} \\ \overline{B}_n^{(1)} &= (A^{(3,n)} \otimes A^{(2,n)}) \overline{\mathcal{G}}^{(n)} \\ \breve{B}_n^{(1)} &= (A^{(3,n)} \otimes A^{(2,n)}) \breve{\mathcal{G}}^{(n)}\end{aligned} \tag{3.22}$$

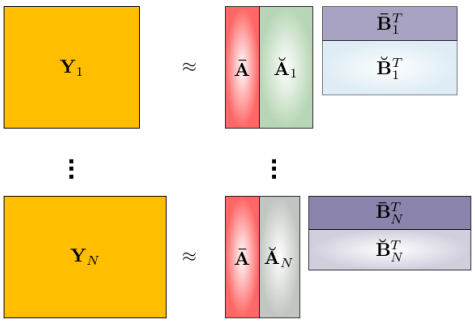


FIGURE 3.20 – Illustration de la formulation matricielle de CIFE (illustration provenant de [124]).

Ainsi, selon cette dernière formulation 3.22, on peut se ramener à une version matricielle des tenseurs initiaux (cf illu 3.20). On peut ainsi enlever tous les exposants (1) qui ne sont utiles qu'en écriture tensorielle. Tout élément $\mathcal{Y}_n$ se décompose alors en une combinaison d'éléments communs $\overline{B}_n \overline{A}^T$ et d'éléments individuel $\breve{B}_n \breve{A}_n^T$. Cette décomposition n'étant pas unique, les auteurs proposent d'ajouter des contraintes qui induisent l'unicité sans perte de généralité. On a alors la formulation finale :

$$\begin{aligned}&\min_{\overline{A}, \breve{A}_n} \sum_{n \in \mathcal{N}} \| Y_n - \overline{A}\overline{B}_n^T - \breve{A}_n \breve{B}_n^T \|_F^2, \\ \text{tels que} \quad &\overline{A}^T \overline{A} = I_C, \breve{A}_n^T \breve{A}_n = I_{R_n - C}, \\ &\overline{A}^T \breve{A}_n = 0, n \in \mathcal{N}\end{aligned} \tag{3.23}$$

avec $\mathcal{Y} = \{Y_n \in \mathbb{R}^{D \times J_n} : n \in \mathcal{N}\}$, $\mathcal{N} = \{1, 2, ..., N\}$ et C le nombre de variables latentes en communs.

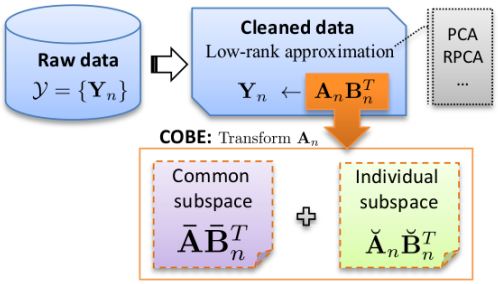


FIGURE 3.21 – Schéma d'extraction de CIFE (illustration provenant de [124]).

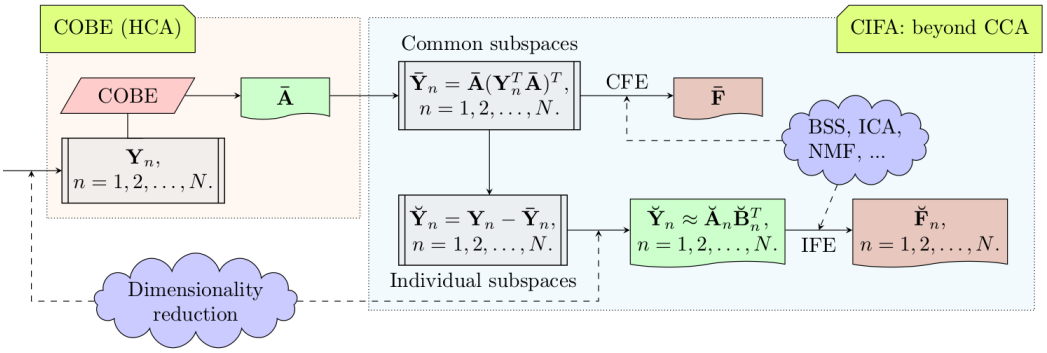


FIGURE 3.22 – Schéma d'extraction de CIFE (illustration provenant de [124]).

Comme illustré dans la figure 3.21, les auteurs proposent d'effectuer une PCA ou autre technique permettant d'enlever du bruit et de réduire les dimensions de chaque sous-groupe avant d'effectuer l'extraction des éléments communs. Cette décomposition a été testée dans un cadre de classification [124]. Dans un objectif de classification, les auteurs proposent d'ajouter au système une étape d'extraction de représentation comme BSS (blind source separation) ou NMF (nonnegative matrix factorization) sur les représentations communes $\overline{A}$ pour les normaliser. Les nouveaux représentants sont notés $\overline{F}$ (cf illustration 3.22). Les données contenues dans $\overline{F}$ représentent alors l'ensemble des données. Les auteurs proposent de classifier les nouvelles données par une recherche du plus proche voisin parmi ces représentations. La figure 3.23 illustre l'exemple de classification fourni dans l'article [124].

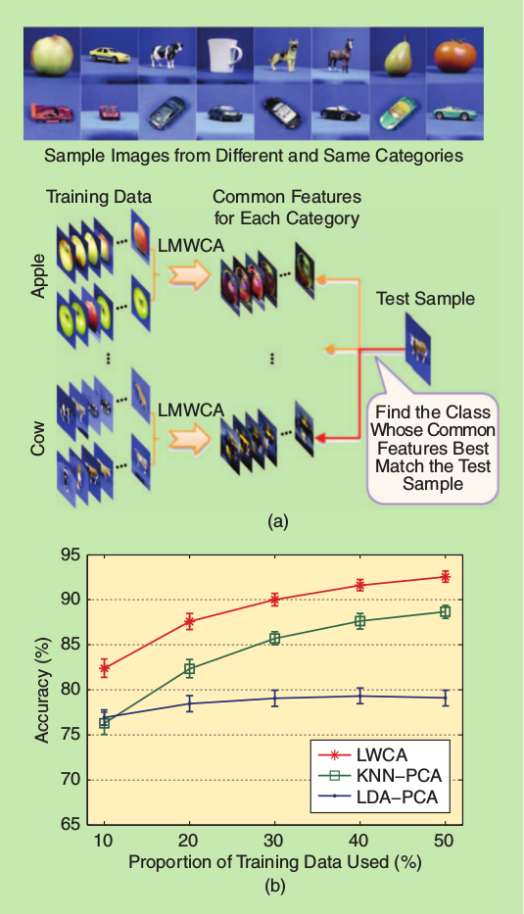
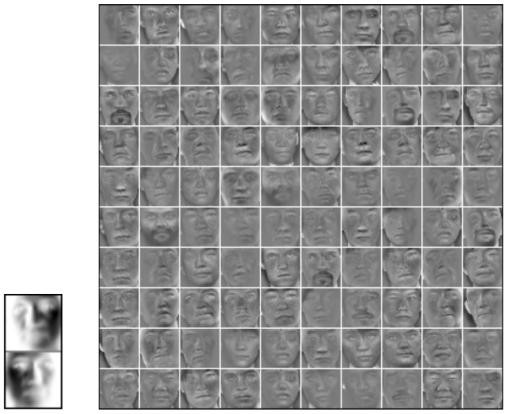


FIGURE 3.23 – Classification d'objets colorés appartenant à différentes catégories en utilisant uniquement les éléments communs calculés par CIFE (illustration provenant de [14]).

Il est à noter que cette méthode est faite pour rechercher des éléments communs à un ensemble de groupes d'éléments. En effet, à partir de l'ensemble $\mathcal{Y} = \{\mathcal{Y}_1, \mathcal{Y}_2, \cdots, \mathcal{Y}_N\}$, on obtient C éléments communs stockés dans $\overline{F}$. Ainsi, cette méthode de classification est à préférer pour des données ayant plusieurs attributs pour classer les exemples en sous-catégories (comme la méthode de classification par TPCA de Vasilescu).

De plus, on obtient au final quelques représentants de l'ensemble caractérisés comme communs à tous et non pas une représentation par élément. Ainsi, à l'arrivée d'un nouvel élément, celui-ci n'est pas projeté mais comparé directement aux représentants de chaque classe. L'illustration suivante provenant de l'article original [124] présente l'extraction de CIFE sur une base de données de visages. Dans cet exemple, on cherche le représentant commun à toute la base. Elle est ainsi découpée en deux groupes selon la direction lumineuse, et les deux premiers représentants communs à toute la base sont illustrés (cf illu. 3.24).



(a) $\bar{\mathbf{F}}$ (b) Examples of individual features $\check{\mathbf{f}}_t$

FIGURE 3.24 – Extraction d'élements communs et individuels à partir de la base Pie (a) Visages communs (b) Les visages individuels obtenant en enlevant les données communs des données bruts (illustration provenant de [124]).

3.3 Nos expériences

Dans ce chapitre, nous cherchons à extraire des descriptions de vidéos grâce aux méthodes tensorielles. Ces descriptions se veulent implicitement discriminatives et génératives puisque les méthodes tensorielles utiliseront les annotations pour construire les descriptions. Nous évaluerons si les descriptions contiennent les informations adéquates grâce à des méthodes de classification. Nous nous sommes tout d'abord intéressés à la généralisation de la régression linéaire : HOPLS. HOPLS est un choix optimal parmi l'état de l'art des méthodes tensorielles existantes permettant de décrire des données brutes en les liant à leurs annotations.

3.3.1 sur HOPLS

Dans la section précédente, nous avons présenté la décomposition HOPLS qui permet de mettre en correspondance deux représentations d'une même donnée. Dans un but de classification, nous voulons lier chaque donnée avec son vecteur de labels.

Pour rappel, nous cherchons à lier les décompositions de deux tenseurs $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ et $\mathcal{Y} \in \mathbb{R}^{J_1 \times \cdots \times J_M}$, ayant une dimension commune $I_1 = J_1$, à travers le partage de vecteurs latents communs t_r.

$$\begin{aligned} \mathcal{X} &= \sum_{r=1}^{R} \mathcal{G}_r \times_1 t_r \times_2 P_r^{(1)} \times_3 \cdots \times_N P_r^{(N-1)} + \mathcal{E}_R \\ \mathcal{Y} &= \sum_{r=1}^{R} \mathcal{D}_r \times_1 t_r \times_2 Q_r^{(1)} \times_3 \cdots \times_N Q_r^{(N-1)} + \mathcal{E}_R \end{aligned} \tag{3.24}$$

avec R étant le nombre de vecteurs latents, $t_r \in \mathbb{R}_1^I$ le r^{ieme} vecteur latent, $P_r^{(n)} \in \mathbb{R}^{I_{n+1} \times L_{n+1}}$ et $Q_r^{(m)} \in \mathbb{R}^{J_{m+1} \times K_{m+1}}$ sont les matrices facteurs de mode n et m respectivement, et $\mathcal{G}_r \in \mathbb{R}^{1 \times L_1 \times \cdots \times L_N}$ et $\mathcal{D}_r \in \mathbb{R}^{1 \times K_1 \times \cdots \times K_M}$ sont les tenseurs noyaux.

Dans notre cadre de classification vidéo, $\mathcal{X} \in \mathbb{R}^{N \times D_{size}}$ représente la base de N vidéos de taille D_{size} et $\mathcal{Y}$ est une matrice de taille $N \times C$ contenant les labels.

Les hyper-paramètres à choisir pour ces décompositions sont la taille des matrices facteurs (et donc des noyaux) et le nombre d'éléments R dans la somme de décomposition.

Cambridge Nous avons testé la méthode de classification par HOPLS sur une base de données de vidéos simples : Cambridge Hand Gesture Dataset [51]. Cette base est constituée de 900 séquences d'images, réparties en 9 catégories de gestes, qui sont définies par 3 formes de main et 3 mouvements (illustration 3.25).

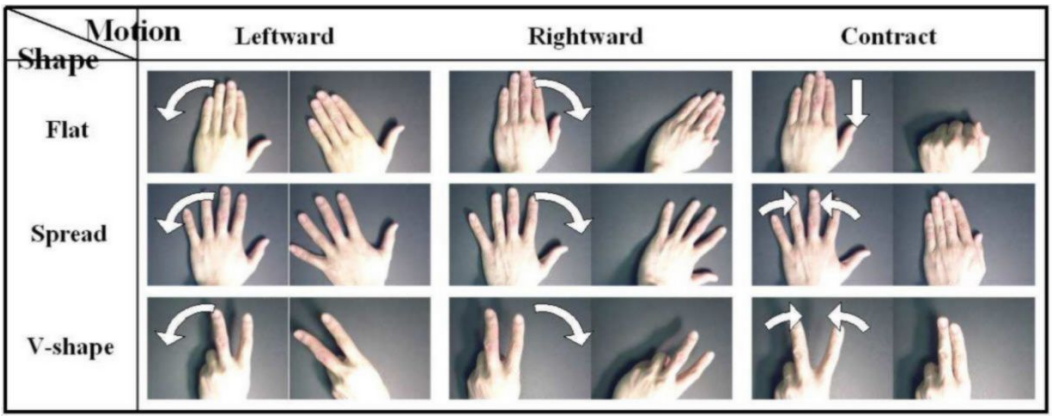


FIGURE 3.25 – Illustration des 9 classes de gestes différents contenus dans la base Cambridge Hand Gesture Dataset (illustration provenant [50]).

Les vidéos de cette base sont de taille $320 \times 240 \times T$ avec T allant de 37 à 119. Nous avons redimensionné les vidéos en $20 \times 20 \times 20$ (illustration 3.26) pour pouvoir utiliser HOPLS sur cette base. En effet, comme HOPLS fonctionne par recherche des valeurs propres, le tenseur initial ne doit pas être de taille trop importante. Nous avons donc mis les images en niveaux de gris, et redimensionné les vidéos spatialement, mais également temporellement pour normaliser la taille des vidéos.

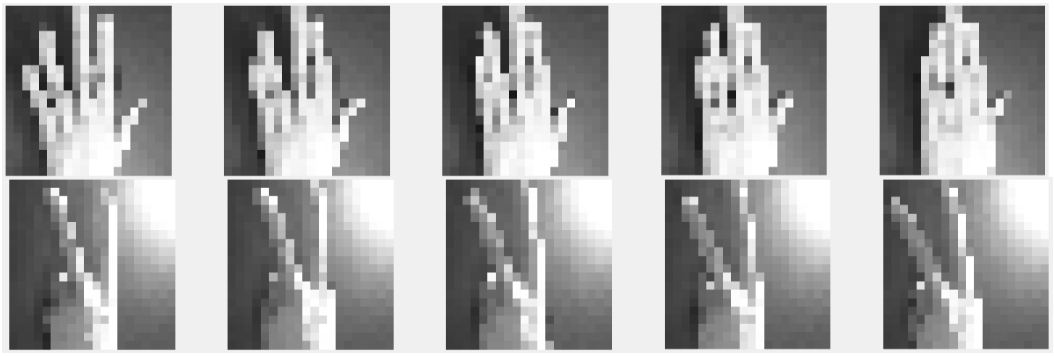


FIGURE 3.26 – Deux exemples de vidéos de la base après avoir été mises en niveaux de gris et redimensionnées en $20 \times 20 \times 20$. De gauche à droite, nous avons les images n°4,8,12,16 et 20. Sur la première ligne est présenté une vidéo de la classe {"spread" , "contract"} et sur la seconde ligne est présenté une vidéo de la classe {"v-shape" , "leftward"}.

L'application de HOPLS sur la base de Cambridge nous fournit alors l'équation de décomposition suivante :

$$\begin{aligned} \mathcal{X} &= \sum_{r=1}^{R} \mathcal{G}_r \times_1 t_r \times_2 P_r^{(1)} \times_3 P_r^{(2)} \times_4 P_r^{(3)} + \mathcal{E}_R \\ \mathcal{Y} &= \sum_{r=1}^{R} \mathcal{D}_r \times_1 t_r \times_2 Q_r^{(1)} + \mathcal{F}_R \end{aligned} \tag{3.25}$$

avec $\mathcal{X} \in \mathbb{R}^{N \times 20 \times 20 \times 20}$ le tenseur contenant les N images, $\mathcal{Y} \in \mathbb{R}^{N \times 10}$ l'ensemble des annotations, $t_r \in \mathbb{R}^N$ les vecteurs latents, les matrices facteurs $P_r^{(i)} \in \mathbb{R}^{20 \times L_i}$ et $Q_r^{(1)} \in \mathbb{R}^{9\ timesK_1}$. Pour être précis, $N = 720$ car nous avons gardé 180 exemples pour l'ensemble de test. Les tenseurs noyau associés sont $\mathcal{G}_r \in \mathbb{R}^{L_1 \times L_2 \times L_3}$ et $\mathcal{D}_r \in \mathbb{R}^{K_1}$. Les paramètres à fixer sont ici R, L_1, L_2, L_3 et K_1.

Sur le schéma 3.27, nous avons illustré l'application de cette décomposition sur Cambridge. Tout d'abord, les vidéos stockées dans un tenseur $\mathcal{X}$ et les labels dans un tenseur $\mathcal{Y}$. Puis la décomposition HOPLS de ces tenseurs définit par l'équation 3.25 est calculée de sorte à optimiser l'approximation par rapport aux exemples d'apprentissage. En phase de test, les noyaux, $\mathcal{G}_r$ et $\mathcal{D}_r$, ainsi que les matrices facteurs liées aux modes provenant de la donnée, soit $P_r^{(i)}$ et $Q_r^{(1)}$, qui correspondent aux dimensions spatiales, à la dimension temporelle et à la dimension des classes. Uniquement les vecteurs latents t_r varient en test. Les vecteurs latents sont d'ailleurs également des matrices facteurs mais cette fois liées au mode 1 qui est de dimension N et qui correspondant à la dimension des numéros d'échantillons ; soit la seule dimension partagée entre les tenseurs initiaux à lier. À l'arrivée d'une nouvelle vidéo x, on utilise les éléments fixes de la première décomposition pour estimer ses variables t_r. Puis on utilise les éléments fixes de la seconde décompo-

sition pour estimer le vecteur de label y.

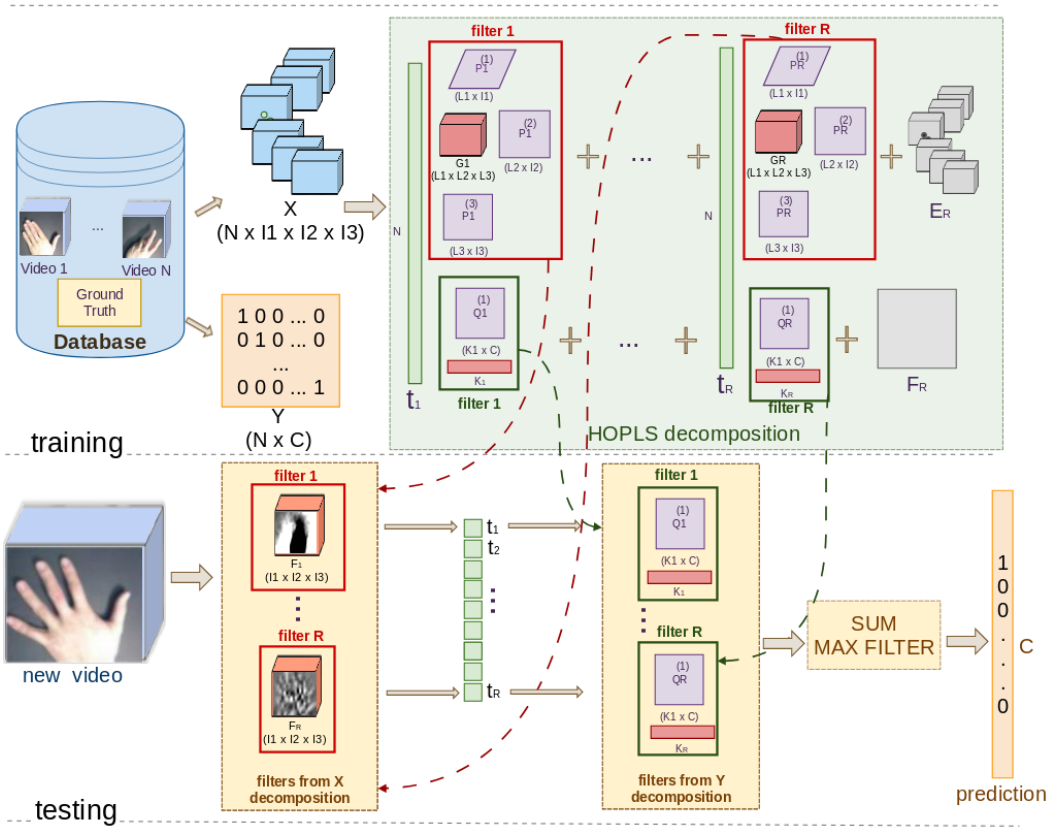


FIGURE 3.27 – Système d'utilisation d'HOPLS sur Cambridge dans un objectif de prédiction.

Le modèle apportant les meilleures performances est celui ayant les hyper-paramètres les plus élevés : $L_i = 19$ et $K_1 = 8$. Pour simplifier la comparaison, par la suite, on désignera un modèle par le choix de ses hyper-paramètres : par exemple, le modèle m-[19,19,19,8]. Et tous les modèles seront décomposés avec HOPLS jusqu'à $R = 200$ facteurs. Sur nos vidéos redimensionnées, ce modèle atteint un taux d'erreur de 30.00 % avec $R = 138$ facteurs. La figure 3.28 illustre le taux d'erreur en fonction du nombre de vecteurs latents R.

On peut tout d'abord remarquer que le taux d'erreur diminue fortement durant l'estimation des 50 premiers filtres, puis décroît lentement jusqu'à atteindre son minimum en $R = 138$. Cette décroissance est cohérente avec la convergence de la méthode. HOPLS est un algorithme qui calcule itérativement la décomposition sur le tenseur initial : on lie $\mathcal{X}$ et $\mathcal{Y}$, puis on leur soustrait leur approximation par décomposition et on répète l'opération de liaison sur les résidus. A chaque itération, l'algorithme estime les décompositions ayant des erreurs minimales $\mathcal{E}_R$ et $\mathcal{F}_R$. Ainsi les décompositions qui approchent au mieux les tenseurs initiaux $\mathcal{X}$

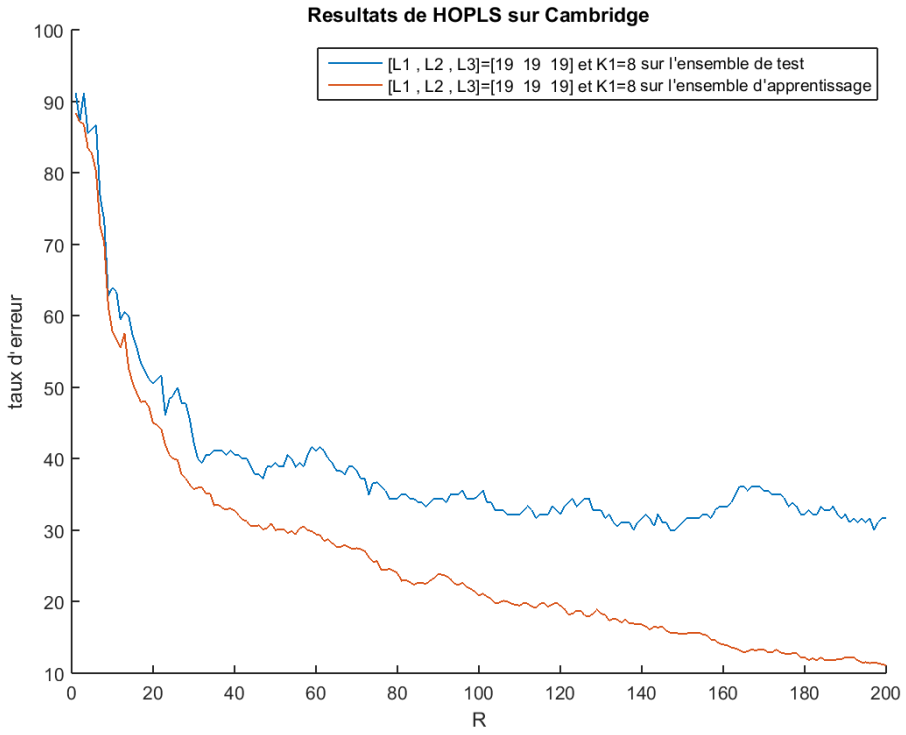


FIGURE 3.28 – Évolution du taux d'erreur en fonction du nombre de vecteurs latents R.

et $\mathcal{Y}$ sont choisies en premier. On peut d'ailleurs remarquer que cette décroissance est plus constante sur l'ensemble d'apprentissage.

Nous allons maintenant visualiser les éléments extraits de cette décomposition permettant de créer les vecteurs latents à partir de nouveaux éléments et ainsi de prédire sa classe. Nous nous intéressons donc plus particulièrement aux matrices facteurs $P_r^{(i)}$ et au tenseur noyau $\mathcal{G}_r$, car ce sont ces éléments, ceux de la décomposition de $\mathcal{X}$, qui s'appliquent au contenu vidéo pour en extraire des motifs. Plus précisément, nous nous intéressons aux éléments $\mathcal{G}_r \times_2 P_r^{(1)} \times_3 P_r^{(2)} \times_4 P_r^{(3)} \in \mathbb{R}^{20\times20\times20}, \forall r$, car c'est le produit de ces éléments avec les t_r qui permet de reconstituer les vidéos. Dans la suite, nous appellerons ces éléments les filtres du modèle. Ainsi, les filtres du modèle HOPLS ont exactement de la même taille que la donnée initiale. Dans l'illustration 3.29, nous pouvons voir certains filtres du modèle m-[19,19,19,8]. On retrouve le côté itératif d'HOPLS avec ces filtres car plus R est grand, plus le filtre contient des détails.

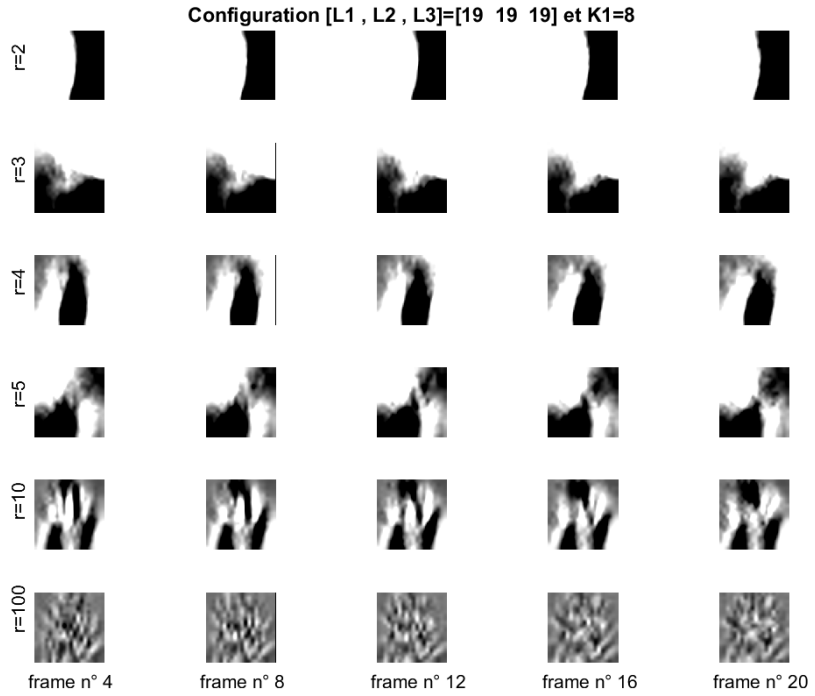


FIGURE 3.29 – Illustration des quatres premiers filtres (par ligne) du HOPLS pour la configuration $L_i = 19$ et $K_1 = 8$. Ces filtres étant des vidéos de 20 frames, les frames suivantes ont été affichées dans l'ordre de gauche à droite : 4,8,12,16,20.

Le choix des hyper-paramètres est important pour la méthode HOPLS car les résultats peuvent varier de 20 % entre deux choix de configuration. La figure 3.30 illustre l'évolution de taux d'erreur en fonction de R pour différentes configurations.

En règle générale, plus les hyper-paramètres sont élevés, meilleure est la classification. En comparant les filtres du modèles m-[19,19,19,8] et m-[1,1,1,1] (cf illu. 3.31), on remarque que plus les L_i sont faibles, plus les formes contenues dans le filtre sont incurvées selon ces dimensions i. Par contre, l'augmentation des détails selon R est présente de façon similaire dans les deux configurations.

La configuration m-[5,19,19,8] atteint un taux de 30.00% comme m-[19,19,19,8] mais plus tard dans l'algorithme, à $R = 194$ au lieu de $R = 138$. D'ailleurs, la restriction sur L_1 n'influe pas beaucoup sur les résultats. Sachant que L_1 est la dimension correspondant au temps (initialement de 20), cela peut être dû au fait que notre base ne contient pas de mouvements complexes et qu'ils peuvent être ainsi détectés par des filtres contenant peu de variations dans le temps. La restriction au niveau de la dimension correspondant à la classe, soit la dimension de taille K_1, impacte le taux d'erreur de 10 % pour certaines configurations (m-[10,10,10,8]

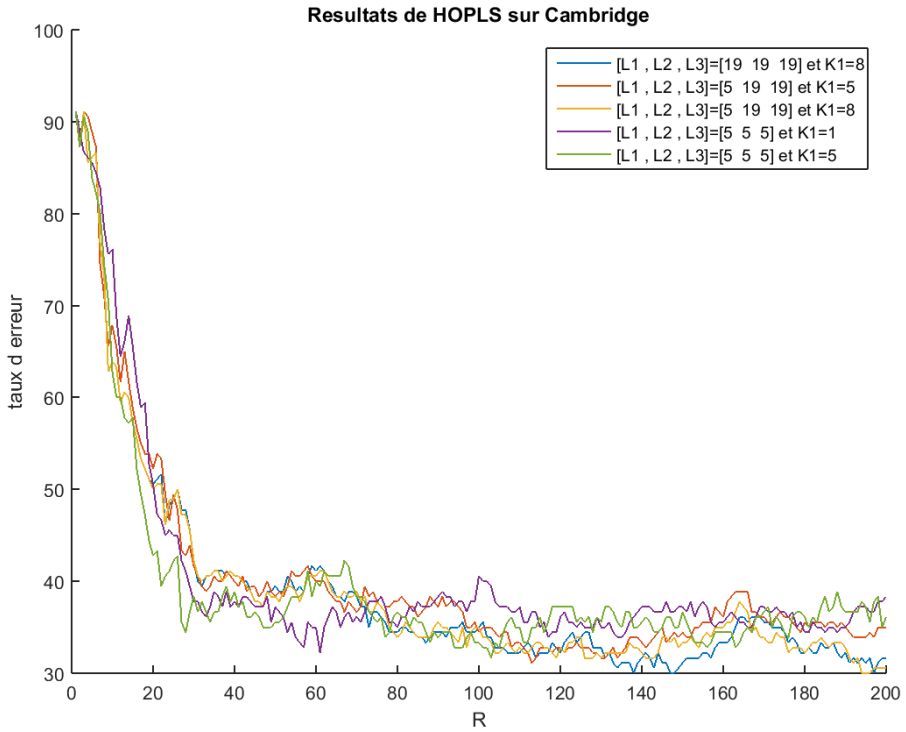


FIGURE 3.30 – Evolution du taux d'erreur en fonction du nombre de vecteurs latents R pour plusieurs configurations.

atteint 33 % alors que m-[10,10,10,1] obtient 40 %). Cette plus forte sensibilité à la réduction peut être due au fait que cette dimension est d'une part déjà assez petite, et d'autre part, c'est une dimension discrète et non ordonnée. On peut alors estimer que la dimension des classes est plus sensible que les autres dimensions. Au niveau des dimensions spatiales L_2 et L_3, nous les avons toujours fait varier symétriquement. On remarque que leurs influences sont également moindres tout comme la dimension temporelle mais qu'elle impacte très négativement la reconnaissance quand elles sont excessivement basses ($L_2 = 1$ et $L_3 = 1$).

Ci-dessous, vous pouvez visualiser les filtres de certains modèles.

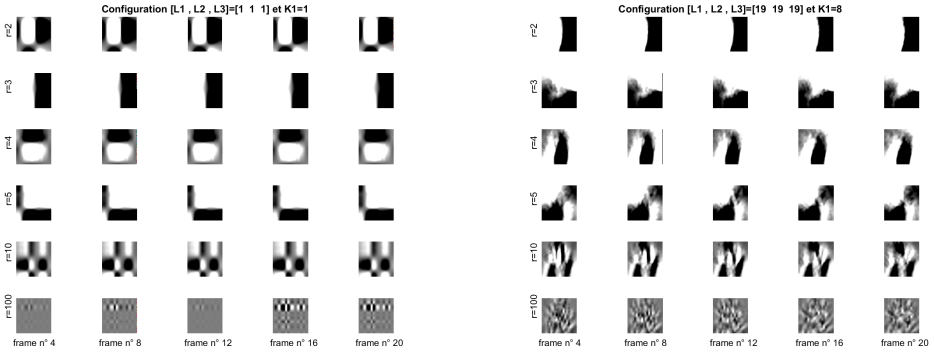


FIGURE 3.31 – Visualisation des filtres des modèles m-[19,19,19,8] et m-[1,1,1,1].

Configuration $[L_1, L_2, L_3, K_1]$	Taux d'erreur minimum	R
1,1,1,3	43.89	25
1,1,1,1	40.56	128
10,10,10,1	40.56	92
1,1,1,2	38.33	70
1,1,2,1	37.22	42
10,10,10,5	35.56	173
10,10,10,8	33.89	126
5,10,10,8	32.78	147
5,5,5,1	32.22	61
5,5,5,5	31.67	103
5,19,19,5	31.11	113
5,19,19,8	30.00	194
19,19,19,8	30.00	138

TABLE 3.1 – Taux d'erreur en fonction de la configuration et nombre de vecteurs latents R où ce taux d'erreur minimal a été atteint.

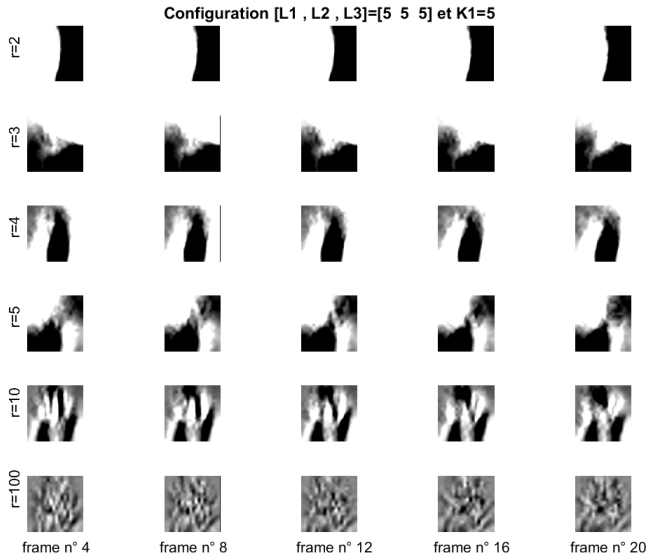


FIGURE 3.32 – Illustration des filtres du model m-[5,5,5,5].

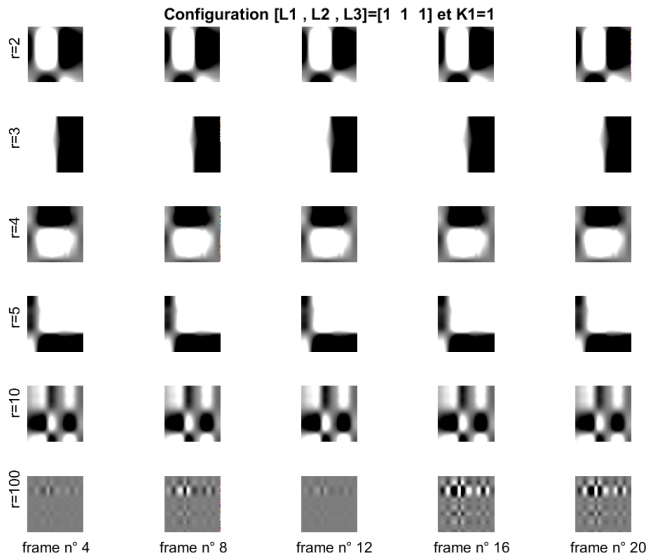


FIGURE 3.33 – Illustration des filtres du model m-[1,1,1,1]

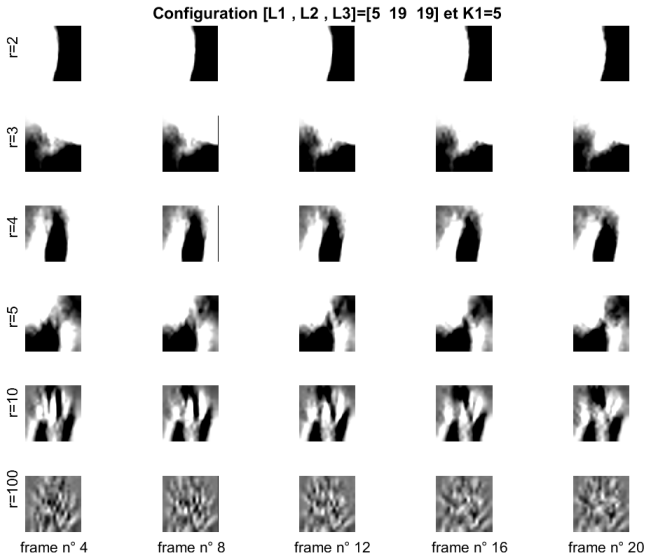


FIGURE 3.34 – Illustration des filtres du model m-[5, 19, 19,5]

Pour finir, bien qu'HOPLS extrait des filtres permettant de lier la donnée et son label, il n'atteint pas des taux de reconnaissance suffisants sur une base de vidéos simples. Une contrainte récurrente des méthodes tensorielles est le besoin de considérer, lors de la décomposition, tous les échantillons de la base dans un même tenseur ; ce qui engendre rapidement des problèmes de stockage selon la taille de la base. C'est pourquoi nous avons redimensionné la donnée en taille $20 \times 20 \times 20$. Dans le but d'expérimenter l'extraction d'information implicite par les méthodes tensorielles sur des bases de données plus fournies, nous passons à l'expérimentation de la méthode CIFA. Contrairement à HOPLS, cette méthode n'explore qu'une classe à la fois et demande ainsi moins de stockage.

3.3.2 sur CIFA

Dans cette expérience, nous avons voulu analyser l'extraction de composants communs par CIFA dans un cadre de classification.

Dans cette expérience, nous avons étudié l'extraction par CIFE sur plusieurs bases de données différentes pour analyser le comportement de la méthode d'extraction dans différentes circonstances. Nous allons tout d'abord présenter et discuter visuellement des représentants communs extraits. Ensuite, nous présenterons les méthodes de classification et les taux de reconnaissance obtenus.

3.3.2.1 Extraction et Visualisation de représentants communs

Pour rappel, la méthode CIFE présentée dans la section précédente (cf subsec. 3.2.3), permet d'extraire des éléments communs sur un ensemble de données constituées en groupes. Or, nous voulons utiliser CIFE pour extraire des représentants communs selon les classes pour pouvoir classifier de nouveaux éléments. Dans un cadre de classification général, il n'y a pas de sous-groupes par classe. Nous les avons donc formés aléatoirement pour pouvoir appliquer CIFE (cf illustration 3.35).

Nous allons donc utilisé l'équation suivante pour extraire les représentants communs.

$$\begin{aligned} &\min_{\overline{A},\breve{A}_n} \sum_{n\in\mathcal{N}} \|Y_n - \overline{A}\overline{B}_n^T - \breve{A}_n\breve{B}_n^T\|_F^2, \\ \text{tels que } &\overline{A}^T\overline{A} = I_C, \breve{A}_n^T\breve{A}_n = I_{R_n-C}, \\ &\overline{A}^T\breve{A}_n = 0, n \in \mathcal{N} \end{aligned} \tag{3.26}$$

avec $\mathcal{Y} = \{Y_n \in \mathbb{R}^{D_{size}\times J_n} : n \in \mathcal{N}\}$ l'ensemble des $\sum_n J_n$ éléments de la classe k , $\mathcal{N} = \{1, 2, ..., N\}$ et C le nombre de variables latentes en communs. Dans nos expériences, nous créons en général $N = 2$ ou 3 sous-groupes.

Et les représentants communs $\overline{F}$ sont ensuite extraits de $\overline{A}$ par une orthogonalisation. Nous avons alors C $\overline{F} \in \mathbb{R}^{1\times D_{size}}$ pour chaque classe. Ce nombre de représentants est directement imposé par l'algorithme COBE. Les représentants communs ont exactement la même taille que les données initiales.

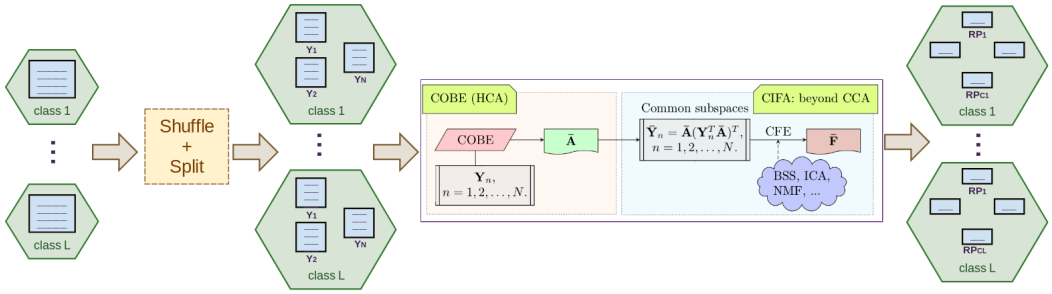


FIGURE 3.35 – Système d'extraction des représentants communs à partir d'une base de données annotées générique.

Dans cette partie, nous allons étudier ces représentants communs sur différentes bases.

Yale Tout d'abord, pour comparer notre extraction avec celle des auteurs [124], nous avons extrait des représentants communs de la base Yale [32]. Sur la figure suivante, vous pouvez visualiser à gauche les représentants extraits à partir de

groupes formés selon la direction lumineuse. Sur la droite, nous présentons les trois premiers représentants communs extraits de chaque sous-ensemble contenant uniquement une classe et formés de sous-groupes construit aléatoirement. Sur cette base, la classe d'une image est l'identité du sujet.

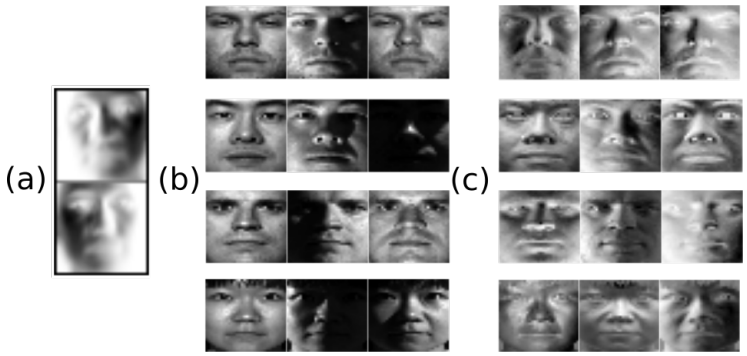


FIGURE 3.36 – (a) Représentants communs de la base Yale provenant de l'article CIFA [124] (b) Échantillons de 4 sujets de la base Yale (c) Représentants communs extraits des classes des sujets présentés en (b).

On remarque que les représentants communs reprennent les traits de leur sujet respectif avec différentes illuminations. Ces illuminations sont tellement précises que le représentant commun fait penser à une reproduction 3D du visage du sujet.

Cambridge La base de données Cambridge a été présenté dans la section précédente. Rappelons que cette base contient des vidéos de taille 320x240 allant de 35 à 117 frames. Comme précédemment, nous avons redimensionné chaque vidéo en 20x20x20. Nous avons également extrait des composants pour la taille 48x64x20.

Sur l'illustration 3.37, on peut voir quelques représentants communs par classe de la base Cambridge. On remarque tout d'abord qu'on retrouve la ressemblance entre les représentants et les vidéos originales de la classe avec des différentes directions lumineuses. On trouve également des zones plus homogènes, plus lisse. De plus, on peut noter que plus C est grand, plus la vidéo du représentant contient des textures complexes. On retrouve ici la complexification des filtres que l'on avait identifiés avec HOPLS. C'est une conséquence de la complexification du modèle avec l'augmentation des facteurs décompositions tensorielles.

Les bases de vidéos d'action Nous avons également extrait les représentants communs avec CIFE sur des bases de vidéos d'actions : KTH, UCF101 et IsoGD. Ces bases illustrent des actions humaines. Les vidéos ont un arrière-plan fixe dans la plupart des échantillons. KTH, UCF101 et IsoGD contiennent respectivement 6, 101 et 249 classes. Les arrières plans sont différents d'un échantillon à l'autre dans les bases UCF101 et IsoGD. Comme précédemment, nous avons redimensionné et

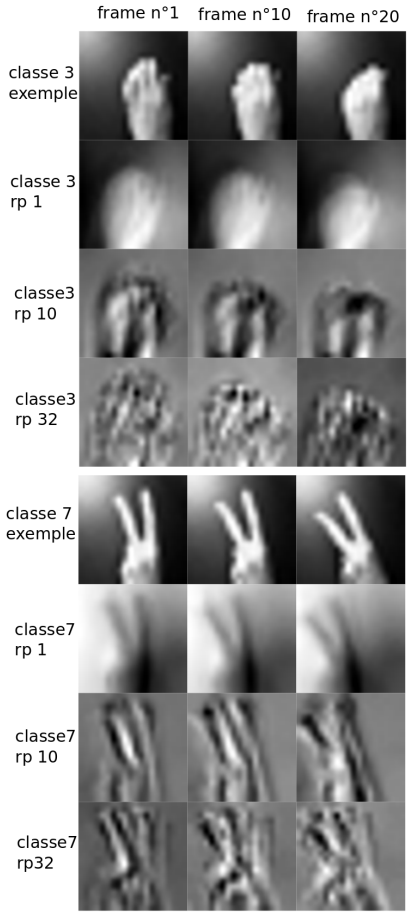


FIGURE 3.37 – Illustration d'un exemple et des représentants communs 1, 10 et 32 des classes 3 et 7 de la base Cambridge Hand Gesture Dataset. 'rp' signifie représentant.

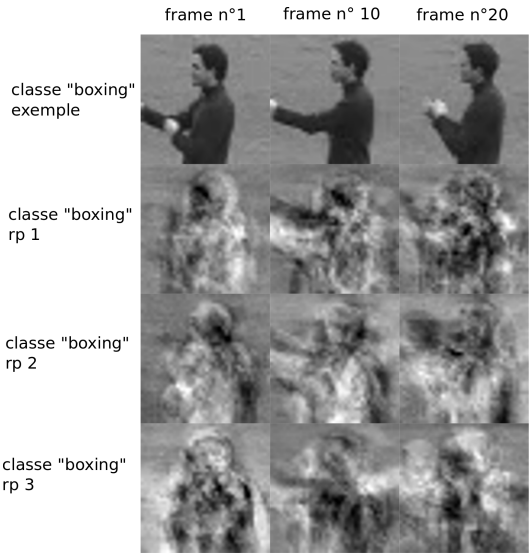


FIGURE 3.38 – Illustration d'un exemple et les représentants communs 1,2 et 3 de la classe "boxing" de KTH

normalisé les vidéos pour les adapter à l'entrée de CIFE. Sur ces trois bases, les représentants que nous obtenons nous mènent aux mêmes conclusions.

On remarque que les représentants communs sont bien moins homogènes sur les bases d'actions humaines que sur Cambridge. La principale différence entre ces bases et Cambridge est la variation spatiale. En effet, sur Cambridge, les vidéos de mains sont très normalisées ; alors que pour les vidéos de nos dernières bases, les sujets se déplacent en montrant l'action. Et on retrouve ces variations spatiales dès les premiers représentants communs extraits par l'aspect texturisé des frames. En visualisant ces représentants vidéos, on perçoit un mouvement régulier et continu de l'action représentée.

Il semble que l'alignement spatial et temporel des données soit une étape nécessaire avant l'extraction des éléments communs par CIFE. D'ailleurs, les auteurs n'ont évalué leur méthode que sur des données très corrélées spatialement comme la base Yale ou une base de radios de poumons [124]. Cet effet est sûrement dû au caractère linéaire de la méthode CIFE.

Pascal VOC Dans les bases précédentes, nous avions relativement peu d'échantillons par classe (au minimum 60 pour isoGD, 100 pour KTH, 100 pour UCF101). Cependant, il est connu que l'effet d'une forte variation peut être diminué par un enrichissement de l'ensemble d'apprentissage. Nous avons donc voulu expérimenter CIFE sur une base contenant des variations spatiales, mais également plus d'échantillons par classe. Pour des raisons de calcul, nous avons choisi une base d'images : PascalVOC. PascalVOC est une base de 17125 images réparties en 20 classes et chaque classe contient plus de 300 exemples.

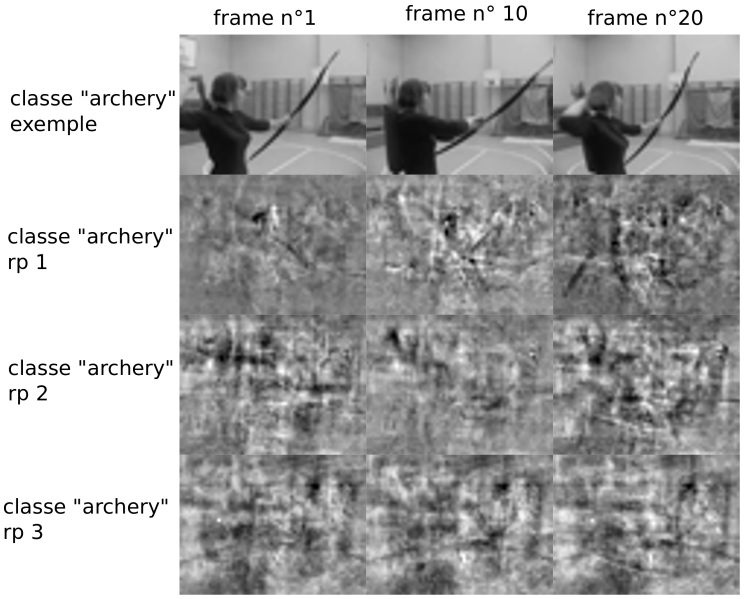


FIGURE 3.39 – Illustration d'un exemple et les représentants communs 1,2 et 3 de la classe "archery" de UCF

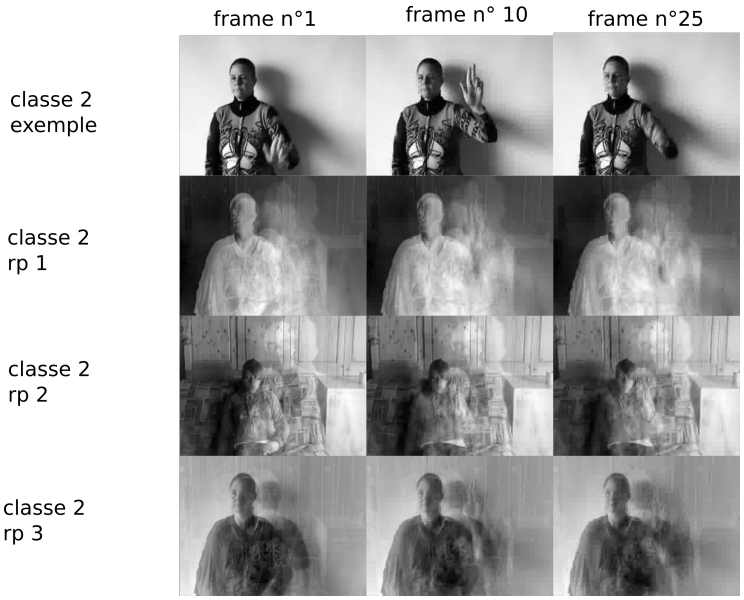


FIGURE 3.40 – Illustration d'un exemple et les représentants communs 1,2 et 3 de la classe 2 d'IsoGD

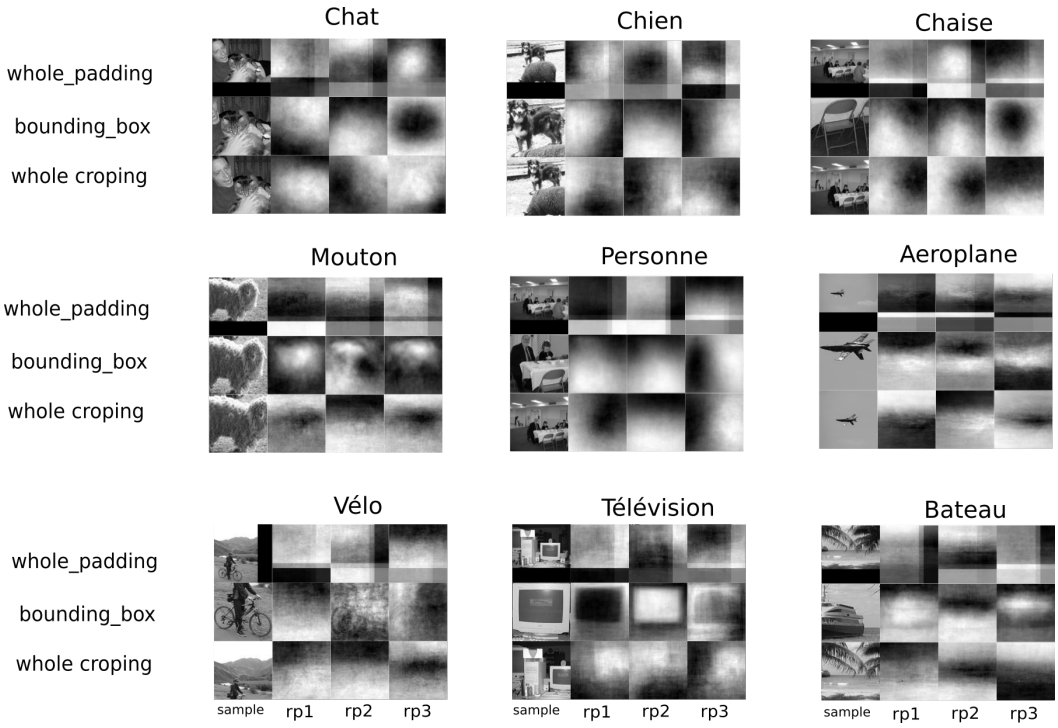


FIGURE 3.41 – Illustration de représentants communs extraits de la base PascalVoc. On peut voir les 3 premiers représentants communs de 9 classes selon 3 différentes normalisations.

Cette base de données est composée d'images de différentes tailles. Nous avons utilisé 3 stratégies de normalisation :

— Remplir l'image par des pixels noirs et redimensionner en carré de taille fixe (whole_padding)

— Rogner à l'intérieur de l'image et redimensionner en carré de taille fixe (whole_croping)

— Avec utilisation des "bounding boxes", rogner un carré à l'intérieur de la boite englobante et le redimensionner (bounding_box_out)

— Avec utilisation des "bounding boxes", rogner un carré contenant la boite englobante et le redimensionner (bounding_box_in)

Il est à noté que la base de données PascalVoc contient beaucoup de variations spatiales, en particulier d'échelle. De plus, les boites englobantes sont parfois approximatives.

Sur la figure 3.41, vous pouvez visualiser les représentants de certaines classes selon les différentes stratégies de normalisation.

Tout d'abord, on retrouve évidemment les bandes noires de la première stratégie. Globalement, les deux dernières stratégies utilisant des boites englobantes permettent d'obtenir des représentants communs, moins lisses.

Nous avons observé que les variations spatiales rendaient les représentants plus texturés. Avec la base PascalVoc, nous remarquons qu'un grand nombre d'exemples et une grande variation spatiale rendent les représentants communs extrêmement lisses, comme pour la classe chat, chien et chaise.

Pour certaines classes contenant moins de variations, les représentants illustrent une silhouette de la classe : personne, mouton, avion, bateau. Ce sont des classes dont les exemples sont souvent semblables : les avions sont photographiés de loin et en général dans le ciel.

Enfin avec un objet comme la télévision, on obtient un représentant commun visuellement ressemblant, mais tout de même lisse.

Pour résumer, les représentants contiennent plusieurs motifs communs aux vidéos originales. Plus il y a des représentants, plus les derniers seront texturés. Plus l'ensemble de base contient des variations spatiales, plus les représentants seront texturés et indiqueront un amas d'instances de la classe. Enfin, si l'ensemble d'apprentissage est conséquent, les représentants deviennent complètement lisses. Nous avons analysé visuellement les représentants communs, nous allons étudier le comportement de méthode de classification sur ces représentants communs.

3.3.2.2 Classification à partir des représentants communs de CIFE

Dans cette partie, nous considérons que nous avons extraits les représentants communs et nous avons donc trois ensembles : l'ensemble d'apprentissage (*train*), des représentants communs à chaque classe (*rp*) et l'ensemble de test (*test*).

Nous allons utiliser comme référence la méthode proposée par les auteurs pour la classification : la distance entre un nouvel élément $\mathcal{X}_{new}$ et la classe k est définie par

$$d(\mathcal{X}_{new}, k) = \sqrt{\sum_{c=1}^{C} < \overline{F}, \mathcal{X}_{new} >^2} \tag{3.27}$$

Cette distance illustre la distance entre un nouvel élément et une classe par la somme des corrélations entre cet élément et les représentants de la classe. Nommons cette prédiction la correspondance par corrélation.

Nous comparons cette méthode de classification à un SVM de trois types de noyaux possibles : linéaire, polynomial d'ordre 2 et à noyau de base radiale (rbf). Ces SVMs seront entraînés sur une fusion de l'ensemble d'apprentissage et de représentants.

Les tableaux suivants présentent une liste non-exhaustive des résultats obtenus sur les différentes bases avec les différentes méthodes de prédiction. Les méthodes de classification sont différentes par trois critères : la taille de normalisation de départ, l'utilisation ou non de l'étape de séparation des sources (BSS) proposée par les auteurs, et la méthode de classification.

Sur la base Cambridge, un SVM avec noyau polynomial surpasse la distance par corrélation. Cependant, lorsqu'on enlève l'étape de séparation des représentants communs (BSS), la distance par corrélation améliore son taux de reconnaissance. Dans cette configuration, les SVM ont tendance à sur-apprendre, en particulier avec le noyau gaussien. Les résultats sont globalement meilleurs pour la taille inférieure, $20 \times 20 \times 20$.

Normalisation des entrées	Méthode de classification	Taux de reconnaissance Train / Test
Taille 20x20x20 ; Rp avec Bss	Correspondance par corrélation	84 / 83
Taille 20x20x20 ; Rp avec Bss	SVM avec noyau linéaire	95 / 72
Taille 20x20x20 ; Rp avec Bss	SVM avec noyau polynomial 2	100 / 91
Taille 20x20x20 ; Rp avec Bss	SVM avec un noyau rbf	100 / 65
Taille 46x64x20 ; Rp avec Bss	Correspondance par corrélation	98 / 77
Taille 46x64x20 ; Rp avec Bss	SVM avec noyau linéaire	100 / 69
Taille 46x64x20 ; Rp avec Bss	SVM avec noyau polynomial 2	100 / 81
Taille 46x64x20 ; Rp avec Bss	SVM avec un noyau rbf	100 / 11
Taille 20x20x20 ; Rp sans Bss	Correspondance par corrélation	97 / 96
Taille 20x20x20 ; Rp sans Bss	SVM avec noyau linéaire	100 / 67
Taille 20x20x20 ; Rp sans Bss	SVM avec noyau polynomial 2	100 / 93
Taille 20x20x20 ; Rp sans Bss	SVM avec un noyau rbf	100 / 11
Taille 46x64x20 ; Rp sans Bss	Correspondance par corrélation	100 / 88
Taille 46x64x20 ; Rp sans Bss	SVM avec noyau linéaire	100 / 57
Taille 46x64x20 ; Rp sans Bss	SVM avec noyau polynomial 2	100 / 66
Taille 46x64x20 ; Rp sans Bss	SVM avec un noyau rbf	100 / 11

FIGURE 3.42 – Taux de reconnaissance sur la base Cambridge selon différentes configurations de la taille de l'entrée et différentes méthodes de classification.

Sur IsoGD, on retrouve un meilleur taux de reconnaissance pour la distance sans BSS qu'avec ; alors que pour les SVM, les résultats sont stables. Il est à noter que la distance par corrélation est effectuée sans PCA alors que les SVM sont entraînés sur les vidéos réduites par PCA (avec un nombre de composants variant entre 300, 1000 et 3000). Sur IsoGD, les taux de reconnaissance sont meilleurs avec une plus grande taille de vidéos d'entrée. Pour information, le réseau de neurones C3D atteint un taux de 56% sur cette base. Ainsi, nous n'obtenons que 40% de classification sur IsoGd qui est une base de vidéo plus difficile que Cambridge. En effet, cette base de 240 classes contient environ 60 vidéos par classe pour une taille de données de $35x64x48$, soit 107520 composants. De plus, nous l'avons choisir pour tester CIFA sur des données contenant plus de variations spatiales. En gardant pour objectif d'évaluer CIFA dans des cas réels, nous l'avons expérimentée sur PascalVOC qui ne contient que 20 classes, au minimum 300 échantillons par classe et demande moins de stockage (ces échantillons étant des images).

Normalisation des entrées	Méthode de classification	Taux de reconnaissance Train / Test
Taille 35x64x48 ; sans Bss	Correspondance par corrélation	96 / 39
Taille 35x64x48 ; sans Bss	SVM avec noyau linéaire ; PCA 300	85 / 30
Taille 35x64x48 ; sans Bss	SVM avec noyau linéaire ; PCA 1000	96 / 33
Taille 35x64x48 ; sans Bss	SVM avec noyau linéaire ; PCA 3000	98 / 31
Taille 35x64x48 ; *avec Bss*	Correspondance par corrélation	37 / 16
Taille 35x64x48 ; *avec Bss*	SVM avec noyau linéaire ; PCA 300	85 / 30
Taille 35x64x48 ; *avec Bss*	SVM avec noyau linéaire ; PCA 1000	96 / 33
Taille 35x64x48 ; *avec Bss*	SVM avec noyau linéaire ; PCA 3000	98 / 31
Taille 35x32x24 ; sans Bss	Correspondance par corrélation	86 / 40
Taille 35x32x24 ; sans Bss	SVM avec noyau linéaire ; PCA 300	86 / 30
Taille 35x32x24 ; *avec Bss*	Correspondance par corrélation	34 / 16
Taille 35x32x24 ; *avec Bss*	SVM avec noyau linéaire ; PCA 300	86 / 30

FIGURE 3.43 – Taux de reconnaissance sur la base IsoGD selon différentes configurations de la taille de l'entrée et différentes méthodes de classification.

Pour PascalVOC, rappelons que nous avons utilisé différents protocoles de normalisation des tailles. Nous réunissons les résultats de "whole_padding" et "whole_croping" qui sont équivalents, de même pour "bounding_box_in et bounding_box_out". Ainsi nous avons la normalisation avec (bounding_box) et sans (whole) l'utilisation de la boite englobante pour normaliser les images d'entrées. Sur cette base, la distance par corrélation avec Bss reste moins performante que celle sans Bss. Les SVM sur-apprennent légèrement autant que la distance par corrélation ne se généralise pas en test. Il y a encore plus de sur-apprentissage avec les boites englobantes alors que nous avions vu précédemment que les représentants communs contenaient plus de motifs visuels de la classe représentée dans cette configuration. Pour finir, même avec plus d'échantillons par classe, la méthode CIFA n'est pas robuste aux variations spatiales.

Normalisation des entrées	Méthode de classification	Taux de reconnaissance Train / Test
Bounding box ; 50x50 ; sans Bss	Correspondance par corrélation	97 / 21
Bounding box ; 50x50 ; sans Bss	SVM avec noyau linéaire	90 / 17
Bounding box ; 50x50 ; sans Bss	SVM avec noyau rbf	99 / 22
Bounding box ; 100x100 ; sans Bss	Correspondance par corrélation	98 / 23
Bounding box ; 100x100 ; sans Bss	SVM avec noyau linéaire	92 / 17
Bounding box ; 50x50 ; avec Bss	Correspondance par corrélation	78 / 17
Bounding box ; 50x50 ; avec Bss	SVM avec noyau linéaire	90 / 17
Bounding box ; 50x50 ; avec Bss	SVM avec noyau rbf	99 / 22
Bounding box ; 100x100 ; avec Bss	Correspondance par corrélation	86 / 17
Bounding box ; 100x100 ; avec Bss	SVM avec noyau linéaire	91 / 17

FIGURE 3.44 – Taux de reconnaissance sur la base Pascal VOC sans l'utilisation des boites englobantes, selon différentes configurations.

Normalisation des entrées	Méthode de classification	Taux de reconnaissance Train / Test
Whole; 100x100 ; sans Bss	Correspondance par corrélation	66 / 22
Whole; 100x100 ; sans Bss	SVM avec noyau linéaire	60 / 14
Whole; 100x100 ; sans Bss	SVM avec noyau gaussien	61 / 18
Whole; 100x100 ; sans Bss	SVM avec noyau poly. d'ordre 2	56 / 12
Whole; 200x200 ; sans Bss	Correspondance par corrélation	68 / 25
Whole; 200x200 ; sans Bss	SVM avec noyau linéaire	60 / 14
Whole; 200x200 ; sans Bss	SVM avec noyau gaussien	66 / 10
Whole; 100x100 ; avec Bss	Correspondance par corrélation	58 / 19
Whole; 100x100 ; avec Bss	SVM avec noyau linéaire	59 / 15
Whole; 100x100 ; avec Bss	SVM avec noyau gaussien	61 / 18
Whole; 100x100 ; avec Bss	SVM avec noyau poly. d'ordre 2	58 / 12
Whole; 200x200 ; avec Bss	Correspondance par corrélation	65 / 23
Whole; 200x200 ; avec Bss	SVM avec noyau linéaire	60 / 14
Whole; 200x200 ; avec Bss	SVM avec noyau gaussien	65 / 10

FIGURE 3.45 – Taux de reconnaissance sur la base Pascal VOC, avec l'utilisation des boites englobantes, selon différentes configurations.

3.4 Conclusion

L'attrait et l'avantage principal de l'utilisation des tenseurs est leurs aspects multi-dimensionnels et ainsi le stockage respectant les dimensions naturelles et les différents attributs de la donnée. De plus, les décompositions permettent de compresser un tenseur et de le reconstruire à partir de données manquantes. La décomposition tensorielle par TPCA permet notamment de séparer une image de l'influence d'une caractéristique, par exemple l'intensité ou la direction lumineuse. Cette séparation de l'influence de caractéristique amène à penser que l'on peut extraire une description exempte de tout contexte en variant suffisamment les échantillons.

Cependant, les méthodes tensorielles subissent des désavantages du fait du caractère exclusivement linéaire des décompositions. D'une part, ces méthodes ne gèrent pas les changements de localisation dans la donnée. Elles ne sont alors pas robustes aux variations spatiales. On le remarque par exemple sur les représentants communs extraits par CIFA sur les bases d'actions humaines. Les exemples représentent des personnes boxant vers la droite ou vers la gauche et les représentants communs illustrent une activité de "boxing" dans les deux directions en même temps. D'autre part, ces méthodes ne gèrent pas les éléments bruités ou mal annotés (les "outliers") et les sous-groupes. Comme les critères d'optimisation sont globaux, un mauvais échantillon influe sur la moyenne des critères d'extraction et, par conséquent, perturbe les éléments extraits. Enfin, les techniques tensorielles s'appliquent généralement sur un seul tenseur contenant toute la base de données ce qui compliquent la gestion de la mémoire et l'exploration de l'ensemble des données. D'autant plus que les décompositions tensorielles demandent souvent d'effectuer des décompositions en valeurs singulières (SVD) sur les matricisations d'un tenseur conséquent, ce qui est très coûteux sur de grandes matrices. Donc les méthodes tensorielles actuelles ne sont pas robustes aux changements spatiaux, aux sous-groupes et aux cas particuliers.

Ces méthodes ont pourtant un pouvoir de compression, de complétion, et même d'extraction de connaissance certain. Ces méthodes doivent être donc utilisées sur des données normalisées spatialement et temporellement. Dans le chapitre suivant, nous allons donc nous intéresser à la normalisation temporelle de la vitesse d'exécution sur un ensemble de données d'une même classe.

CHAPITRE 4

Les Techniques de Déformation Temporelle

Contents

4.1 Introduction

Après avoir étudié et décrit explicitement le mouvement par l'évolution temporelle de singularités et après avoir étudié les méthodes d'extraction implicite du mouvement comme une information commune à un ensemble, nous nous concentrons maintenant sur l'analyse et la réduction de l'élasticité temporelle. Cette réduction de variation sera étudiée dans un objectif de classification.

Nous l'avons présenté comme le coeur de la réflexion de cette thèse : une action possède des informations de mouvements contenant des variations de vitesse d'exécution. Ces différentes vitesses et ces différents styles d'exécution d'une même action augmentent la variation intra-classe. Or une tâche de classification sera d'autant plus aisée que ces classes auront une faible variation intra-classe ainsi qu'une forte variation inter-classe. Ainsi, dans cette section, nous cherchons à minimiser l'élasticité temporelle à l'intérieur d'une classe, ce qui revient à déformer temporellement les séquences. La technique utilisée en vision par ordinateur pour aligner temporellement des séquences est la déformation temporelle dynamique (*Dynamic Time Warping* DTW).

Dans la prochaine section, nous allons exposer les différents travaux autour de DTW dans l'exploration de données. Dans la section 4.3.1, nous présenterons cette méthode de déformation temporelle et ses extensions. Enfin, nous présenterons les expériences que nous avons menées pour normaliser la vitesse d'exécution à l'intérieur d'une classe et l'amélioration de la classification d'action par l'apport de cette normalisation temporelle.

4.2 La déformation temporelle et la reconnaissance d'action

La déformation temporelle est aujourd'hui appliquée à différents domaines comme l'infographie ou la bio-informatique[11, 1]. Elle a fait son entrée dans l'exploration de données par son application à la reconnaissance vocale [80]. Beaucoup de méthodes dans ce domaine utilisent les chaînes de Markov cachées (HMM) ou les réseaux récurrents (RNN), cependant leur utilisation est faite dans un objectif de comparaison de séquences déjà découpées en unités discrètes, plutôt que dans un objectif de synchronisation temporelle [87, 9].

DTW est une méthode d'alignement temporel par la mise en correspondance de deux séquences. L'objectif est de trouver deux déformations temporelles qui maximiseront la corrélation entre ces deux séquences. Or la corrélation entre deux vidéos dépend de la représentation des éléments (les images) et de la distance entre ces éléments. Par exemple, les images peuvent être représentées par l'image originale en RGB, l'image en noir et blanc, une image binaire différenciant l'avant de l'arrière-plan (généralement utilisées dans la reconnaissance d'action humaine et représentées des silhouettes), ou encore l'image peut être décrite par une représentation connue : sac de mots, deep features, skeleton, ... La distance entre ces représentations peut alors être la distance euclidienne ou tout autre distance induite des p-normes vectorielles, une distance géodésique ou une distance adaptée à la représentation choisie.

Par conséquent, tout comme la déformation temporelle, la représentation et la distance peuvent être optimisées par DTW dans le but d'améliorer la corrélation pré-supposée entre les séquences. En effet, la corrélation est présupposée puisqu'on aligne des séquences qui sont sémantiquement liées. Optimiser la représentation dans ce but est équivalent à rendre cette représentation spécifique aux éléments communs et insensible aux informations individuelles. Hsu *et al.* [39] proposent justement de conserver le résidu de la représentation pour décrire le style d'exécution d'un mouvement. Ils présentent ainsi une adaptation de DTW, appelée déformation de mouvement itératif (Iterative Motion Warping IMW) qui alterne entre l'optimisation de la déformation temporelle et l'optimisation de la représentation, tout en ajoutant comme régularisation une contrainte de lissage sur les représentations individuelles. Junejo *et al.* [45] utilisent DTW pour mettre en correspondance deux points de vue d'un même événement et trouver un descripteur invariant aux changements de point de vue. Des transformations spatiales et des représentations complexes ont émergé par la suite, notamment grâce à l'ap-

port des modèles de variété spatio-temporelle (spatio-temporal manifold model STM) et des distances géodésiques associées [104, 34]. DTW est donc principalement utilisée dans la recherche de représentations. Notons que cette recherche de représentation est souvent menée à partir de points 3D fournies par des capteurs de mouvements et non de vidéos 2D classiques. L'approche par variété de Vu *et al.*[104] proposent d'aligner des vidéos classiques en extrayant les squelettes 2D et en les considérant comme des projections de structures 3D existantes, contenant du bruit et des occlusions.

La méthode DTW a été adaptée à des cas précis, mais elle a été également étendue et généralisée à des schémas d'alignement plus génériques. Tout d'abord, suivant l'idée d' IMW d'introduire une transformation spatiale dans DTW, Zhou *et al.* [123] étendent DTW grâce à l'analyse de corrélations canoniques (CCA) qui projettent linéairement les données dans un espace latent maximisant la corrélation entre les projections. Cette méthode est nommée la déformation temporelle canonique (CTW). Cette projection linéaire est un moyen de hiérarchiser l'importance de chaque composant de la représentation dans le calcul de la corrélation et ainsi de différentier ce qui est corrélé de ce qui ne l'est pas. De plus, CTW est compatible à la multi-modalité puisque chaque séquence a sa propre projection et il est alors possible de projeter dans un espace latent commun des représentations sémantiquement différentes, comme dans le cas d'alignement de séquences des capteurs de position 3D et de séquences vidéos 2D. Puis, Trigeorgis *et al.* [98] proposent une transformation spatiale par des réseaux de neurones pour élargir les possibilités de représentations à des transformations non-linéaires complexes (DCTW).

Tandis que l'alignement est effectué par paire dans DTW, CTW et DCTW, Zhou *et al.* [122] proposent la méthode de déformation temporelle canonique généralisée (GCTW). Cette approche modélise la corrélation d'un ensemble de séquences par la somme de la corrélation de chaque paire, comme une multi-CCA (mCCA) [38]. De plus, les auteurs proposent une déformation non-linéaire, paramétrée par une combinaison de fonctions monotones.

Par conséquent, DTW et ses extensions sont des méthodes pratiques pour la mise en évidence d'éléments communs (le label en commun, l'action effectuée ou le même événement) et la réduction des différences entre ces séquences (la vitesse, le point de vue, le style). L'élasticité temporelle est une de ces variations. Ainsi, plusieurs taches d'exploration de données peuvent tirer avantage de cette réduction de variation intra-classe. Wang *et al.* [107] proposent d'apporter plus de poids aux correspondances qui apparaissent en début de vidéo dans la méthode GCTW (TCTW). Ces poids permettent de favoriser une prise de décision rapide dans un objectif de prédiction d'une action faite par une classification par k-NN. Trigeorgis *et al.* [97] proposent DDATW, une combinaison de DCTW avec l'analyse discriminante linéaire (LDA) qui permet d'aligner leurs vidéos à une labellisation temporelle. Par la suite, nous analyserons l'influence de l'alignement temporel sur l'élasticité et sur la classification.

4.3 Alignement temporel et Classification vidéo

4.3.1 DTW et ses extensions

Soit deux séquences temporelles $X = [x_1, \cdots, x_{n_x}] \in \mathbb{R}^{d \times n_x}$ et $Y = [y_1, \cdots, y_{n_y}] \in \mathbb{R}^{d \times n_y}$ de longueurs respectives n_x et n_y. DTW est un algorithme qui cherche à déformer X et Y tel que la somme des distances euclidiennes entre les séquences alignées soit minimisée. DTW cherche le minimum suivant :

$$\min_{\{p_x, p_y\} \in \Psi} J_{DTW} = \sum_{t=1}^{l} \|x_{p_t^x} - y_{p_t^y}\|^2 \tag{4.1}$$

où $l > max\{n_x, n_y\}$ est le nombre d'indices nécessaires pour aligner les échantillons et ainsi la longueur des séquences après déformation. Le l optimal est déterminé par l'algorithme DTW ainsi que les déformations $p^x \in \{1 : n_x\}^l$ et $p^y \in \{1 : n_y\}^l$. Ainsi, le i^{ieme} élément de X, x_i, est aligné au j^{ieme} élément de Y, y_j, si il existe un temps t tel que $p_t^x = i$ et $p_t^y = j$.

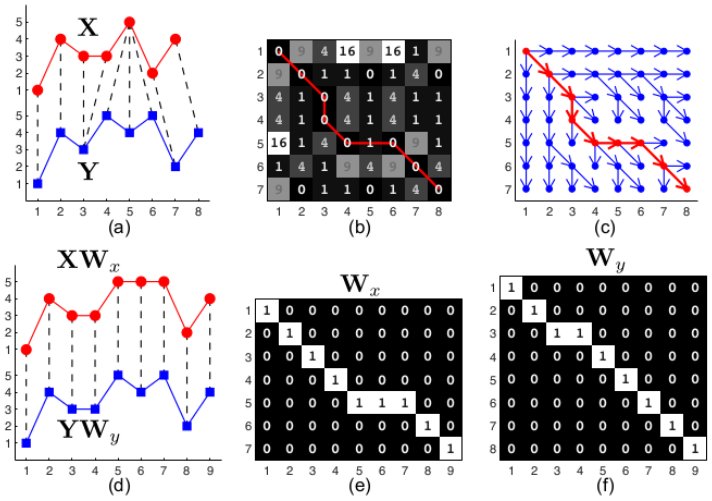


FIGURE 4.1 – Un exemple de DTW pour aligner deux séquences. (a) Deux séquences 1D ($n_x = 7$ et $n_y = 8$) et l'alignement optimal par DTW illustré en pointillés . (b) Matrice des distances euclidiennes entre les éléments des séquences, la courbe rouge est le chemin optimal ($l = 9$). (c) La pratique de la programmation dynamique illustrée par les possibilités de déplacements contraignant l'optimisation. (d) Le résultat des déformations sur les signaux. (e) et (f) Les matrices de déformation temporelle de chaque signal qui construites à partir des chemins p (illustration provenant de [122]).

Les déformations temporelles p^x et p^y sont chacune une séquence d'indices permettant localement d'allonger ou de comprimer la séquence. Ces chemins de déformation doivent respecter les contraintes suivantes :

— bornes : $[p_1^x, p_1^y] = [1, 1]$ et $[p_l^x, p_l^y] = [n_x, n_y]$
— monotonie : $t_1 \leq t_2 \Rightarrow p_{t_1}^x \leq p_{t_2}^x$ et $p_{t_1}^y \leq p_{t_2}^y$
— continuité : $[p_t^x, p_t^y] - [p_{t-1}^x, p_{t-1}^y] \in \{[0, 1], [1, 0], [1, 1]\}$

L'algorithme DTW consiste à construire une matrice de distances euclidiennes entre chaque élément des deux séquences et à trouver le chemin optimal dans cette matrice minimisant le coût du parcours ; *i.e.* la somme des cases traversées. La somme des cases traversées est égale à la distance entre les séquences alignées et donc inversement proportionnelle à leur corrélation. Ce chemin, à l'intérieur de la matrice de correspondances, permet de déterminer les déformations temporelles induites sur chacune des séquences. L'illustration 4.1 montre un exemple d'alignement par DTW. Bien que le nombre de chemins possibles soit exponentiel, l'algorithme DTW se base sur la programmation dynamique pour offrir la solution optimale avec une complexité $O(n_x n_y)$.

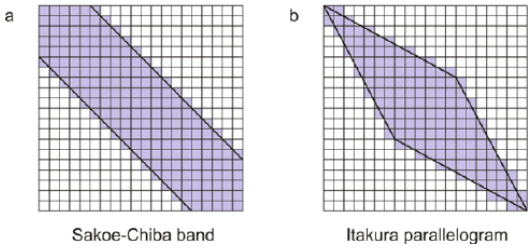


FIGURE 4.2 – Les deux contraintes globales les plus utilisées dans la littérature.

Notons qu'ici la contrainte sur les bornes est stricte, puisqu'on suppose que les séquences sont au moins alignées à la première et dernière frame. Cependant, une séquence peut être une sous-partie d'une autre. Il est possible d'utiliser DTW dans ces cas en augmentant les choix de départ et de directions dans le parcours de la matrice des distances. On pourrait contraindre le premier indice p_1 à être "plus ou moins" au début de la séquence et p_l à être "plus ou moins" la fin comme dans la contrainte de Sakoe-Chiba (illustration 4.2).

La méthode DTW originale s'appuie donc sur la distance euclidienne entre les éléments de la séquence et plus précisément en comparant les composants des représentations à chaque temps t. Sachant que l'on cherche à augmenter la corrélation de séquences sémantiquement et temporellement liées, on peut se demander si les représentations utilisées et leurs composants mettent en avant cette information commune et ainsi utile à l'alignement temporel. Une solution présentée, notamment dans les travaux sur l'alignement d'actions humaines, est de construire une représentation, apprise ou non, qui est sémantiquement liée à ce qui est en commun entre les séquences. Par exemple, des squelettes sont utilisés

pour aligner des séquences d'actions humaines et ainsi supprimer les informations non-communes comme l'arrière plan, les vêtements, les proportions du sujet, etc. Bien que cette stratégie permette de réduire la proportion d'informations non-communes et inutiles à l'alignement, elle n'est pas générique à tous les alignements temporels et elle engendre potentiellement une perte d'informations communes. Zhou *et al.* [123] proposent une alternative générique et sans perte d'information commune pour gérer le poids des composants dans la représentation en combinant DTW et la CCA. La CCA permet de maximiser la corrélation entre deux variables aléatoires par une projection linéaire dans un espace commun latent. Pour intégrer la CCA à la méthode DTW, l'équation 4.1 est reformulée sous forme de produit matriciel comme ci-dessous

$$\min_{\{p_x,p_y\}\in\Psi} J_{DTW} = \|XW_x - YW_y\|_F^2 \tag{4.2}$$

où X et Y représentent toujours les séquences à aligner, $W_x = W(p_x) \in 0,1^{n_x \times l}$ et $W_y = W(p_y) \in 0,1^{n_y \times l}$ sont des matrices binaires représentant le chemin de déformation (illustration 4.1).

La proximité de cette équation à celle de la CCA favorise la combinaison de ces méthodes. La méthode de déformation temporelle canonique (CTW) ajoute donc des transformations linéaires à l'équation matricielle de DTW. CTW cherche à minimiser :

$$\min_{\{V_x,V_y\}\in\Phi,\{p_x,p_y\}\in\Psi} J_{CTW} = \|V_x^T XW_x - V_y^T YW_y\|_F^2 + \phi(V_x) + \phi(V_y) \tag{4.3}$$

où $V_x \in \mathbb{R}^{d_x \times d}$ et $V_y \in \mathbb{R}^{d_y \times d}$ représentent les transformations spatiales et W_x et W_y sont les matrices binaires de déformations temporelles. De façon similaire à la CCA, ϕ est le terme de régularisation sur V_x et V_y calculé par :

$$\phi(V) = \frac{\lambda}{1-\lambda}\|V\|_F^2 \tag{4.4}$$

De plus, les projections doivent satisfaire la contrainte d'orthogonalité et donc appartenir à l'ensemble :

$$\begin{aligned}\Phi = \{\{V_x, V_y\} | V_x^T((1-\lambda)XW_xW_x^TX^T + \lambda I)V_x = I,\\ V_y^T((1-\lambda)YW_yW_y^TY^T + \lambda I)V_y = I\}\end{aligned} \tag{4.5}$$

où $\lambda \in [0,1]$ est un poids pour équilibrer l'impact de l'erreur aux moindres carrées et la régularisation.

Notons que les transformations V_x et V_y sont effectivement spatiales, car elles s'appliquent identiquement sur chaque ligne (les représentants) de X et Y. Cela ajoute alors un mécanisme de sélection des caractéristiques pertinentes dans l'estimation d'une corrélation : CTW diminue l'impact des informations non-communes entre les séquences (le style, le sujet) et diminue la dimension des signaux. De plus,

cette transformation différente pour chaque signal permet également de comparer des signaux ayant des représentations et des dimensions différentes : CTW convient aux problèmes multi-modaux (aligner une vidéo 2D et les données de capteurs de mouvements 3D).

Comme nous l'avons vu, ces techniques combinées permettent de minimiser la distance entre les séquences en passant par une représentation des éléments, une hiérarchisation et une projection des composants dans un espace latent commun et enfin par un alignement temporel. Rajoutons que ces étapes ne sont pas ordonnées, sont dépendantes les unes des autres et ces transformations sont paramétrables et optimisables dans l'objectif de maximiser la corrélation. Ces optimisations sont appliquées notamment dans la recherche de représentations dépendantes de ce qui est commun à ces séquences et indépendantes de ce qui n'est pas commun à ces séquences. Naturellement, on voudrait appliquer cette mise en évidence de ce qui est commun à un ensemble de séquences et non plus à une paire. En effet, plus il y aura d'échantillons représentant cette information commune, mieux elle sera détectée et représentée.

Pour généraliser cet alignement à un ensemble de séquences, Zhou *et al.* se sont basés sur le calcul de corrélation canonique multi-ensemble (mCCA) [38] pour proposer la technique de déformation temporelle canonique généralisée (GCTW) [122].

Soit un ensemble de m séquences, $\{X_i\}_{i=1}^m$, GCTW cherche pour tout $X_i = [x_1^i, \cdots, x_{n_i}^i] \in \mathbb{R}^{d_i \times n_i}$, une transformation spatiale linéaire $V_i \in \mathbb{R}^{d_i \times d}$ et une transformation temporelle non-linéaire $W_i = W(p_i) \in 0, 1^{n_i \times l}$ paramétrée par $p_i \in 1 : {n_i}^l$, tel que les séquences en sortie $V_i^T X_i W_i \in \mathbb{R}^{d \times l}$ sont alignées les unes aux autres, *i.e.* GCTW minimise la somme suivante :

$$\min_{\{V_i\}_i \in \Phi, \{p_i\}_i \in \Psi} J_{GCTW} = \sum_{i=1}^{m} \sum_{j=1}^{m} \frac{1}{2} \|V_i^T X_i W_i - V_j^T X_j W_j\|_2^F + \sum_{i=1}^{m} (\phi(V_i) + \psi(p_i)) \quad (4.6)$$

avec $\psi(\cdot)$ est la fonction de régularisation de la transformation spatiale V_i, *i.e.*,

$$\phi(V_i) = \frac{m\lambda}{1-\lambda} \|V_i\|_F^2 \quad (4.7)$$

avec $\lambda \in [0, 1]$ est un paramètre pour équilibrer l'impact de l'erreur aux moindres carrées et de la régularisation. Suivant les contraintes d'orthogonalité de la mCCA, les transformations spatiales appartiennent à l'espace :

$$\Phi = \{\{V_i\}_i | \sum_{i=1}^{m} V_i^T ((1-\lambda) X_i W_i W_i^T X_i^T + \lambda I) V_i = I\} \quad (4.8)$$

La régularisation des déformations spatiales est faite à travers la fonction $\psi(\cdot)$ et l'espace Ψ pour respecter les conditions de DTW généralisées à plusieurs sé-

quences. Les auteurs de GCTW ont choisi de modéliser les déformations temporelles p par une combinaison non-négatives de déformations monotones.

$$p = \sum_{c=1}^{k} a_c q_c = Qa \tag{4.9}$$

où **a** est un vecteur de poids non-négatif et $Q = [q_1, \cdots, q_k] \in [1, n]^{l \times k}$ est la base composée des k fonctions croissantes pré-définies. Cette paramétrisation permet une projection non-linéaire monotone, car toute combinaison positive de fonctions croissantes est croissante. En ajoutant la contrainte de limite sur les indices, on cherche des coefficients a qui appartiennent à

$$\Psi = \{a | La \leq b\} \text{ avec } L = \begin{pmatrix} -I_k \\ -q^{(1)} \\ q^{(l)} \end{pmatrix} \text{ et } b = \begin{pmatrix} O_k \\ -1 \\ n \end{pmatrix} \tag{4.10}$$

Et la contrainte de continuité est assurée par

$$\psi(a) = \eta \| F_l Q a \|_2^2 \tag{4.11}$$

où $F_l \in \mathbb{R}^{l \times l}$ est le filtre différentiel du premier ordre et η est le poids du terme de régularisation temporelle.

L'illustration 4.3 présente cinq choix de bases monotones communément utilisées dans la littérature, une combinaison de ces fonctions et une déformation à partir de cette combinaison.

4.3.2 Alignement de séquences vidéo

Notre objectif est tout d'abord de réduire la variation de vitesse à l'intérieur d'une classe. Nous avons choisi GCTW parmi les extensions de DTW pour sa capacité à aligner simultanément plusieurs séquences, sa transformation à la fois spatiale et temporelle, et les transformations temporelles non-linéaires paramétrables. La projection spatiale est utile lors de l'optimisation pour placer les séquences dans un espace commun latent pour une estimation optimale de la corrélation. Donc les projections spatiales permettent à l'alignement par GCTW d'être focalisé uniquement sur les informations communes entre les séquences pour les aligner temporellement.

Cependant ces transformations peuvent supprimer des éléments non-communs mais discriminatifs entre les classes. En effet, lors de l'alignement par GCTW, on ne considère qu'une seule classe et on n'optimise pas les projections dans une optique de distances inter-classes. Par conséquent, nous choisissons d'utiliser uniquement les déformations temporelles W_i optimisées par GCTW et pas les déformations spatiales V_i (cf illustration 4.4). Comme nous avons vu dans les travaux connexes, il est nécessaire de choisir une représentation sur les images pour appliquer GCTW aux vidéos. Tout d'abord, nous avons voulu représenter les frames directement par

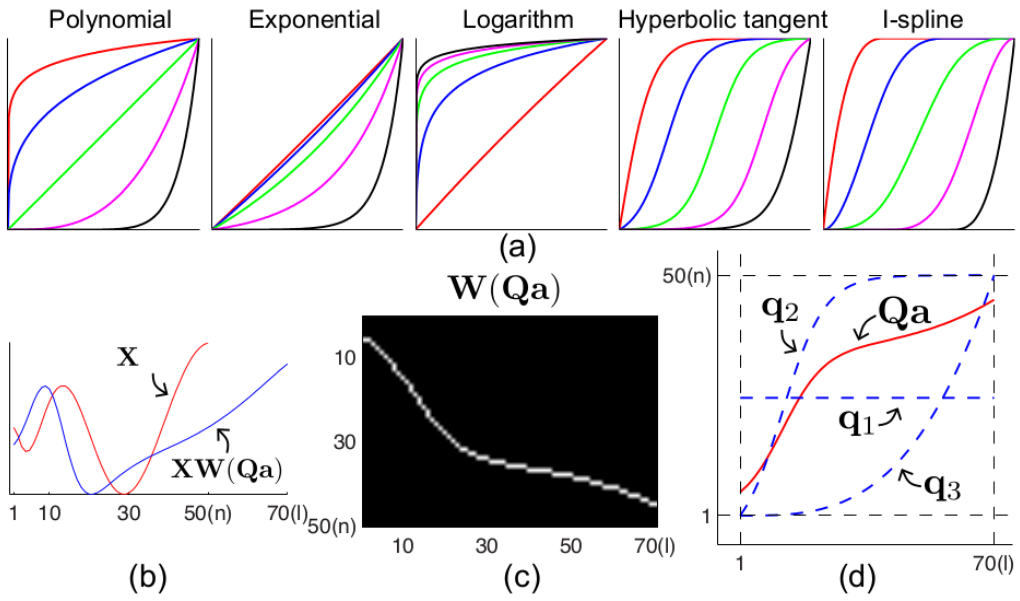


FIGURE 4.3 – (a) Cinq types de bases de fonctions monotones. (b) Une combinaison positive Qa des trois fonctions représentées : la fonction constante q_1 et deux fonctions croissantes q_2 et q_3. (c) La matrice binaire de déformation temporelle $W(Qa)$ induite par Qa. (d) Un exemple de déformation temporelle d'une séquence 1D $X \in \mathbb{R}^{1\times 50}$ en une nouvelle séquence $XW(Qa) \in \mathbb{R}^{1\times 70}$ (illustration provenant de [122]).

leurs pixels en niveaux de gris. Mais nous avons alors rencontré des problèmes de scalabité. La difficulté majeure avec les vidéos reste la redondance et alors le compromis entre la quantité d'information et la puissance de calcul nécessaire. Alors nous nous sommes inspirés du système de représentation du challenge Youtube 8M [2]. Les images ont alors été décrites par un réseau de neurones profond, GoogleNet [92], et ces représentations sont ensuite réduites en dimension en utilisant une PCA. Puis nous alignons les séquences vidéo par classe, en appliquant GCTW sur ces représentations PCA. Pour finir, nous récupérons seulement les transformations temporelles propres à chaque vidéo pour aligner nos vidéos. Ainsi, nous obtenons une base de données où chaque vidéo a la même longueur et où l'élasticité temporelle à l'intérieur d'une classe est réduite.

4.3.3 De l'alignement à la classification

En utilisant notre système d'alignement, nous pouvons aisément supposer qu'une tâche de classification vidéo peut tirer bénéfice de cette réduction de variation intra-classe. Notre système basé sur GCTW nous donne une base de données sans élasticité temporelle à l'intérieur de chaque classe. Par conséquent, nous pouvons

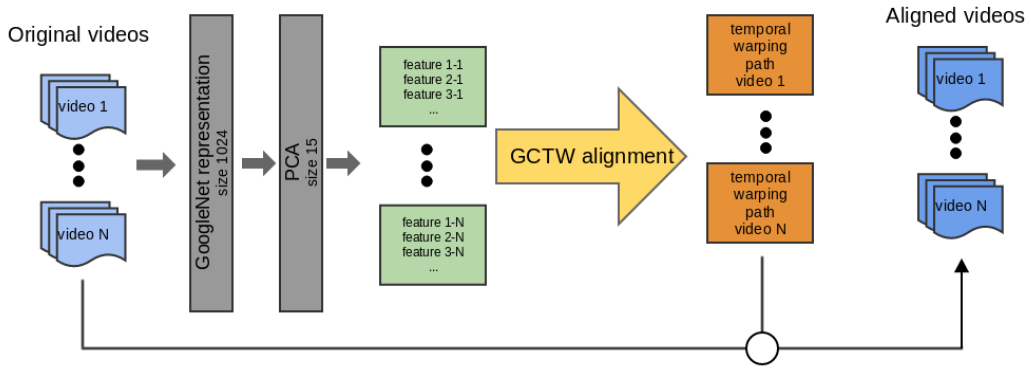


FIGURE 4.4 – Alignment Framework using GCTW

alors entraîner notre classifieur sur cette nouvelle base. Dans nos expériences, nous utiliserons le réseau de neurones C3D comme classifieur vidéo, mais tout type de classification vidéo peut être utilisé. Avec un tel entraînement, notre classifieur apprend sur des vidéos normalisées temporellement. Cependant, pendant la phase de test, le classifieur rencontrera des vidéos non-normalisées. Dans la prochaine section, nous présenterons les différents protocoles de test que nous avons envisagés pour gérer ce problème.

4.4 Nos expériences

4.4.1 Les données choisies

Comme nous utilisons une technique de réduction de variation intra-classe générale mais que nous voulons nous concentrer sur les variations intra-classes temporelles, nous avons choisi une base de données contenant peu de variations intra-classes à l'exception des variations de vitesse d'exécution : le langage des signes américains (ASL) [110].

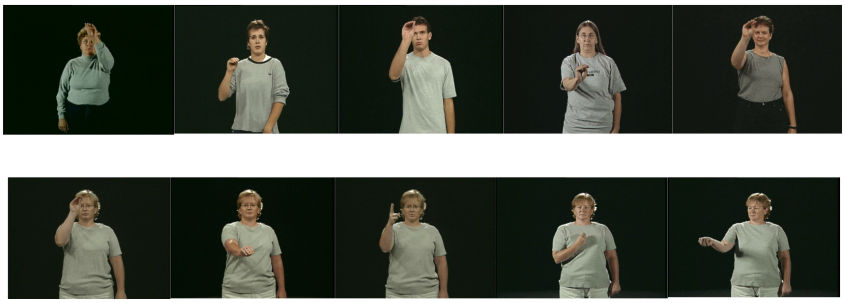


FIGURE 4.5 – Exemples d'images provenant de la base ASL. À la première ligne, plusieurs interprètes effectuent le même signe. À la seconde ligne, plusieurs signes sont effectués par le même interprète.

La base de données ASL est composée de 1204 vidéos RGB illustrant 43 signes différents (cf illustration 4.5). Bien qu'il y ait 14 interprètes différents, deux directions d'illumination et l'insertion de silences avant ou après le signe, les variations intra-classe sont effectivement faibles dans cette base. En effet, les interprètes sont habillés de la même couleur neutre, sur fond noir fixe et tous à la même distance de la caméra.

À l'inverse, la variation intra-classe temporelle est considérable dans cette base. En effet, pour analyser la variation de vitesse d'un signe, la figure 4.6 illustre la distribution des durées des vidéos par classe. En gardant à l'esprit que ces vidéos contiennent un moment de silence avant et après le signe, ce diagramme permet d'avoir un aperçu des différences dans les vitesses d' exécutions. Pour ces raisons, nous avons choisi cette base pour analyser l'élasticité temporelle comme la majeure variation intra-classe.

Nous avons également effectué des tests sur la base de données IsoGD constituées de vidéos RGB-D illustrant des gestes isolés [105]. Cette seconde base de données de gestes est construite à partir des vidéos de la base de données Chalearn LAP (*looking at people*) [105], qui est elle-même une composition de gestes de plusieurs langages : le langage pour les personnes sourdes et mal-entendantes, pour la plongée sous-marine, les signaux pour le trafic routier et les hélicoptères, les gestes italiens d'expression, les gestes symboliques et le langage corporel en général. La base IsoGD est composée de 47933 vidéos de 21 utilisateurs effectuant 249 gestes. Cette base contient plus de variations intra-classe et inter-classe et nous pourrons alors évaluer nos analyses sur cette base plus générique. Les vidéos sont tout de même sur fond fixe et les sujets sont filmés approximativement à la même échelle malgré que les décors soient différents. Quelques images d'illustration sont présentées à la figure 4.7.

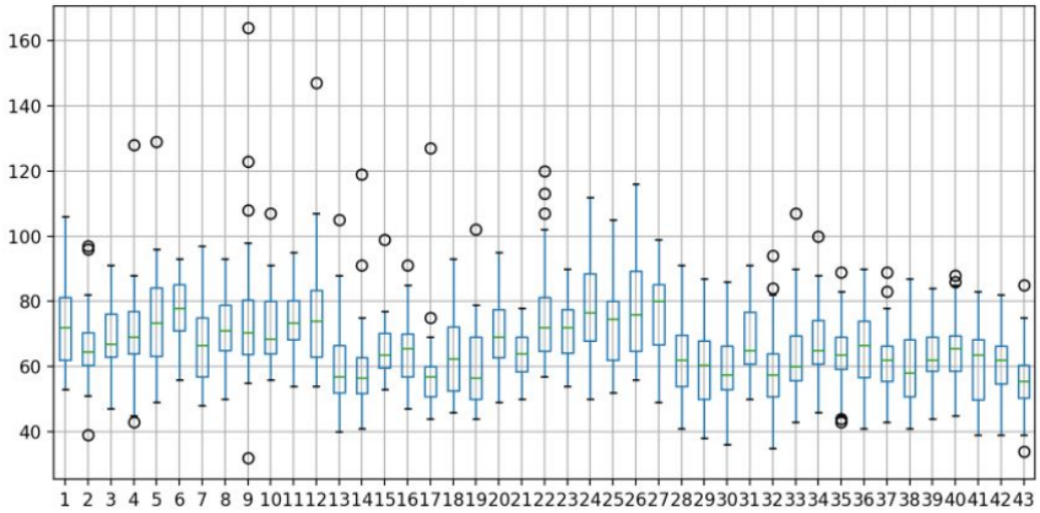


FIGURE 4.6 – Diagrammes en boîte à moustaches illustrant la distribution de la longueur des vidéos à l'intérieur de chaque classe sur la base ASL.

FIGURE 4.7 – Captures d'images de plusieurs interprètes effectuant le même signe provenant de la base IsoGD.

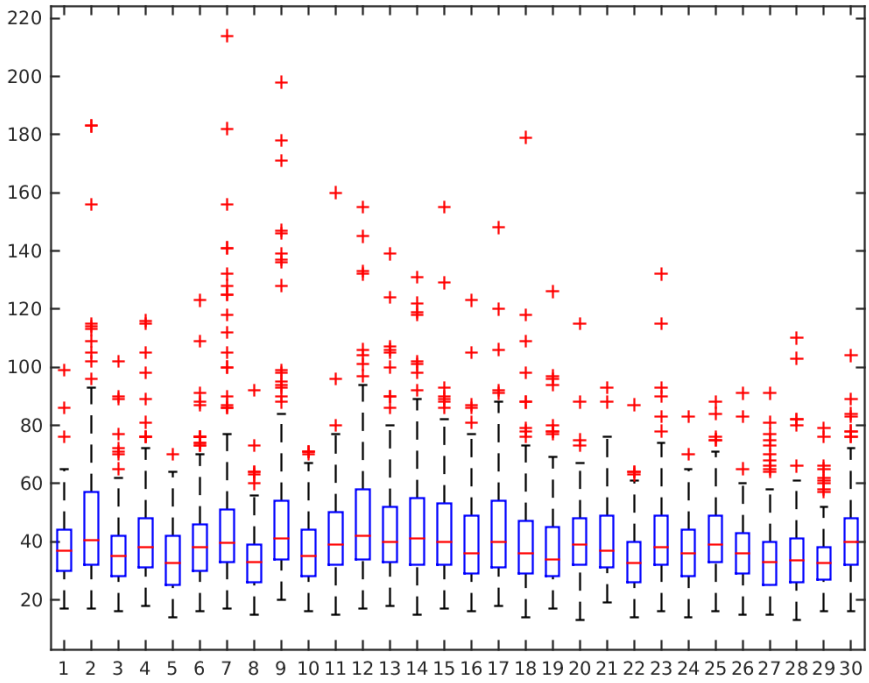


FIGURE 4.8 – Diagrammes en boîte à moustaches illustrant la distribution de la longueur des vidéos à l'intérieur des 30 premières classes pour la base IsoGD.

Dans la section suivante, nous présenterons la classification appliquée à la nouvelle base de vidéos alignées, les différents protocoles de test et ainsi l'impact de l'alignement sur la classification.

4.4.2 Alignement et Classification

Pour aligner la base de données, nous utilisons notre système décrit à la section 4.3.2. Les frames sont décrites par les descripteurs de taille 1024 de l'avant-dernière couche du réseaux profond GoogleNet. Puis nous réduisons la dimension des descriptions grâce à une PCA. Avec 10 composants, nous préservons 90 % de la variance initiale.

Suite à cet alignement, la classification des gestes sera-t-elle plus efficace ?

Le réseau de reconnaissance C3D est une méthode reconnue de l'état de l'art en classification vidéo. Grâce à sa convolution à trois dimensions, ce réseau de neurones capture les motifs spatiaux temporels discriminants de chaque classe.

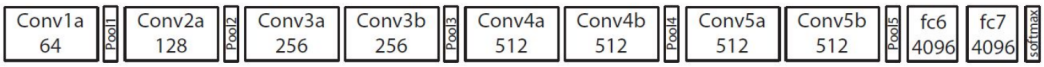


FIGURE 4.9 – Architecture du réseau C3D : 8 couches de convolutions 3D avec des noyaux de taille $3 \times 3 \times 3$, 5 couches de *max-pooling* et 2 couches *fully-connected*, suivies par une fonction de sortie de type *softmax*. Le nombre de filtres est indiqué dans chaque case (illustration provenant de [96]).

La version originale de C3D impose une vidéo de 16 frames en entrée et comme nos vidéos font en moyenne 70 frames, nous avons opté pour l'utilisation de la version de C3D incluse dans l'implémentation du réseau de neurones C3D régional [114]. Cette version du réseau permet d'être ré-entraînée pour toutes les tailles de vidéos. Cependant la séquence d'entrée reste tout de même de taille fixe et comme nous l'avons vu sur la distribution (cf figure 4.6), nos vidéos sont de longueurs très variables. Par conséquent, nous avons décidé de normaliser toutes les vidéos de sorte qu'elles soient de la même durée avant leur entrée dans C3D. Nous avons choisi une grande taille de normalisation comparée à la taille moyenne des vidéos,*i. e.* 140 images. D'un coté, nous avons étendu les vidéos plus courtes en dédoublant la dernière image jusqu'à la taille nécessaire ; d'un autre côté, nous avons réduit les vidéos plus longues en prenant les frames centrales. Ce choix de normalisation est pour nous un bon compromis entre la normalisation et la perte d'information puisque l'on enlève au maximum 0.5 s au début et à la fin de la vidéo et qu'en plus les vidéos particulièrement longues contiennent des silences avant et après l'exécution du geste. Cette stratégie permet de modifier toutes les longueurs des vidéos en préservant la distribution originale des vitesses d'exécution même si une partie de la séquence devient statique. De plus, C3D n'apprend pas de filtres sur ces zones puisque les parties statiques ne devraient pas être discriminantes. Ainsi, cet ajout de partie de vidéo statique n'introduit pas de bruit

dans la classification.

Pour des contraintes matérielles, nous choisissons de réduire le taux d'affichage des images dans la vidéo. Les vidéos d'origines sont réduites d'un taux de 30 FPS à un taux de 7.5 FPS, ainsi les vidéos de 140 images sont réduites à 35 images. Comme vous pouvez le voir sur l'illustration 4.10, cette réduction sur l'une des vidéos les plus courtes (première ligne de l'illustration) ne nuit pas à la reconnaissance du geste effectué. Par la suite, l'adjectif *normalisé* fera référence à cette normalisation sur la longueur des vidéos en conservant la distribution initiale des vitesses d'exécution. À l'inverse, le terme *aligné* fera référence à la normalisation par la longueur et la vitesse d'exécution en utilisant uniquement les déformations temporelles de GCTW. Les vidéos alignées sont de taille $l = 35$, comme celles de la base normalisée.

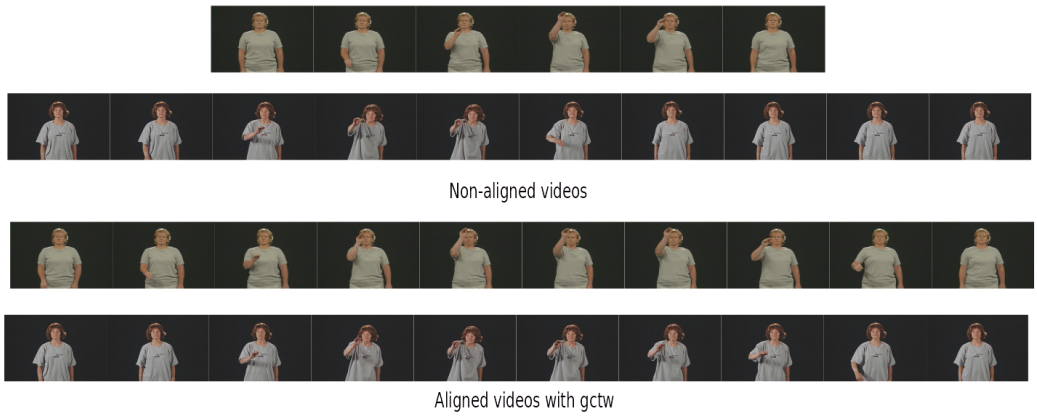


FIGURE 4.10 – Deux séquences vidéo du même signe de la base ASL : avant alignement temporelle en haut et après l'alignement temporel en bas. La première vidéo est une des plus courtes de la base ASL.

Pour chaque entraînement de C3D, que ce soit sur la base normalisée ou la base alignée, nous avons récupéré les poids de C3D entraîné sur la base de classification de vidéo de sport, Sport 1-Milion [47], constituée de 1 million de vidéos Youtube appartenant à 487 classes. L'entrée de ce réseau est constituée de données vidéos brutes dans les deux cas, c'est-à-dire des tableaux de pixels en 3 dimensions. La dernière couche de C3D est ré-entrainée dans l'objectif de la classification de nos bases de gestes.

Pour comparer une classification sur une base normalisée et sur une base alignée, nous avons effectué 4 protocoles de classification :

— **référence :** Classification par C3D sur la base normalisée

— **protocole 1 :** Apprentissage par C3D sur les données alignées et tests sur les données normalisées

— **protocole 2 :** Classification par C3D sur la base alignée

— **protocole 3 :** Apprentissage par C3D sur les données alignées et tests sur les vidéos alignées avec chaque classe

Référence : La référence représente l'utilisation classique des réseaux de neurones en vidéo avec les normalisations des tailles et l'échantillonnage uniforme sur la dimension temporelle. Les protocoles avec alignement seront comparés à cette base pour étudier l'effet de l'alignement sur l'apprentissage à travers la modification des vidéos d'apprentissage et des vidéos de tests.

Protocole 1 : Le protocole 1 permet d'étudier le comportement de C3D en test quand il est confronté à de nouvelles vitesses d'exécution. En effet, en alignant uniquement les échantillons d'apprentissage, le réseau apprend que le geste est toujours effectué à une certaine vitesse. Or, lors du test, on lui présente les vitesses originales d'exécution et donc parfois différentes de la vitesse d'exécution choisie par l'alignement.

Protocole 2 : Le protocole 2 est particulier, car nous avons décidé d'utiliser l'information de classe dans le but de se placer dans le cas hypothétique le plus simple : la réduction minimale de l'élasticité temporelle. Nous avons alors aligné toutes les vidéos selon leur classe, peu importe si elles provenaient de l'ensemble d'apprentissage ou de test.

Protocole 3 : Le protocole 3 a deux rôles. D'une part, il présente un système de classification vidéo contenant une étape d'alignement temporel intra-classe qui ne nécessite pas la connaissance de la classe en phase de test. D'autre part, il nous permet d'étudier l'alignement temporel et la classification d'une séquence vidéo alignée à une classe différente de la sienne. Le protocole 3 consiste à entraîner C3D sur les vidéos d'apprentissage alignées temporellement comme pour les deux précédents protocoles. En phase de test, la nouvelle vidéo X est alignée à chacune des classes. Notons C le nombre de classes. Nous obtenons alors C vidéos transformées par la déformation temporelle associée à la classe : XW^c avec $c \in 1, \cdots, C$. Pour chaque vidéo de test, toutes ces versions alignées sont classifiées par C3D. Nous gardons uniquement le score de classification de la classe à laquelle la vidéo originale de test a été alignée. La classe prédite est celle qui maximise le score de

Protocole de classification	ASL		IsoGD	
	top-1 acc	top-5 acc	top-1 acc	top-5 acc
Référence	76.7	96.2	45	89.05
Protocole 1	14.5	37.9	41.65	71.03
Protocole 2	91.7	98.7	81.27	97.01
Protocole 3	50.4	92	81.34	97.11

TABLE 4.1 – Taux de reconnaissance top-1 et top-5 sur les bases de données ASL et IsoGD selon les différents protocoles.

classification. Notons $\mathcal{F}_{protocole3}$ la fonction de score de ce protocole, $\mathcal{F}_{C3D}$ la fonction de score de C3D et $\mathcal{F}^c_{C3D}$ son score pour la classe c. La prédiction d'une vidéo de test dans le protocole 3 est alors :

$$\arg\max_c \mathcal{F}^c_{protocole3}(X) = \arg\max_c \mathcal{F}^c_{C3D}(XW^c) \tag{4.12}$$

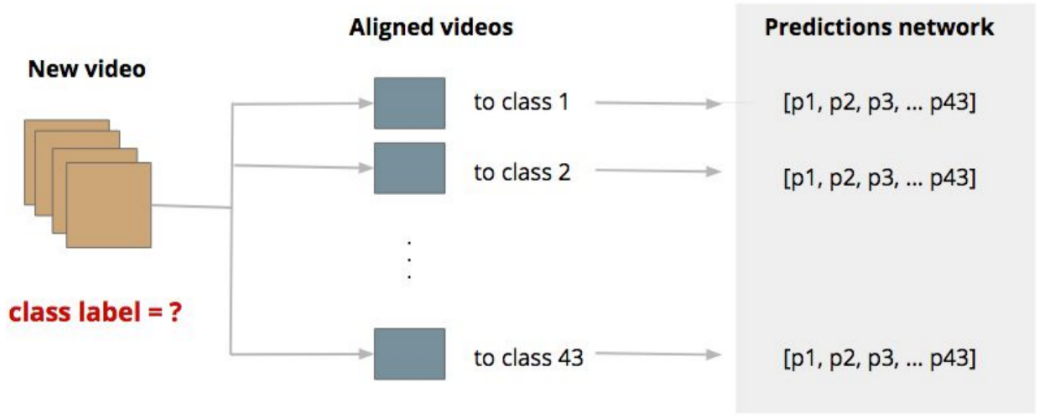


FIGURE 4.11 – Système de classification de la stratégie 3 en alignant la vidéo de test à toutes les classes.

4.4.3 Résultats

Le tableau 4.1 montre les résultats en test sur les deux bases en suivant les différents protocoles.

Tout d'abord, on remarque que le protocole 1 fait chuter la reconnaissance, comparé à la référence, que ce soit sur ASL ou sur IsoGD. Le fait que C3D ait des difficultés à reconnaître des actions exécutées à des vitesses non-observées durant l'entraînement indique que la variation temporelle entrave effectivement la classification par des réseaux de neurones convolutionels vidéo.

Deuxièmement, avec le protocole 2, on remarque que quand nous nous plaçons dans un contexte simple, où il n'y a pas d'élasticité temporelle à l'intérieur des classes, le taux de reconnaissance gagne 15% sur ASL comparé à la référence et 35% sur IsoGD. Par conséquent, si les méthodes de classification vidéo intégrait une gestion de l'élasticité temporelle dans leur représentation, ou si l'on avait une méthode de normalisation des vitesses sans nécessairement utiliser les labels des échantillons, les réseaux de neurones convolutionnels vidéo pourrait améliorer leur taux de reconnaissance de façon notable.

Pour finir, avec le protocole 3, les résultats diffèrent sur ASL et sur IsoGD. L'alignement GCTW est comme nous l'avons vu une méthode générative puisqu'elle ne considère qu'une classe à la fois durant le processus. Il ne prend pas en considération les variations inter-classes également utile à l'apprentissage et à la classification. C'est pourquoi nous avons choisi de n'utiliser que les transformations temporelles W en sortie de GCTW. Au vu du résultat sur ASL, il semble que l'alignement y a diminué la variation inter-classe. ASL étant une base de données très cadrée au niveau de la configuration de leurs vidéos (même fond, même vêtements, etc), il semble que l'alignement se soit basé sur des caractéristiques communes à la classe certes, mais qui sont également communes aux autres classes. Dans ce cas, les exécutions de deux actions qui sont à l'origine faiblement différentes visuellement (une forme de main différente mais un même geste de bras par exemple), seront effectuées à la même vitesse, et donc plus difficilement différentiable qu'avant l'alignement. Sur la figure 4.12, on peut voir tous les scores de prédictions sur toutes les vidéos alignées aux différentes classes à partir d'une seule et même vidéo de test de ASL. Cette vidéo de test est de la classe 3. Les lignes correspondent au numéro de la classe avec laquelle la vidéo est alignée, et les colonnes correspondent aux scores de prédiction de C3D pour cette classe. On remarque sur cette matrice que la vidéo est souvent classée comme étant de la classe 3, même si elle est alignée temporellement à une autre classe. On remarque également qu'elle est aussi classée dans la classe 11 et un peu dans la classe 12. On peut donc en conclure qu'un réseau C3D entraîné pour différencier ces classes par rapport à des caractéristiques spatiales (formes/couleurs) et temporelles (mouvement/vitesse), un nouvel élément qui contiendra les caractéristiques spatiales de la classe 3 mais une vitesse d'exécution choisie pour être semblable à la vitesse d'exécution de la classe 11 pourra être reconnu de la classe 11. On voit alors que la normalisation de la vitesse d'exécution dans une classe peut engendrer une prépondérance des caractéristiques de mouvements face aux caractéristiques spatiales.

Pour résumé, la prédiction d'un élément contenant les caractéristiques spatiales de sa classe et la vitesse d'exécution d'une autre classe peut autant être la vraie classe si l'influence des caractéristiques spatiales est importante, ou d'une autre classe si les caractéristiques spatiales entre ces classes sont assez proches pour que la nouvelle vitesse d'exécution influence la décision vers l'autre classe. En théorie, on voudrait qu'une vidéo contenant les caractéristiques spatiales d'une classe et les caractéristiques temporelles d'une autre ait un score de prédiction faible dans les deux classes. Cette réaction est visible sur la figure 4.12 pour la

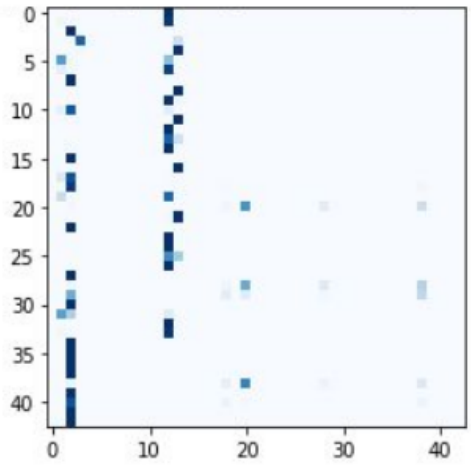


FIGURE 4.12 – Scores de classification d'une séquence de test de la classe 3, alignée à toutes les classes et classée par C3D. Chaque ligne correspond à un alignement à une classe, et chaque colonne représente la classe pour laquelle C3D donne son score.

ligne d'alignement avec la classe 5, 32 et 37 par exemple. Les réactions du classifieur face à ces nouveaux types d'échantillons est donc très variable, en particulier sur ASL.

D'ailleurs sur la base IsoGD, l'effet est plutôt inverse. Les résultats du protocole 3 sont légèrement meilleurs que ceux du protocole 2 (le cas idéal). Donc un élément aligné avec sa classe peut être confondu avec une autre classe (protocole 2) mais que lorsque cet élément est aligné avec cette même classe qui crée la confusion, la discrimination est plus forte. Rappelons que ces deux protocoles sont testés sur le même réseau C3D entraîné sur la base alignée. Soit X cette vidéo alignée d'une part avec sa classe c_1 et d'autre part avec une autre c_2, on a alors

$$\mathcal{F}^{c_2}_{C3D}(XW^{c_2}) < \mathcal{F}^{c_1}_{C3D}(XW^{c_1}) < \mathcal{F}^{c_2}_{C3D}(XW^{c_1}) \tag{4.13}$$

De cette observation, on remarque que, dans certains cas, l'alignement d'une séquence à une classe qui n'est pas la sienne peut réduire le score de prédiction des deux classes. La discrimination n'est pas sous contrôle, et la proximité originale des classes peut autant être accentuée que réduite par l'alignement. Pourtant, la vitesse d'exécution finale après alignement peut être variée et est choisie par GCTW. On peut donc contraindre un peu plus GCTW pour optimiser le choix de la vitesse d'exécution normée de sorte à augmenter la discrimination. Une autre solution serait de contraindre C3D à être plus précis et ainsi à rejeter les exemples qui ne ressemblent à aucune classe. On peut par exemple ajouter une classe "not seen" à C3D qui représenterait les échantillons alignés avec d'autres classes que la sienne.

4.5 Discussion

Les approches DTW ont deux principaux désavantages. D'une part, l'aspect génératif de DTW doit être couplé avec un critère de discrimination comme nous l'avons vu avec les résultats du dernier protocole. De plus, l'ajout d'une discrimination améliora également l'alignement à l'intérieur d'une classe. En effet, l'alignement réduit la variation intra-classe en se basant sur ce qui est commun. Or, ces éléments communs peuvent être dus à d'autres contenus que la classe que nous avons voulu représenter. Par exemple, imaginons un ensemble de vidéos de personnes qui courent dans un parc et que ces vidéos contiennent également des effets lumineux dus aux mouvements des nuages. L'algorithme initial peut aligner les vidéos en fonction de l'action humaine autant qu'en fonction des effets lumineux qui peuvent être perçus comme des éléments communs. On voit dans ce cas le manque de précision de l'hypothèse initiale : les séquences sont corrélées, mais sur quel contenu ? Or, avec la discrimination, en présentant une autre classe où ses effets lumineux sont perceptibles, le réseau se concentrera plus l'action en elle-même, qui est à la fois un élément commun et un élément discriminant. Dans DDATW [97], ils combinent DCTW et LDA pour ajouter une contrainte de discrimination à l'optimisation des déformations spatio-temporelles. Cependant, DDATW requièrt une annotation temporelle des séquences et est donc utilisée dans un contexte spécifique de prédiction. De plus, ajouter un critère de discrimination directement à l'optimisation implique un plus grand nombre de comparaison de séquences et donc augmente le coût de calcul.

D'autre part, les fonctions d'alignement, V_i^c et W_i^c, dépendent à la fois de la séquence et de la classe avec laquelle on l'aligne. Cela implique un calcul de ces fonctions propres à une nouvelle vidéo en phase de test. Une solution que nous envisageons est d'apprendre une fonction S et T qui pourrait directement prédire les fonctions de déformation d'une vidéo sans connaître sa classe. L'idée serait alors d'étendre le principe du réseau de transformations spatiales (Spatial Transformer Network) [40] à réseau de transformations temporelles. Dans cette configuration, les normalisations sont alors directement optimisées dans un objectif de classification, il y aura donc plus de contrôle sur la représentation et la discrimination engendrée par l'alignement.

4.6 Conclusion

Dans ce chapitre, nous nous sommes concentrés sur l'impact de l'alignement temporel sur la classification vidéo. Nous avons présenté un système de classification vidéo générique combinant GCTW pour l'alignement et C3D pour la classification. Notre système montre une forte amélioration sur l'état de l'art quand les vidéos de test sont alignées avec leur classe. Puisque la véritable classe est inconnue au moment du test, nous avons proposé un processus d'alignement et de prédiction non-dépendants de la vraie classe.

L'ajout d'une normalisation indépendante de la donnée et de sa classe et l'ajout d'un critère de discrimination sont des améliorations envisagées. Notre travail préliminaire sur l'extension du principe du réseau de transformation spatiale (*spatial transformer network*) [40] à un réseau de transformation temporelle semble prometteur.

CHAPITRE 5

Conclusion

Contents

5.1 Nos contributions

L'état de l'art en description et en reconnaissance vidéo utilise peu d'outils adaptés au temps. Les réseaux de neurones récurrents introduisent de la temporalité, mais ont une attention limitée dans le temps [89]. Tandis que les convolutions 3D et les réseaux 2-stream ne peuvent capturer entièrement le dynamisme d'un mouvement [96].

Tout au long de cette thèse, nous nous sommes donc intéressés à la description et la reconnaissance de contenu vidéo en se concentrant sur l'information temporelle, le mouvement et ses particularités. Nous avons organisé cette étude en trois phases :

— Une description locale et explicite : les singlets

— Une description globale et implicite : les tenseurs et les décompositions tensorielles

— Une normalisation, un alignement des mouvements : les déformations temporelles

Tout d'abord, nous nous sommes inspirés des singularités de mouvements détectées localement sur des flots optiques. Ces singularités représentent chacune un type de mouvement autour d'un point stable. Elles permettent par exemple de détecter un zoom et sa direction. En traquant ces singularités au cours du temps, nous obtenons une chaîne de mouvements : les singlets. Ces chaînes permettent de décrire une séquence vidéo et de détecter notamment des ralentis ou des moments d'agitation. Nous avons également proposé une chaîne de traitement utilisant les singlets pour résumer un match de football. Ce descripteur fait-main est explicite

et il décrit localement une vidéo. Il est générique au domaine de la vidéo, car il est basé sur le flot optique uniquement.

Par la suite, nous avons étudié une description vidéo, globale et implicite, grâce aux tenseurs multi-dimensionnels et à leurs différentes décompositions. Dans ce chapitre, deux décompositions tensorielles ont été appliquées à des vidéos : HOPLS et CIFA. Étant l'extension multi-dimensionnelle de la régression aux moindres carrées, HOPLS est une méthode linéaire permettant de relier la décomposition d'une base de vidéo et la décomposition des labels de cette base. De ces deux décompositions, nous pouvons extraire des filtres. Plus la décomposition approche le tenseur initial, plus les filtres se complexifient. Bien qu' HOPLS soit une méthode de description discriminante, elle n'atteint pas des taux de classification satisfaisants. De plus, HOPLS demande de construire un tenseur unique contenant la base de données entière. Cette méthode est donc coûteuse en mémoire. La méthode CIFA est une méthode d'extraction des représentants à partir d'une classe. À nouveau, les décompositions tensorielles approchent le tenseur initial en complexifiant l'information extraite, les représentants. Ces représentants apportent de nouveaux repères dans l'espace des vidéos, permettant ainsi une meilleure classification lors de l'utilisation d'une distance par corrélation. Cependant, une classification par vecteurs de support ne permet pas une meilleure reconnaissance. De plus, comme CIFA est une méthode de décomposition linéaire, les représentants extraits sont sensibles aux changements spatiaux et temporels. Ainsi, une classe contenant plus de variations spatiales et temporelles, obtiendra des représentants plus lisses et contenant moins de motifs prédominants.

La différence principale entre les éléments spatiaux et les éléments temporels est leur dynamique d'évolution suivant leur dimension. Si le dynamisme de la dimension temporelle (i.e. son élasticité) est réduit au minimum, alors le comportement de cette dimension sera proche des dimensions spatiales et les méthodes de description qui considèrent une vidéo comme une donnée isotrope devraient alors être plus performantes. Grâce aux dernières méthodes de déformations temporelles multi-modales et non linéaires, nous avons proposé un système d'alignement par classe. Cet alignement déforme temporellement les vidéos pour maximiser la synchronisation des vitesses d'exécutions. Puis une classification vidéo par un réseau de neurones convolutionels 3D permet de prédire l'action. Nous avons montré qu'un alignement par classe, et donc la réduction de l'élasticité, améliore effectivement la classification par C3D.

Pour conclure, les trois approches ont permis d'explorer sous différents angles la description de l'évolution temporelle en vidéo. Dans un premier temps, nous avons proposé une description faite-main ; puis, dans un second temps, nous avons étudié les décompositions tensorielles ; et nous avons déformé temporellement des séquences d'actions pour en diminuer l'élasticité. Ces descriptions et transformations ont toutes été étudiées dans un objectif de classification d'actions.

5.2 Nos perspectives

5.2.1 Ouverture pour la représentation par Singlets

Tout d'abord, nous avons remarqué que les singlets fournissent une description assez riche du mouvement. Une perspective envisagée est alors d'utiliser les singlets dans le but de reconnaître les mouvements d'un chef d'orchestre. Plus particulièrement, le but serait de différencier les intensités de la musique jouée à partir du contenu visuel, donc en fonction de l'agitation des musiciens. Un approfondissement des apports de la représentation en Singlets serait ainsi amorcé.

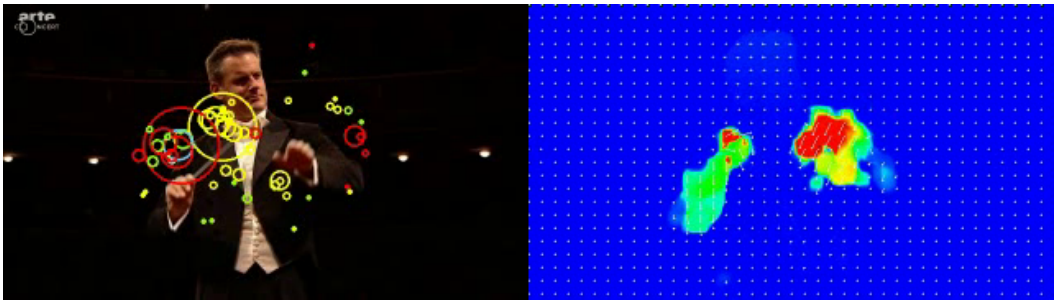


FIGURE 5.1 – Détection des singularités sur une vidéo de chef d'orchestre. Le rayon des singularités illustre leur intensité. À droite se trouve le flot optique associé à l'image et aux singularités de gauche.

5.2.2 Ouverture pour la représentation par CIFA

Plutôt que de changer d'application, la première amélioration envisagée concernant la représentation globale par CIFA est d'ajouter de la discrimination à la construction de représentants. CIFA est un système de représentation qui se concentre sur les variations intra-classes. On pourrait donc ajouter un critère de discrimination entre les classes pour améliorer la classification à partir de ces représentations. Pour rappel, l'équation d'extraction des représentants communs par CIFA est la suivante :

$$\begin{aligned} &\min_{\overline{A},\breve{A}_n} \sum_{n\in\mathcal{N}} \|Y_n - \overline{AB}_n^T - \breve{A}_n \breve{B}_n^T\|_F^2, \\ \text{tels que } &\overline{A}^T\overline{A} = I_C, \breve{A}_n^T\breve{A}_n = I_{R_n-C}, \\ &\overline{A}^T\breve{A}_n = 0, n \in \mathcal{N} \end{aligned} \tag{5.1}$$

avec $\mathcal{Y} = \{Y_n \in \mathbb{R}^{D\times J_n} : n \in \mathcal{N}\}$, $\mathcal{N} = \{1, 2, ..., N\}$ et C le nombre de variables latentes en communs.

Ici $\overline{A} \in \mathbb{R}^{D\times C}$ stocke les C représentants de taille D pour une classe. Supposons que nous ayons L classes ayant chacune C_l représentants extraits par CIFE, alors

on peut s'inspirer de la méthode HODA [15] et introduire un ratio de Fisher dans l'optimisation des représentants contraints par

$$\phi = \frac{\sum_{l=1}^{L} C_l \|\overline{A}_l - \overline{A}_*\|_2^2}{\sum_{k=1}^{\sum C_l} \|\overline{A}^{(k)} - \overline{A}_{l_k}\|_2^2} \tag{5.2}$$

où $\overline{A}_l$ et $\overline{A}_{l_k}$ sont respectivement les représentants moyens de la classe l et de la classe l_k, l_k le label du représentant k $\overline{A}^{(k)}$, et $\overline{A}_*$ la moyenne de tous les représentants. Le ratio de Fisher permet de comparer la variance entre les groupes (numérateur) et la variance à l'intérieur des groupes (dénominateur). Il permet donc d'évaluer à partir d'un ensemble de données groupées, si les sous-groupes sont clairement distincts ou s'il y a au contraire trop de confusion. L'idée est donc d'intégrer ce ratio à CIFA et reconstituer l'algorithme d'extraction COBE.

De plus, nous avions étudié les déformations temporelles pour palier à la sensibilité des décompositions tensorielles (CIFA) par rapport aux variations spatiales et temporelles. Par conséquent, il serait intéressant de tester la représentation par CIFA, sur une base de données alignées temporellement *et* spatialement.

5.2.3 Ouverture pour la normalisation temporelle

Pour finir, dans le chapitre 4, nous avons énoncé le réseau de transformations spatiales (STN) [40] comme une perspective intéressante vers une meilleure normalisation temporelle. Le réseau de transformations spatiales est en fait un réseau contenant un module peu profond servant à une transformation spatiale. Ce module, appelé module spatio-temporel (ST), est composé deux parties : une permettant de localiser des données dans l'entrée et de prédire la transformation , et une permettant d'effectuer cette transformation spatiale sur l'entrée.

La partie de localisation du module peut être composée de couches convolutionelles ou *fully-connected* mais doit se terminer par une couche de régression pour pouvoir prédire au mieux les valeurs de la transformations θ. C'est à cette étape que les informations communes à la classe et nécessaires à l'alignement seront sélectionnées.

La deuxième partie du module concerne la transformation de la donnée par les paramètres de déformation θ prédits.

Ce module est appliqué aux cubes de sorties de la couche précédente. Cette partie du réseau peut donc être placée autant au début du réseau directement sur les images qu'après quelques couches de neurones sur les cartes de descriptions (*feature maps*).

L'idée est alors de s'inspirer de ce réseau pour construire un réseau de normalisation temporelle.
Dans GCTW, la transformation spatiale V_i et la transformation temporelle W_i sont uniques par paire (*video*, *class*). Cette unicité implique une recherche de déforma-

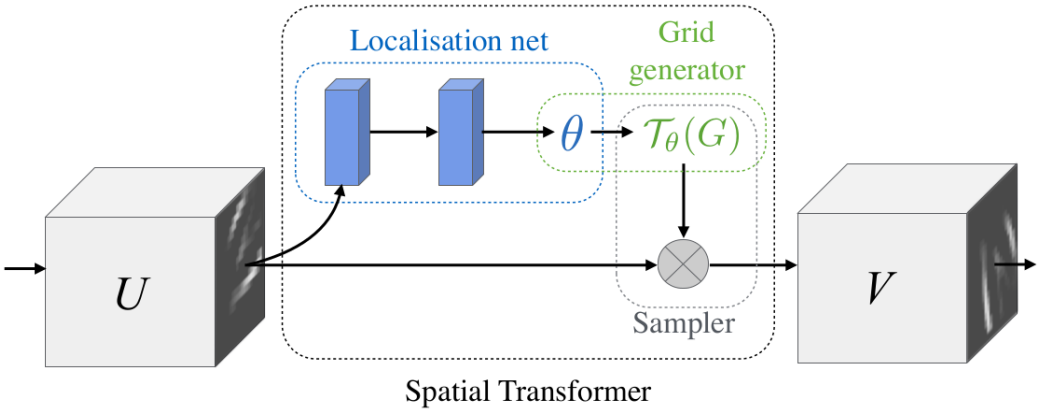


FIGURE 5.2 – Architecture d'un module ST (illustration provenant de [40]).

tion en phase de test qui est contraignante et gourmande en temps de calcul. La transformation V pourra directement être remplacée par le module ST car, dans ce module, la transformation spatiale est déjà affine, comme pour V.
Pour l'adaptation de ce module à une transformation temporelle W, à la place de θ, on peut prédire directement le chemin de déformation p, qui sera alors des indices, ou les paramètres a qui sont les coefficients de la combinaison linéaire permettant de paramétrer la base des déformations dans GCTW.
Avec l'inspiration du réseau TSN qui effectue un échantillonnage uniforme de la vidéo, on peut ajouter à ce module temporel une contrainte sur le nombre d'images restantes après la déformation temporelle de sorte à effectuer certes un échantillonnage qui diminue la redondance et donc le temps de calcul, mais surtout un échantillonnage adaptatif qui permettra également de gérer l'élasticité temporelle.

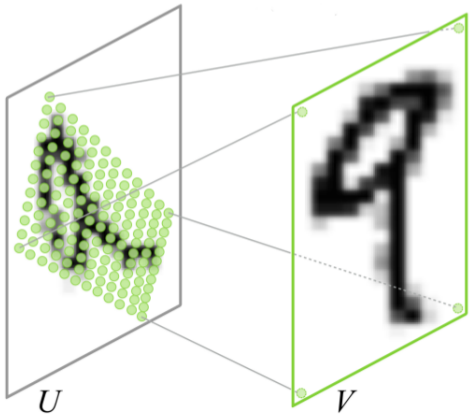


FIGURE 5.3 – Illustration de la déformation spatiale par le module ST : une grille est d'abord construite à partir des paramètres θ prédits par le réseau de localisation, puis cette grille sert de repère pour copier les pixels de l'entrée à la place désirée dans la sortie selon θ (illustration provenant de [40]).

CHAPITRE 6

Annexe

Contents

6.1 Tenseurs : les bases

Un tenseur est un tableau multi-dimensionnel à plusieurs entrées. L'*ordre* d'un tenseur désigne le nombre d'entrées que possèdent ce tableau et chaque ordre va jusqu'à une certaine *dimension*. Un tenseur d'ordre N a N indices et donc N dimensions. Ainsi un tenseur d'ordre 0 n'a pas d'indice, c'est un scalaire ; un tenseur d'ordre 1 est un vecteur, il a une dimension ; un tenseur d'ordre 2 possède 2 dimensions ; et ainsi de suite... Par cette définition, le tenseur est une représentation générale d'objets algébriques complexes. Notons que tout comme une matrice peut être vue comme une application linéaire ou comme une forme bilinéaire, un tenseur peut être vu comme une application ou une forme multi-linéaire.

6.1.1 Vocabulaire général

Nous allons introduire le vocabulaire et les techniques simples associés aux tenseurs pour la clarté de la suite du chapitre.

Un tenseur peut être vu comme composé de *tranches*, des tenseurs d'ordre 2, ou de *fibres*, des tenseurs d'ordre 1 (cf illu. 6.1), en suivant une ou deux dimensions, appelés également les *modes* du tenseur.

La *vectorisation* d'un tenseur consiste à réassembler tous ses éléments pour en faire un vecteur. Comme un tenseur peut être vu comme un ensemble de fibres, alors la concaténation de ces fibres forment une vectorisation du tenseur en suivant l'un de ses modes. Par exemple, considérons un tenseur $\mathcal{T} \in \mathbb{R}^{3\times4\times5}$. Ce tenseur est donc d'ordre 3 et a pour dimensions 3, 4 et 5. Une de ses vectorisations est donc un vecteur $t \in \mathbb{R}^{60}$.

De même, la *matricisation* de mode i d'un tenseur, appelé également son i-mode, consiste à concaténer ces tranches selon un mode pour en faire une matrice.

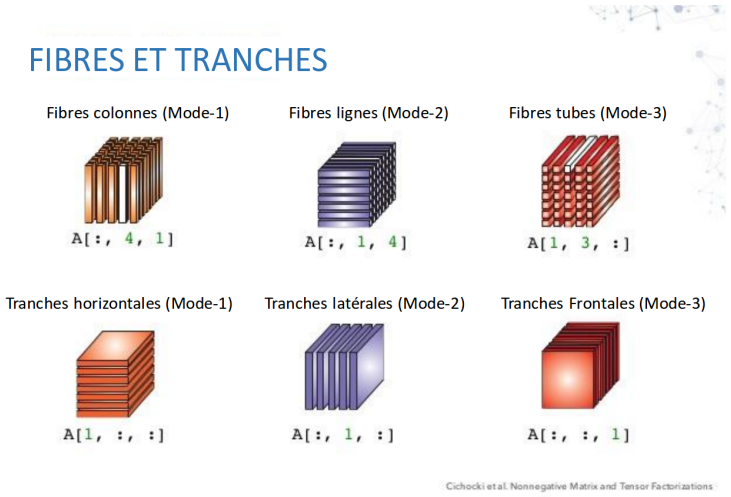


FIGURE 6.1 – Illustration des fibres et des tranches dans les tenseurs (illustration provenant de [15]).

Ainsi, en reprenant mon tenseur d'ordre 3 précédent, son 1-mode est une matrice $T_{(1)} \in \mathbb{R}^{3\times 20}$; son 2-mode est $T_{(2)} \in \mathbb{R}^{4\times 15}$ et son 3-mode est $T_{(3)} \in \mathbb{R}^{5\times 12}$ (cf illu. 6.4).

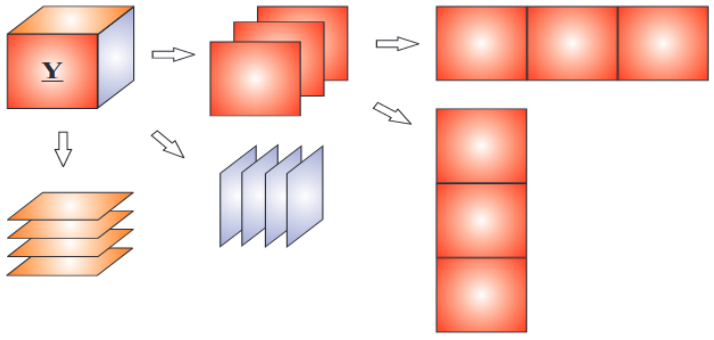


FIGURE 6.2 – Illustration des matricisations possibles d'un tenseur d'ordre 3 (illustration provenant de [15]).

La vectorisation et la matricisation ont d'abord été utilisées pour visualiser et analyser les tenseurs par les outils matriciels et vectoriels connus ; elles sont également utilisées pour re-définir les opérations sur tenseurs.

Un tenseur est *super-diagonal* si $\mathcal{X}_{i_1 i_2 \ldots i_N} \neq 0 \Leftrightarrow i_1 = \cdots = i_N$.

Un tenseur est tout orthogonal si tous ces sous tenseurs sont tout-orthogonaux si toutes les tranches de ce tenseurs sont orthogonales par rapport au produit scalaire des matrices selon toutes les dimensions. Pour un tenseur $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times I_3}$

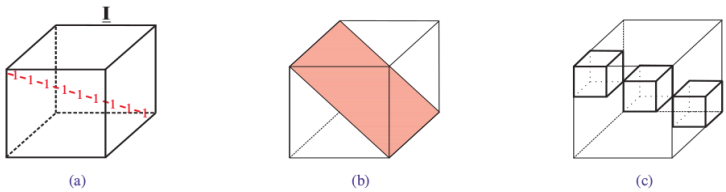


FIGURE 6.3 – Formes spéciales de tenseur : (a) un tenseur super diagonal dont la diagonale est constitué de 1, soit un tenseur super-identité, (b) un tenseur éparse avec les tranches frontales diagonales, (c) un tenseur diagonal par bloc (illustration provenant de [15]).

d'ordre 3, cela a vérifier l'équation suivante :

$$\sum_{i_1,i_2} \mathcal{X}_{i_1,i_2,\alpha}.\mathcal{X}_{i_1,i_2,\beta} = \sum_{i_2,i_3} \mathcal{X}_{\alpha,i_2,i_3}.\mathcal{X}_{\beta,i_2,i_3} = \sum_{i_1,i_3} \mathcal{X}_{i_1,\alpha,i_3}.\mathcal{X}_{i_1,\beta,i_3} = 0\ ,\ \forall \alpha \neq \beta \quad (6.1)$$

Ainsi les tranches horizontales, selon le premier mode, sont mutuellement orthogonales, et dans le même temps les tranches frontales et les tranches verticales sont également mutuellement orthogonales.

Un tenseur est dit ordonné si toutes ses tranches selon tous ses modes sont ordonnées par ordre décroissant selon leur norme de Frobenius.

Un tenseur $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ est dit de *rang 1* s'il peut s'écrire comme le produit externe de N vecteurs.

$$\begin{aligned} \mathcal{X} &= a^{(1)} \circ \ldots \circ a^{(N)} \\ \Leftrightarrow \mathcal{X}_{i_1 i_2 \ldots i_N} &= a^{(1)}_{i_1} \cdot \ldots \cdot a^{(N)}_{i_N}, \forall 1 \leq i_n \leq I_n, \forall 1 \leq n \leq N \end{aligned} \quad (6.2)$$

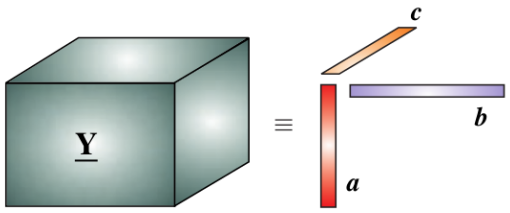


FIGURE 6.4 – Illustration d'un tenseur de rang 1 (illustration provenant de [15]).

6.1.2 Opérations tensorielles de base

Le *produit scalaire* de deux tenseurs du même ordre N et de mêmes dimensions est défini par :

$$< \mathcal{X}, \mathcal{Y} >= \sum_{i_1,i_2,\ldots,i_N} \mathcal{X}_{i_1 i_2 \ldots i_N}, \mathcal{Y}_{i_1 i_2 \ldots i_N} \quad (6.3)$$

Le produit scalaire permet de définir la *norme de Frobenius* d'ordre supérieur d'un tenseur $\mathcal{T}$ comme

$$||\mathcal{X}||_F = \sqrt{\sum_{i_1,i_2,\ldots,i_N} \mathcal{X}^2_{i_1 i_2 \ldots i_N}} \tag{6.4}$$

alors que la *norme* l_1 d'un tenseur est

$$||\mathcal{X}||_1 = \sum_{i_1,i_2,\ldots,i_N} |\mathcal{X}_{i_1 i_2 \ldots i_N}| \tag{6.5}$$

Le *mode-n produit* d'un tenseur $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ avec une matrice $U \in \mathbb{R}^{J \times I_n}$ est un tenseur $\in \mathbb{R}^{I_1 \times \ldots I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N}$ définit par

$$(\mathcal{X} \times_n U)_{i_1 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_N} \mathcal{X}_{i_1 \cdots i_n \cdots i_N} . U_{j i_n} \tag{6.6}$$

ce qui est équivalent, en utilisant les matricisations, à

$$\mathcal{Y} = \mathcal{X} \times_n U \Leftrightarrow \mathcal{Y}_{(n)} = U . \mathcal{X}_{(n)} \tag{6.7}$$

Dans une série de multiplication pour des modes distinct, le produit est commutatif.

$$\mathcal{X} \times_m A \times_n B = \mathcal{X} \times_n B \times_m A, sim \neq n \tag{6.8}$$

Dans le cas de produit du même mode, on a $\mathcal{X} \times_n A \times_n B = \mathcal{X} \times_n (BA)$

Le n-mode produit entre un tenseur $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ et un vecteur $v \in \mathbb{R}^{I_n}$ se note $X \bullet_n v$ et diminue de 1 l'ordre du tenseur initial. Dans ce cas particulier, le produit n'est pas commutatif car l'ordre du tenseur change.

Les produits matriciels présentées ci-dessous sont tout aussi utiles dans l'expression des décompositions tensorielles.

Le produit de Kronecker entre deux matrices $A \in \mathbb{R}^{I \times J}$ et $B \in \mathbb{R}^{K \times L}$ est une matrice de $\mathbb{R}^{IK \times JL}$ définit par

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B \cdots a_{1J}B \\ a_{21}B & a_{22}B \cdots a_{2J}B \\ \vdots & \\ a_{I1}B & a_{I2}B \cdots a_{IJ}B \end{pmatrix} = [a_1 \otimes b_1 \;\; a_1 \otimes b_2 \cdots a_1 \otimes b_L \;\; a_2 \otimes b_1 \cdots a_J \otimes b_L] \tag{6.9}$$

Le produit de Khatri-Rao entre deux matrices $A \in \mathbb{R}^{I \times K}$ et $B \in \mathbb{R}^{J \times K}$ est une matrice de $\mathbb{R}^{IJ \times K}$ définit par

$$A \odot B = [a_1 \otimes b_1 \;\; a_2 \otimes b_2 \cdots a_K \otimes b_K] \tag{6.10}$$

Ces deux produits possèdent les propriétés suivantes :

$$\begin{aligned}(A \otimes B)(C \otimes D) &= AC \otimes BD \\ (A \otimes B)^* &= A^* \otimes B^* \\ A \odot B \odot C &= (A \odot B) \odot C = A \odot (B \odot C)\end{aligned} \tag{6.11}$$

L'équivalence suivante entre la décomposition tensorielle et la décomposition de ses i-modes est particulièrement utile dans la simplification et la compréhension des décompositions. Soit $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ et $A \in \mathbb{R}^{J_n \times I_n}, \forall n \in \{1 \dots N\}$, alors ces deux expressions sont équivalentes :

$$\begin{aligned}&\mathcal{Y} = \mathcal{X} \times_1 A^{(1)} \times_2 A^{(2)} \times_3 \cdots \times_N A^{(N)} \\ \Leftrightarrow\ &\mathcal{Y}_{(n)} = A^{(n)} \mathcal{X}_{(n)} (A^{(1)} \otimes \cdots \otimes A^{(n-1)} \otimes A^{(n+1)} \otimes \cdots \otimes A^{(N)})^T, \forall n \in \{1 \dots N\}\end{aligned} \tag{6.12}$$

6.1.3 Les décompositions tensorielles

Les outils standards de décomposition matriciels, comme la SVD ou la ICA et leur variantes, sont des outils inestimables pour la sélection de caractéristiques, la réduction de dimension ou de bruit et l'exploration de données. Cependant les données ne peuvent alors comporter que deux dimensions ce qui contraint fortement leur représentation. Dans beaucoup d'applications, les données ont naturellement une structure multi-dimensionnelle. En effet, cette thèse s'intéresse à l'analyse de la vidéo, qui possède naturellement 3 dimensions. De même, un ensemble d'expérience se représente facilement donnée multi-dimensionnelle par le sujet de l'expérience, le numéro d'expérience qu'il a passé, les conditions de la tache à effectuer, en ajoutant également le temps et les valeurs des résultats fournis. Ainsi cette "vue plate du monde" est rapidement insuffisante et il est naturel de se diriger vers les tenseurs. Alors toutes les dimensions sont retenues grâce aux modèles multi-linéaires qui produisent souvent des éléments uniques et contenant des informations judicieuses. Utiliser les outils matricielles, tranche par tranche, dans le contexte d'une analyse conjointe mène forcément à perdre les informations d'interactions, de covariance. Pour découvrir les composantes cachés à l'intérieur de la donnée et retenir l'information pertinente, les outils d'analyse doivent refléter la structure multi-dimensionnelle de la donnée.
Décomposer un tenseur permet de l'écrire sous la forme de produit de composantes plus petit en ordre ou en dimension ou plus simple dans leur composition (les tenseurs super-diagonaux par exemple). Alors on peut considérer son rang comme un indicateur de complexité. Pour rappel, un tenseur de rang 1 s'écrit comme un produit de vecteurs.
Nous en venons donc naturellement à la première décomposition en décomposant notre tenseur en somme de tenseurs de rang 1. Cette décomposition est la *décomposition canonique polyadique* (CPD), également appelée CANDECOMP ou PARAFAC

(Parallel Factor Analysis). Soit $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, sa décomposition canonique polyadique est définit par :

$$\begin{aligned} \mathcal{X} &= \sum_{r=1}^{R} \lambda_r \circ a_r^{(1)} \circ \ldots \circ a_r^{(N)} \\ &= [\![\lambda; A^{(1)}, \ldots, A^{(N)}]\!] \\ &= [\![\mathcal{D}; A^{(1)}, \ldots, A^{(N)}]\!] \end{aligned} \tag{6.13}$$

avec $\mathcal{D} = diag_N(\lambda)$ un tenseur super-diagonal d'ordre N.

La décomposition CP peut s'exprimer de façon matricielle grâce aux i-modes :

$$\begin{aligned} \mathcal{X}_{(n)} &= A^{(n)} \mathcal{D}_{(n)} (A^{(N)} \odot \cdots \odot A^{(n+1)} \odot A^{(n-1)} \odot \cdots \odot A^{(1)})^T, \forall n \\ vec(\mathcal{X}) &= [A^{(N)} \odot \cdots \odot A^{(1)}]\lambda \end{aligned} \tag{6.14}$$

Cette décomposition n'est unique que sous certaines conditions. Le *rang* d'un tenseur, $rank(\mathcal{X})$, est le plus petit nombre de tenseur d'ordre 1 dont $\mathcal{X}$ est la somme, c'est donc le plus petit nombre de produits, R, dans une décomposition CP. Pour les tenseurs d'ordre 3 $\in \mathbb{R}^{I \times J \times K}$, il est prouvé que $rank(\mathcal{X}) < min(IJ, JK, IK)$.

La détermination du rang d'un tenseur est un problème NP-hard et nous n'allons pas nous atteler à cette tache par la suite. Tout de même, il reste intéressant de garder en tête ce critère de complexité d'un tenseur.

La deuxième décomposition la plus utilisée est la *décomposition de Tucker*. Elle est définit comme suit pour le même tenseur que précédemment :

$$\begin{aligned} \mathcal{X} &= \mathcal{G} \times_1 A^{(1)} \times_2 A^{(2)} \times_3 \cdots \times_N A^{(N)} \\ &= \sum_{i_1, i_2, \ldots, i_N} g_{i_1 i_2 \ldots i_N} . a_{i_1}^{(1)} \circ \ldots \circ a_{i_N}^{(N)} \\ &= [\![\mathcal{G}; A^{(1)}, \ldots, A^{(N)}]\!] \end{aligned} \tag{6.15}$$

$A^{(i)}$ sont les matrices facteurs et $\mathcal{G}$ le noyau décrivant les niveaux d'interactions entre les différentes composantes. Si les dimensions de $\mathcal{G}$ sont plus petits que celles de $\mathcal{X}$, $\mathcal{G}$ peut être vu comme une version compressée de $\mathcal{X}$ car moins d'éléments permettent de le construire. La version matricielle de la décomposition de Tucker écrite grâce aux i-modes du tenseur est exprimée comme suit :

$$\mathcal{X}_{(n)} = A^{(n)} \mathcal{G}_{(n)} (A^{(N)} \otimes \cdots \otimes A^{(n+1)} \otimes A^{(n-1)} \otimes \cdots \otimes A^{(1)})^T, \forall n \tag{6.16}$$

Deux variantes de cette décomposition pour les tenseurs d'ordre 3 est la décomposition Tucker 2 qui consiste à imposer une des matrices facteurs à être l'iden-

tité, et la décomposition Tucker 1 qui imposent deux matrices facteurs à être l'identité.

$$\begin{aligned} \mathcal{X} = \mathcal{G} \times_1 A = [\![\mathcal{G}; A, I, I]\!] \\ \Leftrightarrow \mathcal{X}_{(1)} = A\mathcal{G}_{(1)} \\ \Leftrightarrow PCA\ de\ \mathcal{X}_{(1)} \end{aligned} \tag{6.17}$$

Les dimensions minimales du noyau G donne le rang multi-linéaire, aussi appelé le rang multiplex ou encore l'ensemble des n-rangs, noté $rank_n(\mathcal{X})$.

Il existe différents algorithmes pour calculer ces décompositions. La plus utilisée étant certainement la méthode HOOI (High-Order Orthogonal Itération) qui suppose que les matrices facteurs sont orthogonales ce qui ramène la décomposition à une simple décomposition en valeurs propres (SVD). C'est une solution simple mais pas optimale. L'algorithme de Newton-Grassman permet de construire la décomposition pour les tenseurs d'ordre 3. La méthode CONCORDIA sert à déterminer le rang idéal, que ce soit le rang original de lLes Tenseursa CPD ou le rang multiplex. La décomposition de Tucker n'est également pas unique mais les sous-espaces engendrés par les matrices facteurs sont uniques.

D'autres décompositions sont construites en ajoutant des contraintes sur les éléments de la décomposition CP ou de Tucker ou en supposant une caractéristique particulière au tenseur décomposé. Parmi ces décomposition, on peut lister les décompositions INDSCAL, PARAFAC2, CANDELINC ou DEDICOM. Certaines de ces contraintes permettent de coller à la réalité ou de faciliter la convergence comme la factorisation non-négative de tenseur à partir de la CPD ou la décomposition de Tucker (NNCP ou NNT). Cela fait déjà une longue liste de décomposition à partir de la CPD et la décomposition de Tucker. Nous allons maintenant détailler les quatre décompositions qui nous intéressent dans l'analyse vidéo.

La *décomposition en termes bloc* (BTD) est une composition de la décomposition CP ou de Tucker : le tenseur est décomposition en somme de tenseur de rang multiplex inférieur (illu. 6.5).

$$\mathcal{X} = \sum_{r=1}^{R} [\![\mathcal{G}_r; A^{(1)}, \ldots, A^{(N)}]\!] \tag{6.18}$$

Cette décomposition est plus contraignante que la CPD mais elle respecte plus les espaces comme la décomposition de Tucker. Elle est généralement utilisée dans la séparation de sources aveugles (BSS).

La décomposition en train de tenseur permet d'écrire un tenseur comme une multiplication de tenseur d'ordre inférieur. Soit $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, alors sa décomposition en train de tenseur s'écrit :

$$\mathcal{X}_{i_1,\cdots,i_N} = \sum_{r_1=1}^{R_1} \cdots \sum_{r_{N-1}=1}^{R_{N-1}} \mathcal{G}^{(1)}_{i_1,r_1} . \mathcal{G}^{(2)}_{r_1,i_2,r_2} . \cdots . \mathcal{G}^{(N-1)}_{r_{N-2},i_{N-1},r_{N-1}} . \mathcal{G}^{(N)}_{r_{N-1},i_N} \tag{6.19}$$

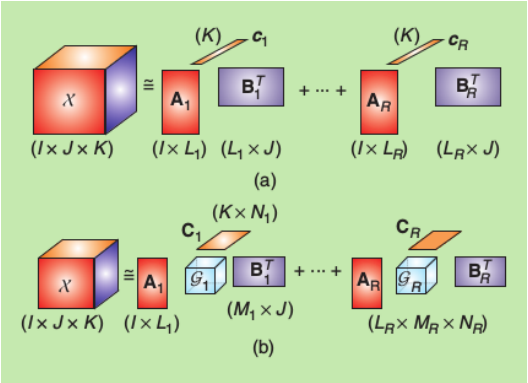


FIGURE 6.5 – La BTD trouve des composantes qui sont structurellement plus complexe que les termes de rang 1 dans la CPD. (a) Décomposition en terme de rang multi-linéaire $(L_r, L_r, 1)$. (b) Décomposition en termes de rang multi-linéaire (L_r, M_r, N_r) (illustration provenant de [14]).

avec les matrices $\mathcal{G}^{(i)} \in \mathbb{R}^{R_{i-1} \times I_i \times R_i} \forall 0 \leq i \leq N$ et R_i des dimensions supplémentaires choisies.

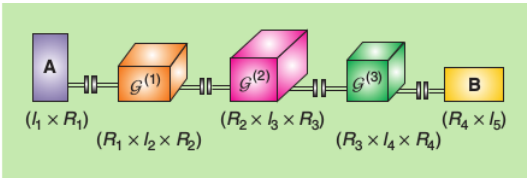


FIGURE 6.6 – La TTD d'un tenseur d'ordre 5 $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_5}$ composée de deux matrices wagons et de trois tenseurs wagons d'ordre 3. Les cinq wagons sont connectés par contractions tensorielles et l'on a $\mathcal{X}_{i_1,i_2,i_3,i_4,i_5} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_5=1}^{R_5} a_{i_1,r_1}.\mathcal{G}_{r_1,i_2,r_2}.\mathcal{G}_{r_2,i_3,r_3}.\mathcal{G}_{r_3,i_4,r_5}.B_{r_4,i_5}$(illustration provenant de [14]).

La prochaine décomposition est inspirée de la régression linéaire généralisée aux ordres supérieurs : *la méthode d'approximation partielle des moindres carrés à l'ordre supérieur* (ou High order partial least square HOPLS). Elle a été introduite par Zhao *et al* [121] comme une méthode généralisée de régression multi-linéaire. L'objectif est de mettre en relation la décomposition de deux tenseurs ayant une dimension commune ; les tenseurs sont alors exprimés dans un sous espace optimisant leur propre approximation en maximisant la covariance des éléments de décomposition. Cette technique est directement inspirée de la méthode d'approximation partielle des moindres carrées pour les relations entre matrices (illu. 6.7).

Soit $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ et $\mathcal{Y} \in \mathbb{R}^{J_1 \times \cdots \times J_M}$ deux tenseurs d'ordres différents ayant le même premier mode, donc $I_1 = J_1$. Alors $\mathcal{X}$ et $\mathcal{Y}$ sont décomposés par BTD tels

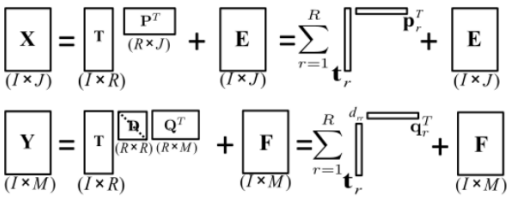


FIGURE 6.7 – Le modèle PLS : décomposition de données comme la somme de matrices de rang 1 (illustration provenant de [121]).

que

$$\begin{aligned} \mathcal{X} &= \sum_{r=1}^{R} \mathcal{G}_r \times_1 t_r \times_2 P_{r(1)} \times_3 \cdots \times_N P_r^{(N-1)} + \mathcal{E}_R \\ \mathcal{Y} &= \sum_{r=1}^{R} \mathcal{D}_r \times_1 t_r \times_2 Q_{r(1)} \times_3 \cdots \times_N Q_r^{(N-1)} + \mathcal{E}_R \end{aligned} \tag{6.20}$$

avec R étant le nombre de vecteurs latents, $t_r \in \mathbb{R}_1^I$ le r^{ieme} vecteur latent, $P_r^{(n)} \in \mathbb{R}^{I_{n+1} \times L_{n+1}}$ et $Q_r^{(m)} \in \mathbb{R}^{J_{m+1} \times K_{m+1}}$ sont les matrices facteurs de mode n et m respectivement, et $\mathcal{G}_r \in \mathbb{R}^{1 \times L_1 \times \cdots \times L_N}$ et $\mathcal{D}_r \in \mathbb{R}^{1 \times K_1 \times \cdots \times K_M}$ sont les tenseurs noyau.

Dans le cas où l'on a deux représentations d'une même donnée, il est intéressant de les organiser sous forme de deux tenseurs ayant donc une dimension en commun, le numéro de l'échantillon, et alors chercher à exprimer les relations entre les deux tenseurs et possiblement prédire une des représentations à partir de l'autre. Par exemple, si l'on voulait prédire des mouvements corporels à partir de données d'électroencéphalographies (EEG), on commencerait par étudier les relations entre les données enregistrées par capteurs EEG et par capteurs spatiaux 3D à partir d'enregistrements sur un ensemble de sujets. De plus, beaucoup de problèmes de décomposition classiques peuvent être reformulés comme des problèmes de régression et régler grâce à cette décomposition : CCA, LDA...

Enfin la décomposition la plus complexe que l'on introduira est *l'analyse en composantes multi-ordre liées* (LMWCA). Elle possède d'autres noms comme l'extraction d'une base commune orthogonale (common orthogonal basis extraction COBE) ou l'analyse ou l'extraction en éléments commun et individuels (commun and individual feature extraction or analysis CIFE ou CIFA). Cette méthode a été créée dans le but d'exploiter la nature liée des données et de leurs dimensions. Dans cette configuration, on considère le tenseur comme une variable aléatoire dont nous avons plusieurs échantillons.

$$\mathcal{X}^{(k)} = \mathcal{G}^{(k)} \times_1 B^{(1,k)} \times_2 \cdots \times_N B^{(N,k)} \tag{6.21}$$

Les matrices facteurs contiennent les éléments communs et individuels et le noyau contient les interactions mutuelles entre ces facteurs.

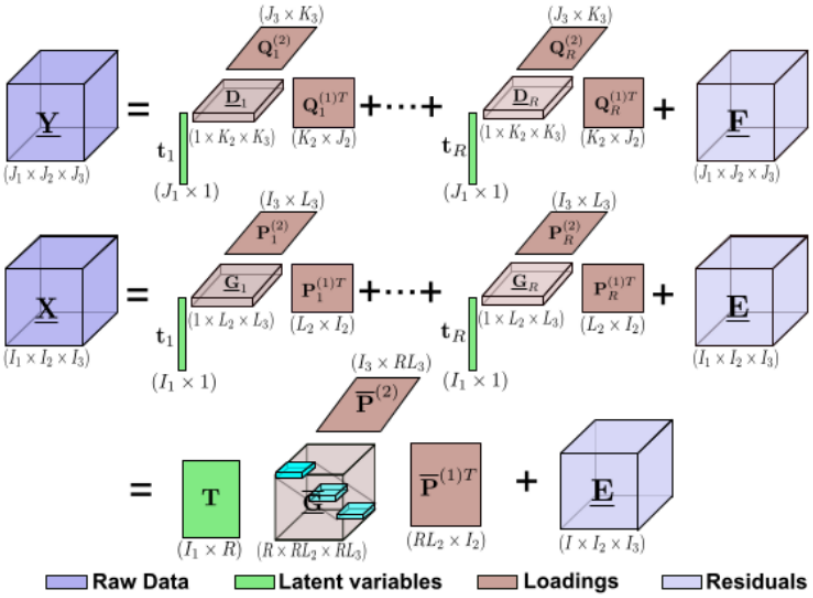


FIGURE 6.8 – Diagramme schématique de la méthode HOPLS : l'approximation d'un tenseur $\mathcal{X}$ par une somme de tenseurs de rang $(1, L_2, L_3)$ et l'approximation de $\mathcal{Y}$ suivant le même principe avec les composantes latentes communes T (illustration provenant de [121]).

Comme on peut le voir la littérature dans le domaine des décompositions tensorielles est vaste. Nous avons présenté une liste non-exhaustive des caractéristiques, des opérations et des décompositions tensorielles en commentant brièvement comment les obtenir, ce qu'elles impliquent et dans quels problèmes elles sont généralement utilisées. Une des nouvelles applications des tenseurs est le domaine de la vidéo. Pratique de par sa forme multi-dimensionnelle, le tenseur permet de stocker et de conserver le maximum d'information d'une vidéo ou d'une base de données. Le tenseur a donc l'avantage de son inconvénient : toutes les informations et donc beaucoup d'information. Le challenge est donc alors d'éliminer les informations redondantes et non importantes dans la classe recherchée en extrayant l'information représentant la classe, supposé commune à chaque échantillon de cette classe. Dans le prochain paragraphe, nous allons nous recentrer sur l'image et la vidéo comme application à l'analyse tensorielle.

Bibliographie

[1] J. Aach and G. M. Church. Aligning gene expression time series with time warping algorithms. *Bioinformatics*, 17(6) :495–508, 2001. (Cited on page 80.)

[2] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8m : A large-scale video classification benchmark. *CoRR*, abs/1609.08675, 2016. (Cited on page 87.)

[3] J. Assfalg, M. Bertini, C. Colombo, A. Del Bimbo, and W. Nunziati. Semantic annotation of soccer videos : automatic highlights identification. *CVIU*, 92(2) :285–305, 2003. (Cited on page 17.)

[4] W. J. Baddar, G. Gu, S. Lee, and Y. M. Ro. Dynamics transfer gan : Generating video by transferring arbitrary temporal dynamics from a source video to a single target image. *arXiv preprint arXiv :1712.03534*, 2017. (Cited on page 9.)

[5] H. Ben-Younes, R. Cadene, M. Cord, and N. Thome. Mutan : Multimodal tucker fusion for visual question answering. In *Proc. IEEE Int. Conf. Comp. Vis*, volume 3, 2017. (Cited on pages viii, 44 and 45.)

[6] J. A. Bengua, H. N. Phien, H. D. Tuan, and M. N. Do. Efficient tensor completion for color image and video recovery : Low-rank tensor train. *IEEE Transactions on Image Processing*, 26(5) :2466–2479, 2017. (Cited on pages viii, 45 and 46.)

[7] A. Bhattacharyya, M. Malinowski, B. Schiele, and M. Fritz. Long-term image boundary prediction. *arXiv preprint arXiv :1611.08841*, 2016. (Cited on page 8.)

[8] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould. Dynamic image networks for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3034–3042, 2016. (Cited on pages vii and 7.)

[9] J. Böer. Multiple alignment using hidden markov models. *proteins*, 4 :14, 1995. (Cited on page 80.)

[10] M. Borga. Canonical correlation : a tutorial. *On line tutorial http ://people. imt. liu. se/magnus/cca*, 4 :5, 2001. (Cited on page 53.)

[11] A. Bruderlin and L. Williams. Motion signal processing. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 97–104. ACM, 1995. (Cited on page 80.)

[12] W. Byeon, Q. Wang, R. K. Srivastava, and P. Koumoutsakos. Fully context-aware video prediction. *arXiv preprint arXiv :1710.08518*, 2017. (Cited on page 8.)

[13] J. Carreira and A. Zisserman. Quo vadis, action recognition ? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733. IEEE, 2017. (Cited on pages vii, 4 and 5.)

[14] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan. Tensor decompositions for signal processing applications : From two-way to multiway component analysis. *IEEE Signal Processing Magazine*, 32(2) :145–163, 2015. (Cited on pages viii, ix, xii, 38, 41, 43, 54, 57 and 112.)

[15] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari. *Nonnegative matrix and tensor factorizations : applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009. (Cited on pages viii, xi, xii, 36, 102, 106 and 107.)

[16] N. Cohen, O. Sharir, and A. Shashua. On the expressive power of deep learning : A tensor analysis. In *Conference on Learning Theory*, pages 698–728, 2016. (Cited on page 40.)

[17] Fédération Internationale. de Football Association (FIFA). 2014 fifa world cup brazil : matches description.
Germany-vs-Portugal : www.fifa.com/worldcup/matches/round=255931/match=300186475/live-blog.html
Nigeria vs Argentine www.fifa.com/worldcup/matches/round=255931/match=300186458/live-blog.html
France vs Honduras : www.fifa.com/worldcup/matches/round=255931/match=300186496/live-blog.html
Switzerland vs France : www.fifa.com/worldcup/matches/round=255931/match=300186514/live-blog.html). (Cited on page 22.)

[18] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4) :1253–1278, 2000. (Cited on page 37.)

[19] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. Del Bimbo. Reconnaissance d actions humaines 3d par l analyse de forme des trajectoires de mouvement. In *Compression et Représentation des Signaux Audiovisuels (CORESA)*, 2014. (Cited on page 2.)

[20] A. Diba, V. Sharma, and L. Van Gool. Deep temporal linear encoding networks. In *Computer Vision and Pattern Recognition*, 2017. (Cited on page 4.)

[21] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015. (Cited on page 4.)

[22] L.-Y. Duan, M. Xu, T.-S. Chua, Q. Tian, and C.-S. Xu. A mid-level representation framework for semantic sports video analysis. In *Proceedings of the*

eleventh ACM international conference on Multimedia, pages 33–44. ACM, 2003. (Cited on page 16.)

[23] L.-Y. Duan, M. Xu, Q. Tian, C.-S. Xu, and J. S. Jin. A unified framework for semantic shot classification in sports video. *IEEE Transactions on Multimedia*, 7(6) :1066–1083, 2005. (Cited on page 25.)

[24] S. Ebadollahi, L. Xie, S.-F. Chang, and J. R. Smith. Visual event detection using multi-dimensional concept dynamics. In *Multimedia and Expo, 2006 IEEE International Conference on*, pages 881–884. IEEE, 2006. (Cited on page 2.)

[25] A. Edison and C. Jiji. Optical acceleration for motion description in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 39–47, 2017. (Cited on page 6.)

[26] A. Ekin. Generic play-break event detection for summarization and hierarchical sports video analysis. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 1, pages I–169. IEEE, 2003. (Cited on page 16.)

[27] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Image Analysis*, pages 363–370. Springer, 2003. (Cited on page 18.)

[28] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. (Cited on page 3.)

[29] C. Feichtenhofer, A. Pinz, and R. Wildes. Spatiotemporal residual networks for video action recognition. In *Advances in neural information processing systems*, pages 3468–3476, 2016. (Cited on pages 3 and 5.)

[30] B. Fernando, S. Gavves, O. Mogrovejo, J. Antonio, A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *Proceedings CVPR 2015*, pages 5378–5387, 2015. (Cited on page 6.)

[31] T. Garipov, D. Podoprikhin, A. Novikov, and D. Vetrov. Ultimate tensorization : compressing convolutional and fc layers alike. *arXiv preprint arXiv :1611.03214*, 2016. (Cited on page 43.)

[32] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many : Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6) :643–660, 2001. (Cited on page 68.)

[33] G. Gkioxari and J. Malik. Finding action tubes. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 759–768. IEEE, 2015. (Cited on page 2.)

[34] D. Gong and G. Medioni. Dynamic manifold warping for view invariant action recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 571–578. IEEE, 2011. (Cited on page 81.)

[35] Y. Gong, L. T. Sin, C. H. Chuan, H. Zhang, and M. Sakauchi. Automatic parsing of tv soccer programs. In *Multimedia Computing and Systems, 1995.,*

Proceedings of the International Conference on, pages 167–174. IEEE, 1995. (Cited on page 16.)

[36] X. Guo, X. Huang, L. Zhang, L. Zhang, A. Plaza, and J. A. Benediktsson. Support tensor machines for classification of hyperspectral remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 54(6) :3248–3264, 2016. (Cited on pages viii, 39 and 40.)

[37] A. Hanjalic. Generic approach to highlights extraction from a sport video. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 1, pages I–1. IEEE, 2003. (Cited on page 16.)

[38] M. A. Hasan. On multi-set canonical correlation analysis. In *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, pages 1128–1133. IEEE, 2009. (Cited on pages 81 and 85.)

[39] E. Hsu, K. Pulli, and J. Popović. Style translation for human motion. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 1082–1089. ACM, 2005. (Cited on page 80.)

[40] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *NIPS*, pages 2017–2025, 2015. (Cited on pages xi, 97, 98, 102, 103 and 104.)

[41] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1) :221–231, 2013. (Cited on page 5.)

[42] C. Jia, G. Zhong, and Y. R. Fu. Low-rank tensor learning with discriminant analysis for action classification and image recovery. In *AAAI*, pages 1228–1234, 2014. (Cited on page 51.)

[43] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. Kautz. Super slomo : High quality estimation of multiple intermediate frames for video interpolation. *arXiv preprint arXiv :1712.00080*, 2017. (Cited on page 8.)

[44] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14(2) :201–211, 1973. (Cited on page 1.)

[45] I. N. Junejo, E. Dexter, I. Laptev, and P. Perez. View-independent action recognition from temporal self-similarities. *IEEE transactions on pattern analysis and machine intelligence*, 33(1) :172–185, 2011. (Cited on page 80.)

[46] S. E. Kahou, V. Michalski, and R. Memisevic. Ratm : recurrent attentive tracking model. *arXiv preprint arX-iv : 1510.08660*, 2015. (Cited on page 6.)

[47] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. (Cited on pages 4 and 92.)

[48] C. Keimel, M. Rothbucher, H. Shen, and K. Diepold. Video is a cube. *IEEE Signal Processing Magazine*, 28(6) :41–49, 2011. (Cited on pages viii and 36.)

[49] O. Kihl, B. Tremblais, and B. Augereau. Multivariate orthogonal polynomials to extract singular points. In *IEEE International Conference on Image Processing 2008. ICIP 2008.*, pages –, San Diego, CA, United States, Oct. 2008. (Cited on pages vii, 17 and 18.)

[50] T.-K. Kim. Hand gesture dataset website. `https://labicvl.github.io/ges_db.htm`. Accessed : 2018-06-25. (Cited on pages ix and 59.)

[51] T.-K. Kim and R. Cipolla. Gesture recognition under small sample size. In *Asian conference on computer vision*, pages 335–344. Springer, 2007. (Cited on page 59.)

[52] T. S. Kim and A. Reiter. Interpretable 3d human action analysis with temporal convolutional networks. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1623–1631. IEEE, 2017. (Cited on page 4.)

[53] A. Kläser, M. Marszałek, C. Schmid, and A. Zisserman. Human focused action localization in video. In *European Conference on Computer Vision*, pages 219–233. Springer, 2010. (Cited on page 2.)

[54] J. Kossaifi, A. Khanna, Z. Lipton, T. Furlanello, and A. Anandkumar. Tensor contraction layers for parsimonious deep nets. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1940–1946. IEEE, 2017. (Cited on pages viii and 44.)

[55] I. Kotsia, W. Guo, and I. Patras. Higher rank support tensor machines for visual recognition. *Pattern Recognition*, 45(12) :4192–4203, 2012. (Cited on page 39.)

[56] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. (Cited on pages 3 and 16.)

[57] I. Laptev. On space-time interest points. *International journal of computer vision*, 64(2-3) :107–123, 2005. (Cited on page 3.)

[58] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. (Cited on pages 3 and 4.)

[59] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, and V. Lempitsky. Speeding-up convolutional neural networks using fine-tuned cp-decomposition. *arXiv preprint arXiv :1412.6553*, 2014. (Cited on pages viii, 40 and 42.)

[60] C.-S. Lee. Human action recognition using tensor dynamical system modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 21–25, 2017. (Cited on page 39.)

[61] R. Leonardi, P. Migliorati, and M. Prandini. Semantic indexing of soccer audio-visual sequences : a multimodal approach based on controlled markov chains. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(5) :634–643, 2004. (Cited on page 16.)

[62] Z. Li, K. Gavrilyuk, E. Gavves, M. Jain, and C. G. Snoek. Videolstm convolves, attends and flows for action recognition. *Computer Vision and Image Understanding*, 166 :41–50, 2018. (Cited on page 6.)

[63] X. Liang, L. Lee, W. Dai, and E. P. Xing. Dual motion gan for future-flow embedded video prediction. *arXiv preprint*, 2017. (Cited on page 8.)

[64] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. *IEEE transactions on pattern analysis and machine intelligence*, 35(1) :208–220, 2013. (Cited on page 45.)

[65] Z. Liu, R. Yeh, X. Tang, Y. Liu, and A. Agarwala. Video frame synthesis using deep voxel flow. In *ICCV*, October 2017. (Cited on page 8.)

[66] Z. Liu, L. Yuan, X. Tang, M. Uyttendaele, and J. Sun. Fast burst images denoising. *ACM Transactions on Graphics (TOG)*, 33(6) :232, 2014. (Cited on page 7.)

[67] E. F. Lock, K. A. Hoadley, J. S. Marron, and A. B. Nobel. Joint and individual variation explained (jive) for integrated analysis of multiple data types. *The annals of applied statistics*, 7(1) :523, 2013. (Cited on page 53.)

[68] A. P. B. Lopes, R. S. Oliveira, J. M. de Almeida, and A. d. A. Araújo. Comparing alternatives for capturing dynamic information in bag-of-visual-features approaches applied to human actions recognition. In *Multimedia Signal Processing, 2009. MMSP'09. IEEE International Workshop on*, pages 1–6. IEEE, 2009. (Cited on page 36.)

[69] D. Mahajan, F.-C. Huang, W. Matusik, R. Ramamoorthi, and P. Belhumeur. Moving gradients : a path-based method for plausible image interpolation. In *ACM Transactions on Graphics (TOG)*, volume 28, page 42. ACM, 2009. (Cited on page 7.)

[70] H. Mao, S. Han, J. Pool, W. Li, X. Liu, Y. Wang, and W. J. Dally. Exploring the granularity of sparsity in convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017. (Cited on page 40.)

[71] P. K. Mital, T. J. Smith, R. L. Hill, and J. M. Henderson. Clustering of gaze during dynamic scene viewing is predicted by motion. *Cognitive Computation*, 3(1) :5–24, 2011. (Cited on page 1.)

[72] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets : Deep networks for video classification. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 4694–4702. IEEE, 2015. (Cited on page 4.)

[73] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International journal of computer vision*, 79(3) :299–318, 2008. (Cited on pages 3 and 4.)

[74] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Com-*

puter Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 1717–1724. IEEE, 2014. (Cited on page 3.)

[75] I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5) :2295–2317, 2011. (Cited on page 42.)

[76] H. Pan, P. Van Beek, and M. I. Sezan. Detection of slow-motion replay segments in sports video for highlights generation. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 3, pages 1649–1652. IEEE, 2001. (Cited on page 28.)

[77] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318, 2013. (Cited on page 4.)

[78] A. H. Phan and A. Cichocki. Tensor decompositions for feature extraction and classification of high dimensional datasets. *Nonlinear theory and its applications, IEICE*, 1(1) :37–68, 2010. (Cited on pages ix and 51.)

[79] X. Qian. *Global Motion Estimation and Its Applications*. INTECH Open Access Publisher, 2012. (Cited on page 25.)

[80] L. R. Rabiner and B.-H. Juang. *Fundamentals of speech recognition*, volume 14. PTR Prentice Hall Englewood Cliffs, 1993. (Cited on page 80.)

[81] A. Raventos, R. Quijada, L. Torres, and F. Tarres. Automatic summarization of soccer highlights using audio-visual descriptors. *arXiv preprint arXiv :1411.6496*, 2014. (Cited on page 17.)

[82] C. L. M. D. F. René and V. A. R. G. D. Hager. Temporal convolutional networks for action segmentation and detection. *IEEE International Conference on Computer Vision (ICCV)*, 2017. (Cited on page 4.)

[83] Y. Runsheng, S. Zhenyu, and Q. Laiyun. Unsupervised learning aids prediction : Using future representation learning variantial autoencoder for human action prediction. *arXiv preprint arXiv :1711.09265*, 2017. (Cited on page 7.)

[84] D. Sadlier, N. E. O'Connor, et al. Event detection in field sports video using audio-visual features and a support vector machine. *Circuits and Systems for Video Technology, IEEE Transactions on*, 15(10) :1225–1233, 2005. (Cited on page 16.)

[85] S. M. Safdarnejad, X. Liu, and L. Udpa. Robust global motion compensation in presence of predominant foreground. In *BMVC*, pages 21–1, 2015. (Cited on page 24.)

[86] M. Saito, E. Matsumoto, and S. Saito. Temporal generative adversarial nets with singular value clipping. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2830–2839, 2017. (Cited on page 9.)

[87] S. C. Sajjan and C. Vijaya. Comparison of dtw and hmm for isolated word recognition. In *Pattern Recognition, Informatics and Medical Engineering (PRIME), 2012 International Conference on*, pages 466–470. IEEE, 2012. (Cited on page 80.)

[88] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. (Cited on page 5.)

[89] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 1961–1970. IEEE, 2016. (Cited on pages 4 and 99.)

[90] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852, 2015. (Cited on page 7.)

[91] Y. Su, H. Wang, P. Jing, and C. Xu. A spatial-temporal iterative tensor decomposition technique for action and gesture recognition. *Multimedia Tools and Applications*, 76(8) :10635–10652, 2017. (Cited on page 51.)

[92] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, et al. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. (Cited on page 87.)

[93] R. Szeliski. Prediction error as a quality metric for motion and stereo. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 781–788. IEEE, 1999. (Cited on pages 7 and 8.)

[94] Y. Tabii and R. O. Thami. A new method for soccer video summarizing based on shot detection, classification and finite state machine. In *Proceedings of The 5th international conference SETIT*, 2009. (Cited on page 17.)

[95] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *European conference on computer vision*, pages 140–153. Springer, 2010. (Cited on page 5.)

[96] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4489–4497, 2015. (Cited on pages xi, 3, 5, 91 and 99.)

[97] G. Trigeorgis, M. A. Nicolaou, B. W. Schuller, and S. Zafeiriou. Deep canonical time warping for simultaneous alignment and representation learning of sequences. *IEEE Trans. PAMI*, pages 1128–1138, 2018. (Cited on pages 81 and 97.)

[98] G. Trigeorgis, M. A. Nicolaou, S. Zafeiriou, and B. W. Schuller. Deep canonical time warping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5110–5118, 2016. (Cited on page 81.)

[99] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz. Mocogan : Decomposing motion and content for video generation. *arXiv preprint arXiv :1707.04993*, 2017. (Cited on page 9.)

[100] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2017. (Cited on page 5.)

[101] M. A. O. Vasilescu. A multilinear (tensor) algebraic framework for computer graphics, computer vision, and machine learning. *University of Toronto*, 2009. (Cited on page 48.)

[102] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles : Tensorfaces. In *European Conference on Computer Vision*, pages 447–460. Springer, 2002. (Cited on pages viii, ix, 46, 47, 48 and 50.)

[103] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621, 2016. (Cited on page 9.)

[104] H. T. Vu, C. Carey, and S. Mahadevan. Manifold warping : Manifold alignment over time. In *AAAI*, volume 1, page 8, 2012. (Cited on page 81.)

[105] J. Wan, Y. Zhao, S. Zhou, I. Guyon, S. Escalera, and S. Z. Li. Chalearn looking at people rgb-d isolated and continuous datasets for gesture recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 56–64, 2016. (Cited on page 89.)

[106] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169–3176. IEEE, 2011. (Cited on pages 3 and 16.)

[107] H. Wang, W. Yang, C. Yuan, H. Ling, and W. Hu. Human activity prediction using temporally-weighted generalized time warping. *Neurocomputing*, 225 :139–147, 2017. (Cited on page 81.)

[108] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks for action recognition in videos. *IEEE transactions on pattern analysis and machine intelligence*, 2018. (Cited on pages 5 and 6.)

[109] X. Wang, A. Farhadi, and A. Gupta. Actions˜ transformations. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2658–2667, 2016. (Cited on page 6.)

[110] R. Wilbur and A. C. Kak. Purdue rvl-slll american sign language database. *School of Electrical and Computer Engineering, Purdue University*, 2006. (Cited on page 88.)

[111] C. Wu, Y.-F. Ma, H.-J. Zhan, and Y.-Z. Zhong. Events recognition by semantic inference for sports video. In *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on*, volume 1, pages 805–808. IEEE, 2002. (Cited on page 16.)

[112] L. Xie, S.-F. Chang, A. Divakaran, and H. Sun. Structure analysis of soccer video with hidden markov models. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 4, pages IV–4096. IEEE, 2002. (Cited on page 16.)

[113] Z. Xiong, X. S. Zhou, Q. Tian, Y. Rui, and T. S. Huang. Semantic retrieval of video. *IEEE Signal Processing Magazine*, 23(2) :18, 2006. (Cited on page 16.)

[114] H. Xu, A. Das, and K. Saenko. R-c3d : Region convolutional 3d network for temporal activity detection. In *The IEEE International Conference on Computer Vision (ICCV)*, volume 6, page 8, 2017. (Cited on page 91.)

[115] P. Xu, L. Xie, S.-F. Chang, A. Divakaran, A. Vetro, and H. Sun. Algorithms and system for segmentation and structure analysis in soccer video. In *ICME*, volume 1, pages 928–931, 2001. (Cited on page 16.)

[116] Y. Yang, D. Krompass, and V. Tresp. Tensor-train recurrent neural networks for video classification. *arXiv preprint arXiv :1707.01786*, 2017. (Cited on page 43.)

[117] Q. Ye, Q. Huang, W. Gao, and S. Jiang. Exciting event detection in broadcast soccer video with mid-level description and incremental learning. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 455–458. ACM, 2005. (Cited on pages 17 and 24.)

[118] D. Yow, B.-L. Yeo, M. Yeung, and B. Liu. Analysis and presentation of soccer highlights from digital video. In *proc. ACCV*, volume 95, pages 499–503, 1995. (Cited on page 17.)

[119] H. M. Zawbaa, N. El-Bendary, A. E. Hassanien, and T. Kim. Event detection based approach for soccer video summarization using machine learning. *International Journal of Multimedia and Ubiquitous Engineering*, 7(2) :63–80, 2012. (Cited on pages 17 and 28.)

[120] J. Zhang, C. Xu, P. Jing, C. Zhang, and Y. Su. A tensor-driven temporal correlation model for video sequence classification. *IEEE signal processing letters*, 23(9) :1246–1249, 2016. (Cited on page 39.)

[121] Q. Zhao, C. F. Caiafa, D. P. Mandic, Z. C. Chao, Y. Nagasaka, N. Fujii, L. Zhang, and A. Cichocki. Higher order partial least squares (hopls) : a generalized multilinear regression method. *IEEE transactions on pattern analysis and machine intelligence*, 35(7) :1660–1673, 2013. (Cited on pages ix, xii, 52, 53, 112, 113 and 114.)

[122] F. Zhou and F. De la Torre. Generalized canonical time warping. *IEEE transactions on pattern analysis and machine intelligence*, 38(2) :279–294, 2016. (Cited on pages x, xi, 81, 82, 85 and 87.)

[123] F. Zhou and F. Torre. Canonical time warping for alignment of human behavior. In *Advances in neural information processing systems*, pages 2286–2294, 2009. (Cited on pages 81 and 84.)

[124] G. Zhou, A. Cichocki, Y. Zhang, and D. P. Mandic. Group component analysis for multiblock data : Common and individual feature extraction. *IEEE transactions on neural networks and learning systems*, 27(11) :2426–2439, 2016. (Cited on pages ix, x, 53, 55, 56, 57, 58, 68, 69 and 71.)

[125] G. Zhou, Q. Zhao, Y. Zhang, T. Adalı, S. Xie, and A. Cichocki. Linked component analysis from matrices to high-order tensors : Applications to biomedical data. *Proceedings of the IEEE*, 104(2) :310–331, 2016. (Cited on pages 53 and 54.)

[126] Y. Zhou, Y. Song, and T. L. Berg. Image2gif : Generating cinemagraphs using recurrent deep q-networks. *arXiv preprint arXiv :1801.09042*, 2018. (Cited on page 9.)