Détection de comportements et d'événements potentiellement mortels dans les prisons à partir d'analyse de vidéos par intelligence artificielle

par

Alban MAIN DE BOISSIERE

MÉMOIRE PAR ARTICLES PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE AVEC MÉMOIRE EN GÉNIE ÉLECTRIQUE
M. Sc. A.

MONTRÉAL, LE 16 JUIN 2020

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE:

Mme Rita Noumeir, directeur de mémoire
Département de génie électrique à l'École de technologie supérieure

M. Mohamed Cheriet, président du jury
Département de génie des systèmes à l'École de technologie supérieure

Mme Catherine Laporte, membre du jury
Département de génie électrique à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 22 MAI 2020

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Tout d'abord, je tiens à remercier ma directrice Rita Noumeir pour son expertise technique, son suivi et ses conseils pertinents tout au long du projet. La confiance qu'elle m'a accordée m'a permis d'explorer avec autonomie le fabuleux univers de la recherche et de l'apprentissage profond.

Je remercie également notre partenaire industriel, Aerosystems International (ASI), de proposer et financer le projet. Celui-ci est particulièrement concret et avec une forte retombée.

Je tiens à remercier Nicolas Lemieux et Oussema Keskes, mes partenaires de laboratoire. Avancer ensemble sur un projet commun est particulièrement stimulant.

Un grand merci à mes parents, qui me soutiennent, et à mon grand frère qui a toujours été de bon conseil.

Merci à Laurent, mon partenaire de galère et d'escalade. Merci à Aurélien, un vrai bro.

Et enfin, merci à Chloé qui illumine mon quotidien.

Détection de comportements et d'événements potentiellement mortels dans les prisons à partir d'analyse de vidéos par intelligence artificielle

Alban MAIN DE BOISSIERE

RÉSUMÉ

Le suicide est un phénomène complexe. Selon l'Organisation Mondiale de la Santé (WHO), une personne se donne la mort toutes les 40 secondes, soit 800 000 personnes par an. L'acte est d'autant plus fréquent dans les prisons. Les méthodes d'automutilation les plus fréquentes sont la pendaison, majoritaire, et le sectionnement.

Si détectées suffisamment tôt, les tentatives de suicide peuvent mener à une hospitalisation rapide et à une survie probable. La télévision en circuit fermé est une solution, mais, lorsqu'un agent a plusieurs caméras à surveiller, il perd 95% de sa capacité d'attention en une vingtaine de minutes. La vidéosurveillance humaine soulève également des questionnements d'éthique. Ce mémoire s'inscrit dans une volonté d'automatiser la surveillance en détectant les comportements dangereux. Aucune reconnaissance faciale ou d'identité n'est effectuée, aucun enregistrement.

Nous proposons une méthode de reconnaissance d'activités humaines par intelligence artificielle. Les caméras rouge-vert-bleu + profondeur (RGB+D), comme la *Kinect* de *Microsoft*, sont des outils intéressants pour la reconnaissance d'activités. Celles-ci proposent différents flux de données : vidéo rouge-vert-bleu (RGB), infrarouge, nuage de points et représentation dans une espace à trois dimensions (3D) des articulations du squelette humain. Les images RGB ne sont pas adaptées à un contexte de sécurité, car elles ne fournissent aucune information dans le noir. Le squelette est une représentation intéressante de par sa faible dimensionnalité et son insensibilité au décor environnant, mais s'avère peu discriminant pour des actions fines impliquant des objets. Or, les tentatives de suicide impliquent l'utilisation d'objets. Nos travaux montrent que l'infrarouge est une alternative puissante qui fonctionne dans le noir.

Une architecture profonde est proposée, combinant infrarouge et articulations 3D. Deux modules indépendants extraient des caractéristiques propres à chaque flux. Un troisième et dernier module étudie les caractéristiques conjointement et propose une classification finale. L'architecture développée obtient des résultats dignes de l'état de l'art sur la plus grande base de données RGB+D publique, NTU RGB+D, de reconnaissance d'activités humaines à ce jour.

Ces résultats sont ensuite transférés d'un contexte *offline* à un réseau de reconnaissance anticipée *online*. Le nouveau réseau peut être utilisé pour de la détection d'activités *online*, qui réplique un scénario temps réel, en ajoutant un module de proposition de segments temporels. À nouveau, nous obtenons des résultats dignes de l'état de l'art sur NTU RGB+D et PKU-MMD, deux bases de données de référence.

Mots-clés: Apprentissage profond, Détection d'activités, Infrarouge, Reconnaissance anticipée, Reconnaissance d'activités, Squelette, Vision par ordinateur

Detection of behaviors and potentially deadly events in prison using video analysis and artificial intelligence

Alban MAIN DE BOISSIERE

ABSTRACT

Suicide is a complex phenomenon. According to the World Health Organization (WHO), a person kills itself every 40 seconds, for a total of 800,000 deaths per year. This is even more frequent in prison. The most common self-harm methods are hanging then wrist-cutting.

If detected soon enough, suicide attempts can lead to an early hospitalization and a likely survival. Closed-circuit television is a potential solution, but, a security agent loses 95% of its attention in a twenty-minute span when working with multiple cameras. Human surveillance also raises ethical preoccupations. This works aims to automate surveillance by detecting dangerous behaviors. No facial or identity recognition is performed. Data are not recorded.

We propose a human action recognition framework. Red-green-blue + depth (RGB+D) cameras, such as the Microsoft Kinect, are powerful new tools for action recognition. They propose different streams : RGB video, infrared, three-dimensional (3D) point clouds, and estimated human skeleton. RGB images are not suited to security applications as they do not yield any information in the dark. Skeleton data are an interesting representation because of their low dimensionality and their insensibility to background information, but prove insufficient when detecting object-related actions or actions with similar motions. Suicides fall into this category. We find infrared videos to be a strong alternative while working in the dark.

A deep learning architecture is proposed, combining infrared video with 3D pose data. Two independent modules extract features from each stream. A third and final module fuses those features and studies them conjointly before emitting a final prediction. We achieve state-of-the-art results on the largest RGB+D action recognition dataset to date : NTU RGB+D.

Those results are then transferred from an offline context to an online early prediction network. The new network can then be used for online action detection, which mimics real-time scenarios, with an additional temporal segment proposal module. Again, we achieve state-of-the-art results on two benchmark datasets : NTU RGB+D and PKU-MMD.

Keywords: Action recognition, Computer vision, Deep learning, Early prediction, Infrared, Online action detection, Skeleton

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

2D	Deux dimensions
3D	Trois dimensions
ASI	*Aerosystems International Inc.*
CNN	*Convolutional Neural Network* / Réseau de Neurones Convolutifs
CS	*Cross-Subject*
CSC	*Correctional Service Canada* / Service correctionnel Canada
CV	*Cross-View*
ECCV	*European Conference on Computer Vision*
IASP	*International Association for Suicide Prevention* / Association Internationale pour la Prévention du Suicide
IEEE	*Institute of Electrical and Electronics Engineers*
LSTM	*Long Short Term Memory* / Longue mémoire à court terme
NTU	*Nanyang Technological University* / Université de Technologie de Nanyang
PKU-MMD	*Peking University Multi-Modality Dataset*
R-CNN	*Region-CNN* / CNN par région
RGB	*Red-Green-Blue* / Rouge-Vert-Bleu
RGB+D	*Red-Green-Blue* + *Depth* / Rouge-Vert-Bleu + Profondeur
RNN	*Recurrent Neural Network* / Réseau de Neurones Récurrent
TRN	*Temporal Recurrent Network* / Réseau Récurrent Temporel
STA	*Spatiotemporal Accumulator* / Accumulateur Spatiotemporel
WHO	*World Health Organization* / Organisation Mondiale de la Santé
YOLO	*You Only Look Once* / "Vous ne regardez qu'une fois"

INTRODUCTION

Le suicide est un phénomène complexe. Selon l'Organisation Mondiale de la Santé (WHO), une personne se donne la mort toutes les 40 secondes, soit 800 000 personnes par an (WHO, 2019). Ce phénomène est d'autant plus fréquent dans les prisons. Les méthodes d'automutilation les plus fréquentes sont la pendaison, majoritaire, puis le sectionnement.

Si détectées suffisamment tôt, les tentatives de suicide peuvent mener à une hospitalisation rapide et à une survie probable. Malheureusement, les techniques actuelles sont exigeantes en ressources financières et humaines. Qui plus est, lorsqu'un agent a plusieurs caméras à surveiller, il perd 95% de sa capacité d'attention en une vingtaine de minutes (Ainsworth, 2002). La vidéosurveillance humaine soulève également des questionnements d'ordre éthique. Ce mémoire s'inscrit dans une volonté d'automatiser la supervision des détenus en détectant les comportements dangereux de façon non intrusive. Aucune reconnaissance faciale ou d'identité n'est effectuée, aucun enregistrement et aucune surveillance par un humain.

La reconnaissance d'activités humaines est une branche importante de la vision par ordinateur. Ses applications incluent la robotique, la conduite autonome ou encore la surveillance. Récemment, l'avènement de l'apprentissage profond a permis une hausse rapide des performances sur les principales bases de données publiques (Wang, Li, Ogunbona, Wan & Escalera, 2018a).

Parallèlement, la démocratisation de caméras rouge-vert-bleu + profondeur (RGB+D) grand public, comme la *RealSense* d'*Intel* (Keselman, Iselin Woodfill, Grunnet-Jepsen & Bhowmik, 2017) ou la *Kinect* de *Microsoft* (Zhang, 2012) ont permis de bâtir des bases de données de reconnaissance d'activités humaines proposant les différents flux de données qu'offrent ces caméras. Ces derniers sont : vidéo rouge-vert-bleu (RGB), vidéo infrarouge, nuage de points 3D de l'environnement, images de profondeur à partir des nuages de points et coordonnées 3D des articulations du squelette humain.

L'acquisition du squelette en temps réel (Shotton, Fitzgibbon, Cook, Sharp, Finocchio, Moore, Kipman & Blake, 2011) promulgue ce flux au rang de candidat de choix pour la reconnaissance d'activités. En effet, sa faible dimensionnalité offre un temps d'inférence rapide et donc un processus itératif de recherche accéléré. Par ailleurs, des travaux précurseurs de psychophysique ont montré que le squelette 3D est un descripteur puissant pour la reconnaissance d'actions (Johansson, 1973).

Néanmoins, la limite discriminatoire du squelette 3D se situe au niveau d'actions de gestuelles similaires et/ou impliquant des objets. Par exemple, il paraît délicat de faire la différence entre "lire un livre" et "jouer sur son téléphone" simplement avec des coordonnées 3D, même pour un humain. Dans notre cas, les tentatives de suicide impliquent des objets (corde, couteau, lame de rasoir) et ne doivent pas être confondues avec des actions similaires (mettre un chandail, se frotter les mains).

Les autres flux largement étudiés dans la littérature scientifique sont la vidéo RGB et la profondeur (projetée sur un plan deux dimensions (2D) pour former une image). Bien que précis, le flux RGB n'est pas adapté pour une application de sécurité, car il est sensible aux conditions d'éclairage. Les images de profondeur demeurent une option, mais sont peu précises. La représentation d'objets est grossière puisqu'il s'agit d'une projection de points 3D sur une image 2D. Reste le flux infrarouge, riche en informations, mais qui n'a été que très peu étudié à notre connaissance.

À la lumière de ces réflexions préalables, nous proposons une architecture de réseau de neurones combinant infrarouge et squelette 3D. Celle-ci est évaluée sur *Nanyang Technological University* RGB+D (NTU RGB+D) (Shahroudy, Liu, Ng & Wang, 2016), la plus importante base de données de reconnaissance d'activités humaines RGB+D à notre connaissance. Aucune base de données de comportements suicidaires n'existe, ce qui motive le choix de travailler sur une base de données généraliste. Les résultats obtenus sont dignes de l'état de l'art et font l'objet d'un article de journal présenté au chapitre 3, actuellement à l'étude.

La reconnaissance d'activités n'est pas directement transposable à une application en temps réel. Domaine similaire, la reconnaissance anticipée consiste à reconnaître une action avant que celle-ci ne soit terminée. Mais cette branche n'est pas non plus transposable directement à un système temps réel. La détection d'activités *online* est ce qui s'en rapproche le plus. Dans la littérature scientifique, ces domaines sont bien souvent étudiés séparément. Une méthodologie est présentée dans un article de conférence au chapitre 5 pour relier ces trois domaines de recherche.

Une stratégie de prétraitement *online* sur le flux infrarouge est proposée. Une architecture de reconnaissance anticipée *online* est construite à partir du module infrarouge implémenté dans l'article (chapitre 3). On montre qu'on peut utiliser cette architecture pour de la détection d'activités *online*. La méthode se base sur une stratégie d'apprentissage distillé et obtient des résultats dignes de l'état de l'art sur deux bases de données de référence.

L'ensemble des codes du projet et la documentation sont disponibles en ligne (annexe I). Un tutoriel pour prendre en main le modèle présenté au chapitre 3 est inclus (annexe II).

CHAPITRE 1

REVUE DE LITTÉRATURE

La revue de littérature se divise en deux axes majeurs. Un premier cherche à comprendre le suicide d'un point de vue psychophysique. L'objectif est d'obtenir une vision globale du phénomène dans les prisons, avec un accent sur le Canada, et de comprendre pourquoi et comment il se manifeste. Le deuxième axe explore les domaines techniques que sont la reconnaissance anticipée et la détection d'activités humaines.

1.1 Le suicide en prison

Au Canada, la prévention du suicide est devenue une politique d'ordre public durant les années 1970 (Guenat,Claire ; Bérard, 2018). L'Organisation Mondiale de la Santé créa l'Association Internationale pour la Prévention du Suicide (IASP) en 1960 comme étendard de sensibilisation.

Il faut attendre 1979 pour qu'un archétype du suicidé incarcéré soit établi (Guenat, 2018). Le prisonnier a entre 20 et 30 ans. Le suicide arrive promptement après une crise. Son stress, et a fortiori la probabilité de passer à l'acte, est décuplé s'il est entouré de prisonniers violents. Il est ensuite placé dans une cellule de protection. Il a précédemment informé le personnel compétent de sa détresse et de son intention de mettre fin à ses jours. Il a déjà tenté de se suicider. L'équipe médicale et psychologique l'a examiné, mais n'a pas diagnostiqué de troubles psychotiques.

Plus récemment, (Fazel, Grann, Kling & Hawton, 2011) explorent des tentatives de suicide dans 12 pays différents, dont le Canada, entre 2003 et 2007. Sur 861 suicides déclarés, 810 ont été commis par des hommes. De prime abord, il n'existe pas de lien évident entre les taux de suicide d'un pays dans les prisons par rapport à la population générale. Au Canada, on déplore 70 suicides pour 100 000 incarcérés, soit 3,4 fois plus que pour la population générale. Les taux canadiens sont similaires à ceux de pays comme l'Australie ou la Nouvelle-Zélande et bien moins importants qu'en Europe.

Dernier article à ce jour à notre connaissance, Fazel, Ramesh & Hawton (2017) étudient les taux de suicide dans différents pays développés, dont le Canada, entre 2011 et 2014. L'étude confirme qu'il n'existe pas de lien entre fréquences de suicide en prison ou en liberté. Les résultats montrent un taux annuel de 27 suicides pour 100 000 prisonniers au Canada, une baisse significative par rapport à l'étude précédente. Encore une fois, le Canada figure parmi les pays les moins touchés par le phénomène. Mais ces résultats peuvent être sous-évalués selon Suto & Arnaut (2010). Il arrive que les institutions compétentes déclarent une cause de décès différemment selon que la victime soit morte en prison ou à l'hôpital. Vérifier l'exactitude de ces résultats va au-delà du champ d'études de ce mémoire. Néanmoins, ils sont confirmés par le Service Correctionnel du Canada (CSC, 2014). Même avec des taux comparativement faibles, le suicide reste une des causes de mort majeure dans les prisons canadiennes entre 1994 et 2014.

La méthode de suicide la plus courante est la pendaison. Dans un article de McKee (1998), 754 suicides dans des prisons de Caroline du Sud entre 1985 et 1994 ont été étudiés. La pendaison y est deux fois plus fréquente que le sectionnement (poignet ou bras). Qui plus est, la pendaison est 19 fois plus probable d'être fatale que le sectionnement. L'étude révèle un fait important, 63% des suicides ont lieu le premier jour d'emprisonnement. Un personnel psychiatrique apparaît comme fondamental, puisque 90% des prisonniers n'ont jamais été examinés.

Au Canada, ces résultats sont confirmés par le CSC (2014). Sur les 30 suicides présentés, 27 sont morts d'asphyxie, dont 25 par pendaison. À noter que 14 des suicides ont eu lieu en isolement.

1.2 Suicide par sectionnement

Cette section s'intéresse aux modalités d'un suicide par sectionnement. Ces informations sont utiles afin de correctement simuler ces données pour entraîner des modèles d'apprentissage machine. Un portrait correctement dressé de l'acte permettra à ces derniers de mieux généraliser. Comme discuté précédemment, le suicide par sectionnement est un phénomène rare. Les articles suivants proviennent d'analyses de tentatives de la population générale et de patients psychiatriques de différents pays.

1.2.1 Types de blessures

Selon Krywanczyk & Shapiro (2015), un homicide ou une tentative de suicide peuvent être différenciés en étudiant les spécificités d'une plaie. Il existe trois catégories, à savoir des plaies par poignardement, par incision ou d'hésitation, cette dernière étant spécifique aux tentatives de suicide. Le poignardement est rare en cas de suicide avec deux cas présentant ce type de plaie (8%), dont un cas (4%) en présentant 4. Selon Vassalini, Verzeletti & De Ferrari (2014), en cas de poignardement pour une tentative de suicide, la poitrine est principalement visée.

Les plaies d'hésitation incarnent une étape préliminaire qu'un suicidé peut effectuer. D'après Racette, Kremer, Desjarlais & Sauvageau (2008), elles apparaissent sous forme de coupes parallèles. Elles sont interprétées comme un signe d'incertitude et d'évaluation de la douleur. Selon Karlsson (1998), elles sont présentes dans 62% des suicides par sectionnement. Ce résultat est confirmé par Vassalini *et al.* (2014), où 64,3% des suicides présentent des plaies d'hésitation.

Des dégâts aux os ou cartilages sont rares. Ils n'apparaissent que dans 4% des suicides selon Krywanczyk & Shapiro (2015).

1.2.2 Localisation des blessures

La localisation des blessures d'auto-sectionnement est bien expliquée dans la littérature. L'écrasante majorité se situe sur le haut du corps. Dans un article de Brunel, Fermanian, Durigon & de la Grandmaison (2010), 4 régions anatomiques sont suggérées. La première inclut la tête, la nuque, le dos, les côtes et les mains. La deuxième : le cou, le thorax et l'abdomen. Région 3 : les avant-bras. Région 4 : une combinaison des régions 2 et 3. Sur les 48 suicides étudiés, 2,1% se situent au niveau de la région 1, 56,2% au niveau de la région 2, 20,8% au niveau des régions 3, 4,2% au niveau de la région 1 et 2, 12,5% au niveau de la région 4 et 4,2% sur toutes les régions. La région la plus représentée est la 2, ce qui peut être expliqué par le plus grand potentiel de fatalité de cette zone. En désaccord avec Krywanczyk & Shapiro (2015), l'auto-poignardement était le plus fréquent (43,8% contre 39,6% pour l'incision et 16,7% pour les deux).

D'après Krywanczyk & Shapiro (2015), les plaies sont plus fréquemment présentes aux extrémités (bras ou main) avec 48%, puis 40% pour le poignet et 32% pour le cou. D'autres régions moins fréquentes incluent : le creux du coude (12%), les membres inférieurs (8%) et l'abdomen (4%).

Dans un article de Vassalini *et al.* (2014), la localisation des plaies (fatales ou non) est grandement détaillée. La majorité apparaît au niveau de l'avant-bras gauche (67,9%) ou droit (50%), le cou (35,7%), la moitié gauche de la poitrine (32,1%). Les plaies fatales ont lieu au niveau du cou (32%), de la moitié gauche de la poitrine (28,6%), de l'avant-bras gauche (25%) et de l'avant-bras droit (14,3%). L'étude n'a pas conclu à une prédominance de sectionnement de l'avant-bras opposé à la main d'écriture. À ce sujet, Ersen, Kahveci, Saki, Tunali & Aksu (2017) étudient 41 individus, dont 32 droitiers. Parmi eux, 78% sectionnent leur avant-bras gauche, 7% le droit et 14% les deux.

Selon Karlsson (1998), en considérant tout type de blessure, le poignet est prédominent (59%), suivi du cou (32%). Les autres zones fréquentes sont le torse (24%), le creux du bras (15%), et les extrémités (13%).

Racette *et al.* (2008) étudient en détail les plaies d'hésitation. Elles sont prédominantes au niveau des membres supérieurs (69%) incluant le poignet (29%), la main (22%), l'avant-bras (10%) et le creux du bras (7%). À nouveau, aucune prédominance n'a été relevée entre côté gauche ou droit.

L'apparition des plaies est donc contrastée. Il convient donc de simuler le sectionnement de différentes zones lors de la création d'une base de données visant à entraîner des modèles de reconnaissance d'activités humaines.

1.2.3 Nombre de plaies

Un suicide par auto-sectionnement est généralement le résultat de plaies multiples. Brunel *et al.* (2010) déclarent une moyenne de 8,6 ± 18,8 plaies pour les suicides. Vassalini *et al.* (2014)

rapportent qu'une plaie unique est détectée dans 35,7% des cas. Et Krywanczyk & Shapiro (2015) comptent seulement 12% (pour 25 cas) des suicides par plaie unique. Deux plaies dans 16% des cas, 4 et plus pour le reste.

Selon la zone, le nombre de plaies moyen varie. Karlsson (1998) montre une moyenne de 2,87 plaies au poignet, la zone la plus abondante, suivi du cou (1,34 plaie) et du creux du bras (0,56 plaie).

Cependant, en considérant seulement le nombre de plaies sévères, étudiées par Karger, Niemeyer & Brinkmann (2000), la majorité sont des plaies uniques. Dans le groupe "suicide", 37% montrent une seule plaie grave, 15% en montrent 2, 20% en montrent 3, 14% entre 4 et 9, 9% entre 10 et 20 et 5% au-delà. Ces résultats confirment les travaux de Brunel *et al.* (2010).

Il est donc courant qu'un suicidé s'auto-sectionne plusieurs fois, ce qui représente une information de taille pour la création d'un système de détection automatisé. Il apparaît qu'environ un tiers ou moins des suicides ont lieu par plaie unique, soit une minorité.

1.2.4 Outils d'auto sectionnement

Peu d'études indiquent l'outil utilisé pour l'auto-sectionnement. Lorsque c'est le cas, les résultats sont contrastés. Cela est dû au type de population étudiée (psychiatrique, générale) qui va avoir une influence directe sur les outils disponibles. Néanmoins, les outils récurrents sont les lames de rasoir, les couteaux (de cuisine), les ciseaux et les copeaux de verre. D'après Ersen *et al.* (2017), 51% des sujets se sectionnent le poignet à l'aide d'une lame de rasoir, 36% avec un couteau, 9% avec des copeaux de verre et 4% avec des ciseaux. Dans une étude de Fujioka, Murakami, Masuda & Doi (2012), les patients s'auto-mutilant régulièrement utilisent principalement des lames de rasoir (88%). Dans d'autres études, le couteau est l'outil de prédilection. Selon Vassalini *et al.* (2014), 61% des suicides sont commis avec un couteau, contre 22% avec soit une lame de rasoir, soit un cutter. Karger *et al.* (2000) confirment ces résultats (62% pour les couteaux, 15% pour les lames de rasoir). De même dans (Karlsson, 1998) avec 32% pour les couteaux, 28% pour les rasoirs chez les hommes.

Dans les prisons, l'outil le plus utilisé est probablement la lame de rasoir. Le seul cas d'auto-mutilation décrit par le CSC (2014) utilise une lame de rasoir.

1.2.5 Dégâts sur les vêtements

Bien que rares, des cas de dégâts sur les vêtements sont parfois à déplorer. D'après Krywanczyk & Shapiro (2015), une telle situation a lieu dans deux cas (8% du groupe "suicide"). Des proportions similaires sont reportées dans les travaux de Karlsson (1998). Un unique cas certain (5%), deux incertains (10%), le reste ne présentant pas de dégâts.

1.2.6 Profil de la victime

Un portrait-robot psychiatrique d'un suicidé peut être créé à partir de la littérature scientifique. Fujioka *et al.* (2012) étudient 31 sujets avec des plaies auto-infligées. Ils sont divisés en 2 groupes selon le degré de sévérité des plaies : 15 dans le groupe "plaies profondes", 16 dans le groupe "plaies superficielles". Les patients dans le premier groupe ont généralement un passé d'auto-mutilation et pratiquent plus souvent. Les sujets du deuxième groupe se coupent des zones plus variées, avec n'importe quel outil à disposition.

Matsumoto, Yamaguchi, Chiba, Asami, Iseki & Hirayasu (2004) séparent les sujets en un groupe "non-auto-mutilé" et 3 groupes "auto-mutilé" (poignets, bras, poignets et bras). Des événements passés traumatisants comme le harcèlement, l'abus sexuel ou physique ou l'abandon par les parents sont plus fréquents dans les groupes "auto-mutilé".

Plus encore, Vassalini *et al.* (2014) déclarent que 32.1% des cas étudiés ont été diagnostiqués d'un trouble psychiatrique. Pour les 67.9% restants, le diagnostic n'était pas disponible. Des épisodes dépressifs sont les plus probables, suivis de schizophrénie. Karger *et al.* (2000) montrent que la majorité des 65 cas de suicide par sectionnement étudiés ont un passé psychiatrique.

1.3 Prévention du suicide

Dans cette section, nous passons en revue les méthodes de prévention à ce jour, avec un accent sur les milieux carcéraux.

1.3.1 Symptômes

Les victimes peuvent déclarer les symptômes suivants :

- attente d'un jugement ou condamné à une peine longue (Kaster, Martin & Simpson, 2017),
- en conflit avec d'autres prisonniers ou avec le personnel (Kaster *et al.*, 2017),
- stress, isolement, dépression, abus physique, abus de drogues (Marzano, Hawton, Rivlin, Smith, Piper & Fazel, 2016),
- passé d'auto-mutilation (Cramer, Wechsler, Miller & Yenne, 2017).

1.3.2 Mesures de prévention en prison

Cramer *et al.* (2017) et Marzano *et al.* (2016) démontrent l'efficacité d'une politique de prévention promouvant des contacts sociaux pour les détenus. Cela encourage la création de communautés, par exemple des "pseudo-familles", dans les milieux correctionnels.

Par ailleurs, les méthodes suivantes ont démontré leur efficacité :

- évaluation régulière du risque de suicide,
- préparation à une crise,
- interactions positives entre incarcérés et personnel,
- formation annuelle pour le personnel médical,
- autopsie psychologique en cas de suicide afin d'améliorer la recherche et les efforts de prévention.

Les recommandations de Hayes (2013) sont en adéquation avec les méthodes précédentes. Cependant, l'accent est mis sur un renforcement de l'effectif du personnel et la fréquence des évaluations psychologiques.

1.3.3 Prévention et outils de détection

Hayes (2013) présente des outils technologiques de prévention.

Des capteurs de force au niveau du sol devraient lancer une alarme dès lors que le poids du prisonnier quitte le sol trop longtemps. En théorie, cela voudrait dire qu'il se pend. Mais en pratique, le poids du prisonnier n'est pas toujours totalement soutenu. Qui plus est, ce système ne permet pas de détecter d'autres types de suicide.

Une autre initiative consiste à porter un capteur de signes vitaux. Mais ce dispositif ne s'est jamais démocratisé à cause de sa nature intrusive. Des radars de signes vitaux sont en cours de développement et sont peut-être la prochaine révolution de la détection de suicides en temps réel (Gagnon, 2016).

Dans un autre registre, une proposition serait d'utiliser des tissus indéchirables pour les vêtements et les draps.

La méthode la plus utilisée reste la surveillance par télévision en circuit fermé. Mais cela requiert du personnel et n'est pas particulièrement efficace (Ainsworth, 2002).

1.3.4 Modèles d'apprentissage prédictifs

Dans cette section, nous étudions des efforts de recherche utilisant des données sociodémographiques et des réponses à des échelles psychiatriques pour prédire des tentatives de suicide.

Delgado-Gomez, Blasco-Fontecilla, Sukno, Ramos-Plasencia & Baca-Garcia (2012) utilisent deux échelles psychiatriques sur 883 adultes, dont 347 ayant tenté de se suicider. Différents

algorithmes d'apprentissage machine sont déployés (régression linéaire, arbres de décision, LARS et SVM). L'objectif est de faire la distinction entre les deux groupes. La meilleure performance est de 83,6% sur l'ensemble de test. Ces travaux fournissent un outil clinique avec des taux de prédiction raisonnables.

Des études plus récentes ont amélioré ces résultats. Oh, Yun, Hwang & Chae (2017) étudient l'importance de 41 caractéristiques (31 provenant d'échelles psychiatriques et 10 caractéristiques sociodémographiques) pour la prédiction d'une tentative de suicide chez des patients avec des troubles de l'anxiété. Les 573 participants ont rempli un questionnaire ainsi que leur historique de tentatives. Un réseau de neurones obtient au mieux 93,7% d'exactitude sur une fenêtre d'erreur d'un mois. Les résultats sont encourageants, mais il est à noter que le réseau prédit des tentatives passées, et non futures.

Une autre expérience dirigée par Barros, Morales, Echávarri, García, Ortega, Asahi, Moya, Fischman, Maino & Núñez (2017) extrait plus de 300 variables provenant de 5 questionnaires psychiatriques et en isole 22. Les participants ont été catégorisés en deux groupes : "suicide" (n=349) et "non-suicide" (n=358). Le meilleur modèle est un SVM et obtient 78% d'exactitude.

Walsh, Ribeiro & Franklin (2017) appliquent l'apprentissage machine sur 5 167 dossiers électroniques dont 3 250 ont au moins une tentative de suicide. Les meilleurs résultats sont de 79%, mais avec une fenêtre d'erreur de seulement 7 jours.

Ces études pionnières illustrent le potentiel de l'apprentissage machine pour la prédiction de suicides à court et moyen terme. Mais en pratique, les modèles ne sont pas assez performants pour être utilisé comme unique outil. Qui plus est, les études sont majoritairement rétroactives. Cela justifie et motive la recherche en détection temps réel comme solution de dernier recours.

1.4 Apprentissage machine pour la reconnaissance d'activités

Une revue de littérature concernant la reconnaissance d'activités humaines est développée chapitre 3. Afin d'éviter d'alourdir inutilement le document, nous profitons plutôt des sections

suivantes pour établir le cadre théorique souvent considéré comme acquis dans les articles actuels.

La reconnaissance d'activités humaines est organisée selon le cadre suivant. Des séquences temporelles (format vidéo ou série temporelle de points 3D) sont prédécoupées. Chaque séquence contient une action manuellement identifiée. Le travail de recherche consiste donc à proposer une architecture permettant de reconnaître l'action et de la classifier correctement.

Les caméras RGB+D proposent quatre flux de données différents (vidéo RGB, infrarouge, squelette 3D et profondeur). À l'exception de l'infrarouge, ces flux ont été largement utilisés comme données en entrée de réseaux de neurones. Tout particulièrement, le squelette 3D est de faible dimensionnalité et présente un pouvoir de représentation intéressant.

La reconnaissance d'activités diffère de la reconnaissance anticipée et de la détection *online*. En effet, la reconnaissance étudie une séquence dans son ensemble. La reconnaissance anticipée est similaire avec uniquement une sous-portion de l'action. La détection *online* traite des séquences longues contenant plusieurs actions, image par image, sans accès à des informations du futur. La fin ou la durée d'une action sont des informations du futur, par exemple.

1.4.1 Réseaux de neurones convolutifs

La démocratisation de l'apprentissage profond n'a eu lieu qu'en 2012. L'exploitation d'architectures de calcul permettant d'accélérer les temps d'apprentissage a permis de lancer une forte dynamique de recherche dans ce domaine. Pourtant, le premier article scientifique utilisant une architecture moderne de réseau convolutif est un travail du millénaire précédent de LeCun, Bottou, Bengio & Haffner (1998) schématisé Figure 1.1.

ImageNet (Deng, Dong, Socher, Li, Li & Fei-Fei, 2009) est une base de données de plus d'un million d'images de classes variées. Depuis 2010, un challenge est organisé autour de celle-ci. La métrique d'évaluation est l'exactitude. En 2012, pour la première fois, une architecture profonde convolutive remporte le concours (Figure 1.2). Elle obtient une erreur top-5 de 15,3%, soit 10%

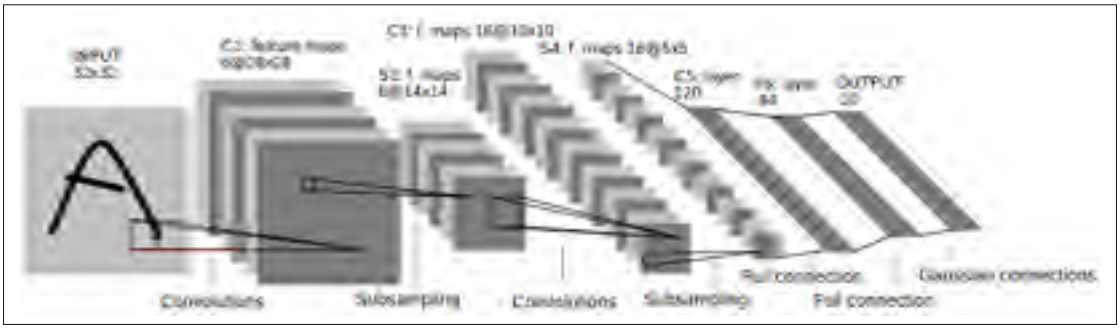


Figure 1.1 *LeNet*, première architecture moderne de réseau convolutif
Tirée de LeCun *et al.* (1998)

de mieux que le second. L'architecture, *AlexNet*, a été publiée par Krizhevsky, Sutskever & Hinton (2012).

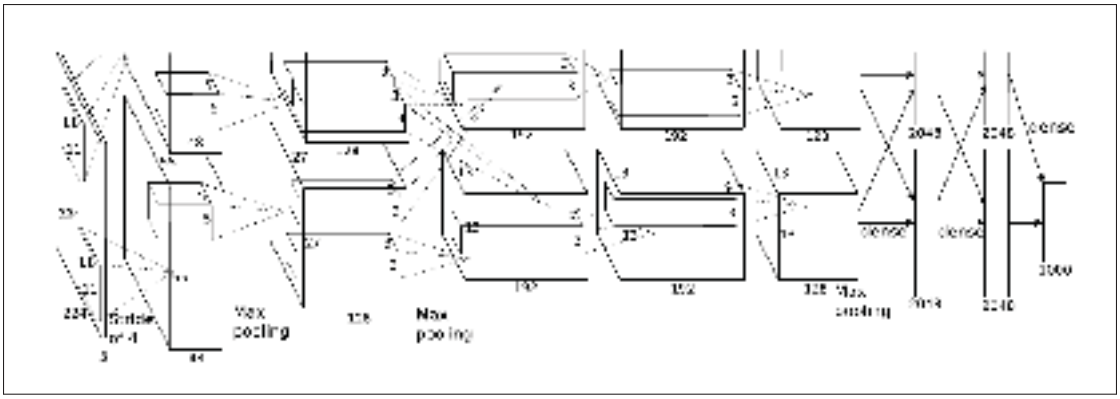


Figure 1.2 *AlexNet*, le réseau qui démocratisa l'apprentissage profond
Tirée de Krizhevsky *et al.* (2012)

À première vue, les architectures de *LeNet* et *AlexNet* sont très similaires. Les différences se situent au niveau de la profondeur et, par conséquent, du nombre des paramètres à optimiser. La méthode de sous-échantillonnage est également différente. *LeNet* moyenne, alors que *AlexNet* conserve la valeur maximale (*average pooling* vs. *maximum pooling*). La fonction d'activation de chaque couche profonde est également différente. *LeNet* utilise la fonction sigmoïde alors que *AlexNet* utilise la fonction *Rectified Linear Unit*.

Un réseau de neurones convolutif profond est généralement composé des blocs suivants, peu importe la dimension de l'entrée étudiée :

- couche de convolution, soit différents noyaux de convolution qui interprètent les données en entrée ;
- couche de sous-échantillonnage qui compresse l'information ;
- couche de correction par une fonction d'activation généralement non linéaire. Sinon, on peut montrer qu'un réseau multicouche est équivalent à un réseau monocouche.

Un réseau va converger vers un objectif dicté par une fonction de perte. Pour une tâche de classification, la fonction de perte par entropie croisée *softmax* est largement utilisée. Elle permet à la dernière couche d'un réseau de neurones de converger vers une distribution de probabilités de K classes, en fonction de l'entrée.

Pour un vecteur $\mathbf{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K$, la fonction *softmax* est définie ainsi :

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \text{pour tout } j \in 1, ..., K \tag{1.1}$$

L'apprentissage se fait de façon itérative par rétropropagation du gradient, qui peut être formalisé de la façon suivante. On note x l'entrée d'un réseau de neurones, y la sortie désirée et $\hat{y}$ la prédiction du réseau. De même $W^{\ell-1,\ell}$ la matrice de poids permettant de passer du neurone i de la couche $\ell - 1$ au neurone j de la couche l, a^ℓ la valeur des neurones de la couche ℓ après activation, g la fonction d'activation, b^ℓ les biais de la couche ℓ. Les valeurs des neurones de la couche ℓ se calculent de la façon suivante.

$$a^\ell = g(W^{\ell-1,\ell} \cdot a^{\ell-1} + b^\ell) = g(Z^\ell) \tag{1.2}$$

La propagation avant est effectuée jusqu'à obtenir la prédiction $\hat{y}$. L'erreur E entre la sortie désirée et la prédiction est ensuite calculée pour chaque neurone. Celle-ci est alors rétropropagée en chaîne jusqu'à l'entrée du réseau, ce qui permet de calculer les gradients des différents poids :

$$\Delta W^{\ell-1,\ell} = -\lambda \frac{\partial E^{\ell}}{\partial W^{\ell-1,\ell}} = -\lambda a^{\ell-1} \cdot \delta^{\ell} \tag{1.3}$$

Avec λ, un hyperparamètre définissant le taux d'apprentissage et δ^{ℓ} le signal d'erreur.

Les poids sont ensuite mis à jour à l'aide des gradients ainsi calculés :

$$W^{\ell-1,\ell} := W^{\ell-1,\ell} - \Delta W^{\ell-1,\ell} \tag{1.4}$$

D'un point de vue théorique, une opération de convolution peut être ramenée à un produit matriciel. L'algorithme de propagation peut donc fonctionner aussi bien pour des réseaux de neurones classiques que pour des réseaux convolutifs.

1.4.2 Réseaux résiduels

Les réseaux résiduels modifient l'architecture traditionnelle d'un réseau en y introduisant des sauts de connexions. C'est-à-dire que les valeurs des neurones d'une couche intermédiaire sont utilisées par la couche suivante et la couche d'après. Formellement, la propagation avant s'écrit :

$$a^{\ell} = \mathbf{g}(W^{\ell-1,\ell} \cdot a^{\ell-1} + b^{\ell} + W^{\ell-2,\ell} \cdot a^{\ell-2}) \tag{1.5}$$

$$= \mathbf{g}(Z^{\ell} + W^{\ell-2,\ell} \cdot a^{\ell-2}) \tag{1.6}$$

Les réseaux résiduels sont une solution au problème de disparition du gradient. En quelques mots, il existe un compromis entre profondeur du réseau et potentiel d'apprentissage. En effet,

la rétropropagation revient à un produit de dérivées en chaîne. On comprend bien qu'un faible gradient pour une couche a des répercussions sur les autres et ne met que faiblement à jour les poids. En introduisant des sauts de connexions, les poids sont ainsi influencés par plusieurs graphes de dérivation, plus ou moins longs. Des dynamiques d'entraînement différentes sont donc encouragées, et, empiriquement, donnent d'excellents résultats. Une connexion résiduelle est schématisée Figure 1.3.

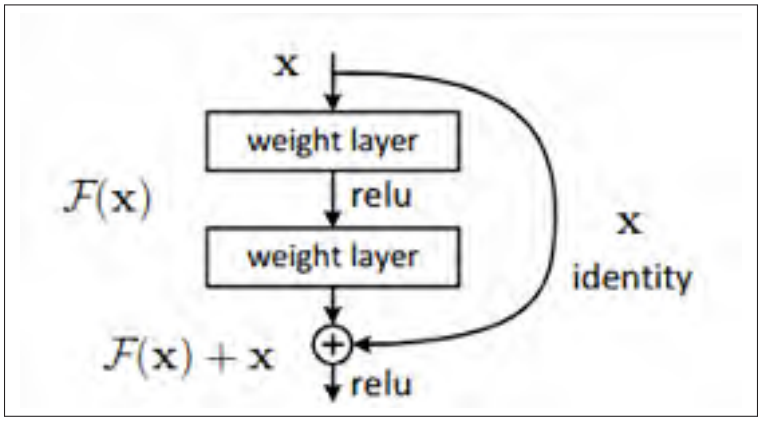


Figure 1.3 Apprentissage résiduel
Tirée de He *et al.* (2016)

Les connexions résiduelles permettent d'une part d'accélérer l'apprentissage en optimisant la mise à jour des poids. D'autre part, ces derniers sont réutilisés, ce qui diminue le nombre de neurones et a fortiori la taille totale du réseau.

Pour les performances sur *ImageNet*, He *et al.* (2016) obtiennent des résultats comparables à l'état de l'art de l'époque, avec sensiblement moins de paramètres.

Dans nos travaux, nous utilisons des modèles résiduels convolutifs 2D pour le traitement de séquences d'articulations 3D du squelette humain, et 3D pour le traitement vidéo (2 dimensions spatiales, 1 temporelle).

1.4.3 Normalisation par *batch*

La normalisation par *batch* est une technique permettant d'améliorer la vitesse d'apprentissage et de généralisation d'un réseau de neurones introduite par Ioffe & Szegedy (2015). Initialement, cette technique de normalisation a été développée dans l'espoir de conserver une distribution stable pour les valeurs des neurones des couches intermédiaires. Lors de la rétropropagation, les poids des neurones sont mis à jour et vont donc modifier la distribution des valeurs intermédiaires. Sans normalisation, le réseau est obligé d'apprendre une nouvelle distribution à chaque itération, ce qui ralentit le temps de convergence.

Formellement, à l'apprentissage, la moyenne μ et l'écart type σ d'un *batch* B de $|B|$ éléments sont calculés pour chaque couche ℓ, avec a la valeur des neurones :

$$\mu_B^\ell = \frac{1}{|B|} \sum_{i=1}^{|B|} a_i^\ell \tag{1.7}$$

$$\sigma_B^{\ell^2} = \frac{1}{|B|} \sum_{i=1}^{|B|} (a_i^\ell - \mu_B^\ell)^2 \tag{1.8}$$

On normalise ensuite les poids, avec ϵ une faible constante arbitraire, et calcule un nouveau vecteur :

$$\hat{a}^\ell = \gamma^\ell \circ \frac{a^\ell - \mu_B^\ell}{\sqrt{\sigma_B^{\ell^2} + \epsilon}} + \beta^\ell \tag{1.9}$$

Avec γ^ℓ et β^ℓ, deux tenseurs, autrement dit des tableaux multidimensionnels, appris par rétropropagation. En effet, toutes les étapes de la normalisation par *batch* sont dérivables. L'opérateur $\circ$ signifie une multiplication élément par élément.

Lors de l'inférence sur l'ensemble de test, les moyennes et écarts types ne sont plus calculés par *batch*. À la place, on utilise les statistiques de l'ensemble d'entraînement de façon à ce que la

prédiction d'un réseau soit déterministe. C'est-à-dire que les valeurs de μ^ℓ et σ^ℓ ne dépendent plus du *batch*. Ces valeurs fixes sont calculées pendant l'entraînement. Ce qui veut dire que pour une même entrée, le réseau donnera toujours le même résultat. Il est déterministe.

Dans nos travaux, des couches de normalisation par *batch* agrémentent les architectures des différents réseaux résiduels utilisés.

1.4.4 Réseaux récurrents et LSTM

Pour un réseau de neurones classique, toutes les entrées sont indépendantes entre elles. Par exemple pour un problème de classification d'images, ces dernières ne dépendent pas l'une de l'autre. Il existe des réseaux spécialement adaptés au traitement de séquences. Ils sont appelés réseaux de neurones récurrents (RNN). Par exemple, une phrase est une séquence. Chaque mot de la phrase dépend de ceux qui le précèdent et succèdent. L'intuition derrière un RNN est que celui-ci "mémorise" une information passée. Cela peut être le mot précédent, la dernière phrase, etc.

Mathématiquement, chaque cellule du réseau possède une matrice U_t de poids. Les équations suivantes résument les opérations réalisées par un RNN classique d'Elman (1990). C'est l'introduction du vecteur d'état caché h_t qui modélise les dépendances. La Figure 1.4 illustre un RNN classique. Formellement :

$$h_t = \sigma(W_h \cdot x_t + U_h \cdot h_{t-1} + b_h) \tag{1.10}$$

$$y_t = \sigma(W_y \cdot h_t + b_y). \tag{1.11}$$

Sans les termes U_t et h_{t-1}, on retrouve l'équation 1.10 une couche linéaire suivie de la fonction d'activation sigmoïde. En les ajoutant, on introduit à cela une information temporelle. Le vecteur

d'état caché h_t est généralement utilisé comme vecteur profond de caractéristiques final, comme le montre l'équation 1.11.

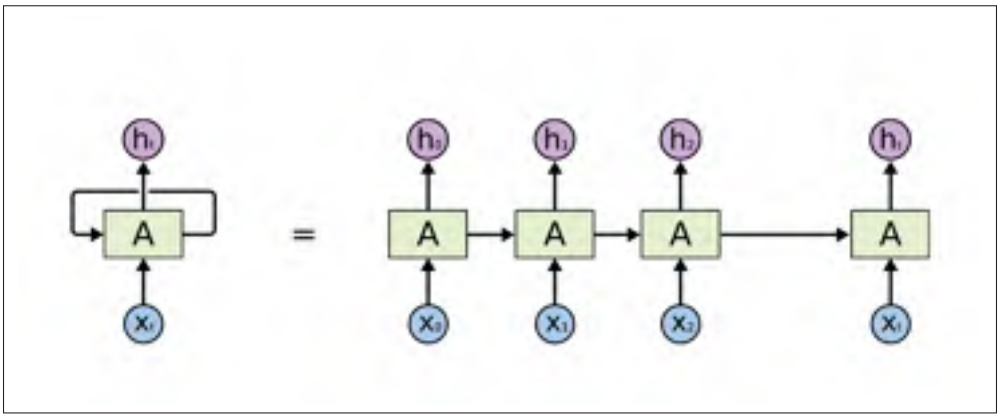


Figure 1.4 RNN classique aussi appelé réseau d'Elman
Tirée d'Olah (2015)

Hochreiter & Schmidhuber (1997) introduisent une variation modélisant des dépendances temporelles à long terme : le réseau récurrent à longue mémoire à court terme (LSTM), schématisé Figure 1.5.

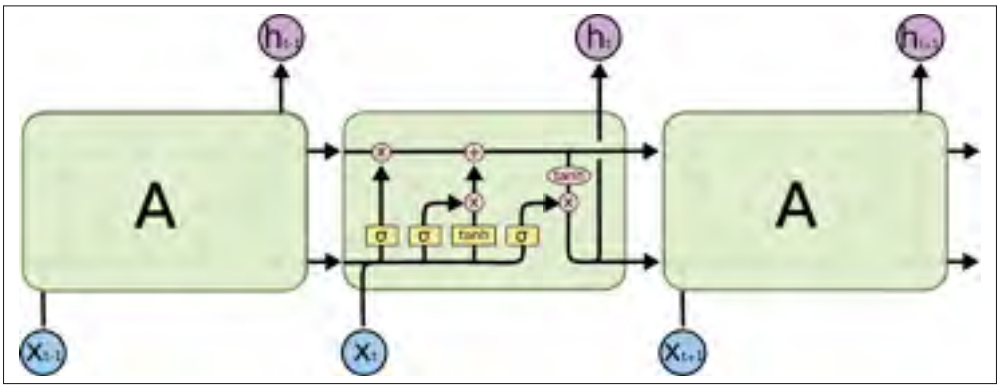


Figure 1.5 Réseau LSTM
Tirée d'Olah (2015)

La force des LSTM s'opère par l'introduction d'une cellule c_t qui joue le rôle de mémoire en plus de l'état caché h_t. Les différents modules d'une cellule LSTM sont des portes qui régulent

l'information transitée par c_t. On a une première porte servant à oublier des informations passées (*forget gate*). Pour le traitement de texte, on peut imaginer que cette porte réinitialise la cellule mémoire à chaque nouvelle phrase.

Une autre porte régule quelles informations en entrée sont mémorisées (*input gate*). Pour le traitement de texte, cela peut être le genre du sujet d'une phrase. Enfin, le réseau décide quelles informations stocker. Il reste une dernière porte pour valider la réécriture de la cellule (*output gate*).

Les équations des états internes résument l'intuition derrière les LSTM. Pour une séquence x, avec b les biais de chaque porte, W les matrices de poids, c la mémoire, h l'état caché, F, I, O les vecteurs intermédiaires issus des portes, y la sortie :

$$F_t = \sigma(W_F \cdot x_t + U_F \cdot h_{t-1} + b_F) \quad \text{(forget gate)} \tag{1.12}$$

$$I_t = \sigma(W_I \cdot x_t + U_I \cdot h_{t-1} + b_I) \quad \text{(input gate)} \tag{1.13}$$

$$O_t = \sigma(W_O \cdot x_t + U_O \cdot h_{t-1} + b_O) \quad \text{(output gate)} \tag{1.14}$$

$$c_t = F_t \circ c_{t-1} + I_t \circ \tanh(W_c \cdot x_t + U_c \cdot h_{t-1} + b_c) \tag{1.15}$$

$$h_t = O_t \circ \tanh(c_t) \tag{1.16}$$

$$y_t = g(W_y \cdot h_t + b_y) \tag{1.17}$$

À partir du vecteur d'état caché du pas de temps précédent h_{t-1}, on calcule les vecteurs F_t (équation 1.12), I_t (équation 1.13) et O_t (équation 1.14). La fonction sigmoïde restreint les valeurs de chaque élément entre $[0, 1]$. Ensuite, F_t va conditionner la mémoire c_{t-1} (équation 1.15) et I_t l'entrée (équation 1.15), qui sont sommées élément par élément pour calculer c_t. Le vecteur d'état caché h_t est calculé à partir de c_t et conditionné par O_t (équation 1.16). Enfin, comme pour un réseau d'Elman, le vecteur h_t est utilisé pour une inférence finale, avec g une fonction d'activation.

Dans nos travaux, nous utilisons des réseaux récurrents LSTM pour de la reconnaissance anticipée d'activités humaines *online* et pour de la détection d'activités *online* (chapitre 5).

1.5 Reconnaissance anticipée d'activités humaines

La reconnaissance anticipée d'activités humaines est un domaine de recherche similaire à la reconnaissance d'activités traditionnelle. Seule différence, une portion de la séquence réelle est maintenant utilisée, et non la séquence entière. L'objectif est donc de classifier correctement sans avoir le contexte global de l'action. La plupart des travaux s'intéressent à la reconnaissance de séquences entières. Ce domaine étant maintenant bien documenté, ces dernières années ont vu l'apparition d'architectures spécifiques à la reconnaissance anticipée d'activités humaines.

Ke, Liu, Bennamoun, Rahmani, An, Sohel & Boussaid (2018) confrontent séquence partielle à séquence entière lors de l'entraînement. Un module de régularisation encourage le réseau étudiant la séquence partielle à émettre un vecteur de caractéristiques similaire à celui généré par le réseau étudiant la séquence entière. Ke, Bennamoun, Rahmani, An, Sohel & Boussaid (2019) développent plus tard ce paradigme et obtiennent de bien meilleurs résultats.

Wang, Hu, Lai, Zhang & Zheng (2019) distillent l'information apprise par un RNN bidirectionnel "professeur" vers un RNN unidirectionnel "étudiant". De même, le professeur apprend une représentation globale que l'étudiant est encouragé à répliquer depuis un contexte local. L'approche est d'autant plus intéressante qu'elle pourrait être déployée pour de la détection *online* d'activités en modifiant le prétraitement des données. L'architecture du réseau est résumée à la Figure 1.6. L'erreur quadratique entre les vecteurs d'état caché de l'étudiant et du professeur est minimisée pendant l'entraînement, ainsi que la divergence maximale moyenne de l'ensemble de ces vecteurs. L'ensemble des vecteurs est utilisé pour la prédiction.

Pang, Wang, Hu, Zhang & Zheng (2019) développent une architecture basée autour d'un autoencodeur bidirectionnel. Les résultats sont comparables à ceux de Wang *et al.* (2019).

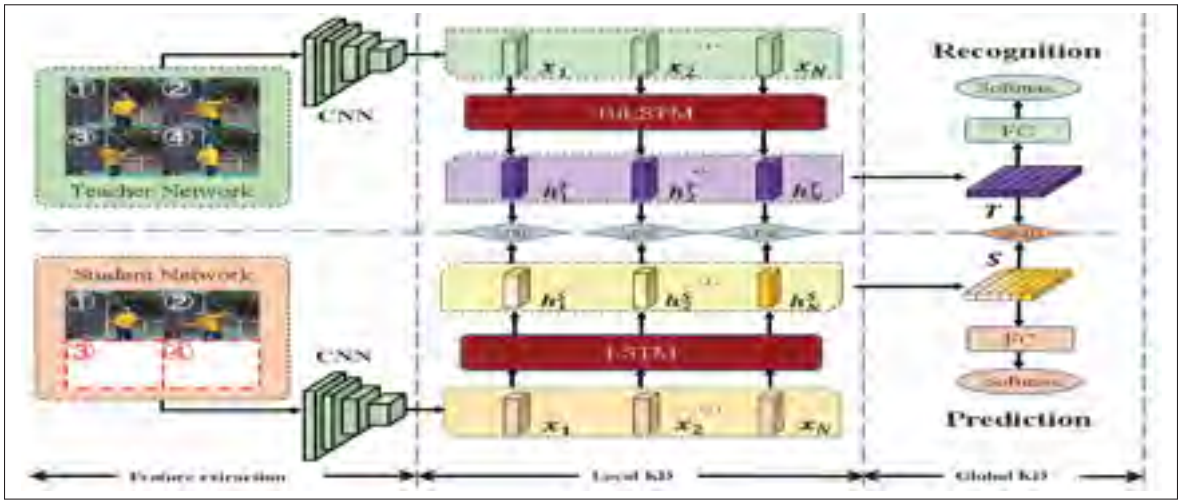


Figure 1.6 Apprentissage distillé "professeur-étudiant". KD : *Knowledge Distillation*, MSE : *Mean Squared Error*, MMD : *Maximum Mean Discrepancy*, FC : *Fully connected*, T : *Teacher*, S : *Student*
Tirée de Wang *et al.* (2019)

Une remarque et l'encouragement d'un nouveau protocole d'évaluation peuvent être émis au niveau de la méthodologie de ces travaux. En effet, les séquences sont découpées en un nombre fixe de sous-blocs afin de normaliser le format d'entrée des réseaux. Un nombre généralement choisi est $N = 40$ blocs. Seulement, certaines séquences font moins de 20 images vidéo, amenant à une redondance de l'information. Par ailleurs, diviser les séquences implique de connaître au préalable sa durée totale. De fait, ces architectures ne sont pas *online*. Nous proposons chapitre 5 un cadre permettant de distiller l'information d'un réseau professeur *offline* à un réseau étudiant *online* dans un contexte de reconnaissance anticipée.

1.6 Détection d'activités humaines par apprentissage machine

La détection d'activités humaines est une branche de recherche qui s'intéresse à des séquences temporelles non segmentées. Plusieurs actions d'intérêt peuvent avoir lieu au sein d'une unique séquence. Les temps de transition où le sujet est inactif sont considérés comme une action neutre, ou une absence d'action. Ainsi, les différences par rapport à la reconnaissance d'activités résident dans la détection temporelle et la multitude d'actions à détecter.

De ce cadre de recherche découlent plusieurs sous-domaines. La détection *offline* analyse une séquence en entier avant de proposer des segments temporels et de les classifier. La reconnaissance anticipée peut être intimement liée à la détection d'activités humaines *online*. Contrairement au contexte *offline*, une séquence doit être classifiée au fur et à mesure qu'elle apparaît, sans avoir accès aux informations du futur. L'architecture doit donc être capable de proposer des segments temporels contenant une action et les classifier sans avoir vu la séquence dans son ensemble. Cette dernière branche de recherche s'inscrit dans une volonté d'utiliser un réseau profond sur des flux d'information en temps réel.

1.6.1 Détection d'activités humaines *offline*

Les premières approches de détection d'activités implémentent une fenêtre glissante sur une séquence brute, dont sont extraites des caractéristiques, puis un SVM classifie chaque fenêtre (Ni, Yang & Gao, 2016 ; Wang, Qiao & Tang, 2014 ; Yuan, Ni, Yang & Kassim, 2016). Les caractéristiques sont soit définies à la main (Wang & Schmid, 2013), soit extraites à partir d'un réseau de neurones préentraîné. Une limite réside dans le choix de la taille de la fenêtre temporelle, toutes les actions n'étant pas de même durée. Pour pallier ce problème, Yuan *et al.* (2016) sous-échantillonnent des caractéristiques sur différentes échelles temporelles. Cependant, les temps de calcul deviennent conséquents et la détection nécessite un post-traitement important.

Un parallèle peut être fait entre la détection d'objets dans une image et la détection d'activités *offline*. Une famille d'architectures divise la détection d'objets en deux tâches (Girshick, Donahue, Darrell & Malik, 2014). La première consiste à proposer des régions d'intérêt. La deuxième implique d'évaluer et classifier ces régions. On appelle cette famille les *Region*-CNN (R-CNN). Ont ensuite été introduits *Fast* R-CNN par Girshick (2015) puis *Faster* R-CNN par Ren, He, Girshick & Sun (2015). Les différentes itérations laissent toujours plus d'autonomie au réseau et améliorent considérablement les temps de calcul.

De nombreuses méthodes de détection d'activités suivent cette approche. Caba Heilbron, Carlos Niebles & Ghanem (2016) présentent une architecture proposant différents segments

contenant une action à l'aide d'un dictionnaire de caractéristiques. Un sous-ensemble de candidats est finalement conservé à l'aide de ces caractéristiques. Escorcia, Heilbron, Niebles & Ghanem (2016) utilisent une architecture profonde pour générer des candidats de tailles différentes. Des progrès sont faits au niveau des performances et des temps de calcul. Cependant les candidats restent limités par les tailles des fenêtres proposées. Il en est de même pour les travaux de Shou, Wang & Chang (2016). Un effort de classification est réalisé, mais à nouveau des fenêtres de tailles variables sont utilisées. Heilbron, Barrios, Escorcia & Ghanem (2017) utilisent le contexte sémantique d'une action pour améliorer la détection et la classification d'un segment temporel (Figure 1.7). Par exemple, la présence d'un chien est un excellent indice pour l'action "promener son chien". Les approches sémantiques permettent de donner une forte compréhensibilité aux architectures. Le premier bloc du réseau propose des segments temporels. Ceux-ci sont rejetés ou retenus. Les segments retenus sont passés en entrée d'un encodeur sémantique. Ce dernier évalue la scène et les objets présents. À l'aide de connaissances a priori, le segment est évalué et associé à une classe. Enfin, les propositions sont réévaluées.

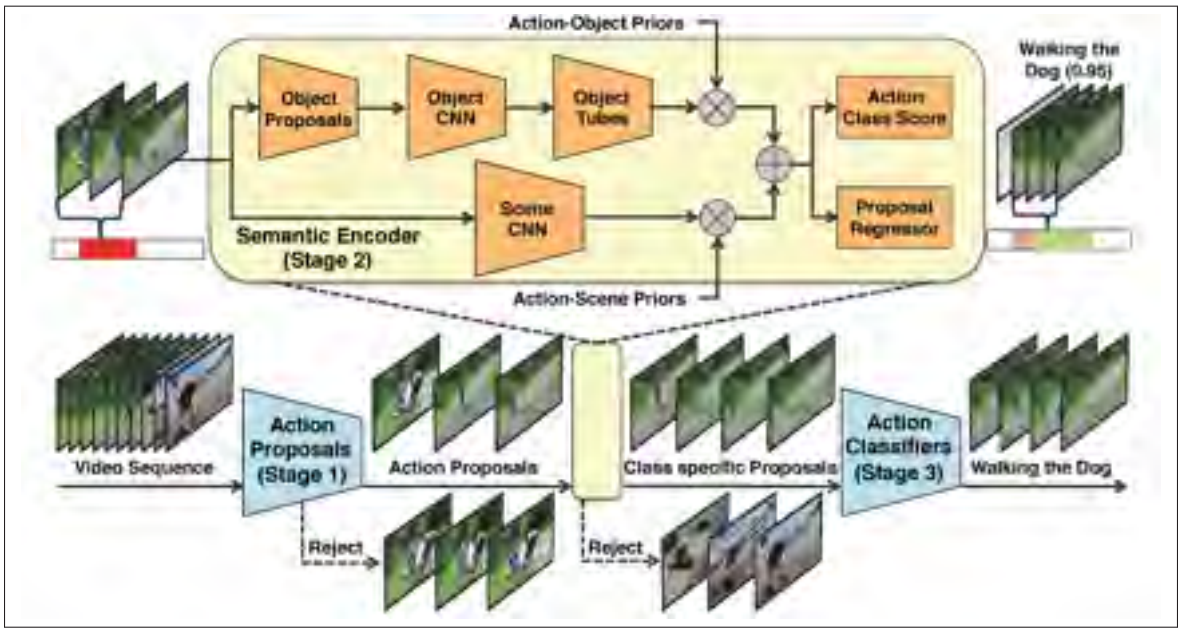


Figure 1.7 Modèle de détection d'activités en cascade à l'aide d'indices sémantiques
Tirée de Heilbron *et al.* (2017)

D'autres efforts s'inscrivent dans les améliorations proposées par l'architecture *Faster* R-CNN. Les architectures de (Dai, Singh, Zhang, Davis & Qiu Chen, 2017 ; Gao, Yang, Chen, Sun & Nevatia, 2017a ; Gao, Yang & Nevatia, 2017b ; Xu, Das & Saenko, 2017) sont maintenant entraînables de bout en bout, plus rapides et avec de meilleures performances. L'architecture *Temporal Action Localization* de (Chao, Vijayanarasimhan, Seybold, Ross, Deng & Sukthankar, 2018) est actuellement la plus performante. Cependant, cette architecture n'est pas déployable en temps réel, puisqu'elle demande d'analyser la séquence en son ensemble. Cela revient à utiliser des informations du futur.

À l'instar de la famille *You Only Look Once* (YOLO) (Redmon, Divvala, Girshick & Farhadi, 2016) pour la détection d'objets, certaines approches génèrent un unique vecteur de caractéristiques par image, ou groupe d'images. Ainsi, chaque image n'est étudiée qu'une seule fois. Buch, Escorcia, Shen, Ghanem & Carlos Niebles (2017b) présentent un réseau mélangeant CNN 3D et RNN pour la proposition de segments temporels. L'architecture, illustrée Figure 1.8, intègre ensuite la classification des segments conjointement (Buch, Escorcia, Ghanem, Fei-Fei & Niebles, 2017a). Une vidéo est découpée en fenêtres adjacentes de tailles δ. Un CNN 3D préentraîné joue le rôle d'encodeur visuel et émet un vecteur de caractéristiques pour chaque fenêtre. Deux RNN prennent en entrée ces vecteurs et sont contraints sémantiquement pendant l'entraînement. Le premier se focalise sur la proposition de segments, le deuxième sur la classification. Les vecteurs d'état caché des deux RNN sont concaténés et proposent des segments rétroactifs de tailles variables, avec une classe associée. On peut reprocher à ce réseau sa complexité et le nombre important de segments proposés, ainsi qu'une taille maximale. Nous proposons chapitre 5 un réseau de détection d'activités humaines *online* similaire, mais en résolvant les problèmes relevés.

Lin, Zhao & Shou (2017) s'intègrent dans cette lignée en favorisant des convolutions temporelles plutôt que des réseaux récurrents. Zhang, Dai, Wang & Wang (2018) réalisent la proposition et la classification simultanément à l'aide d'un CNN 3D entièrement convolutif. Cependant, le réseau prend en entrée une séquence de taille fixe et n'est pas transposable dans un contexte temps réel.

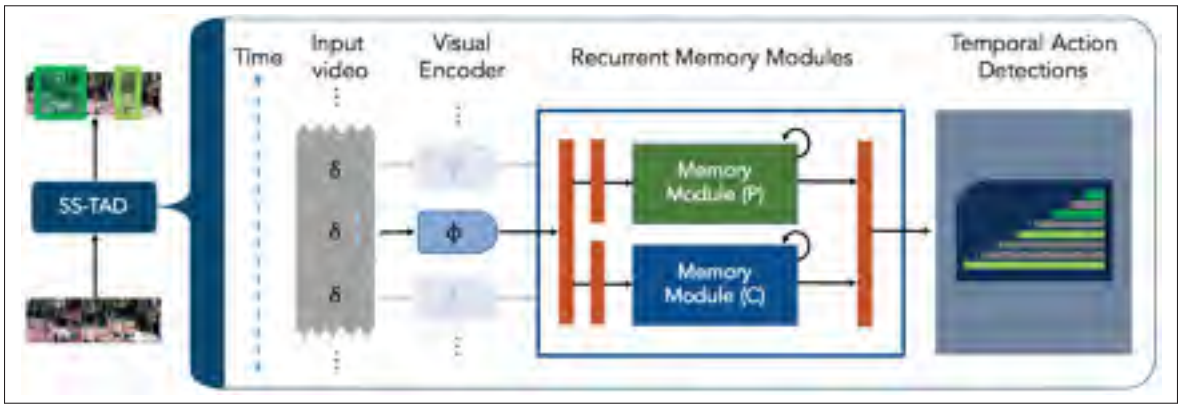


Figure 1.8 Modèle de détection d'activités à flux unique. SS-TAD : *Single Stream-Temporal Action Detection*. P : Proposal. C : Classification
Tirée de Buch *et al.* (2017a)

1.6.2 Détection d'activités humaines *online*

La détection d'activités humaines *online* consiste à détecter et classifier des actions au fur et à mesure qu'une séquence se déroule. Ainsi, à chaque instant, seuls le présent et le passé sont disponibles. Le contexte futur, donnant une vision globale de la séquence, n'est pas disponible. Les méthodes *offline* ne sont donc pas déployables puisqu'elles étudient des séquences dans leur entièreté. Certaines méthodes de reconnaissance anticipées d'activités se confondent bien avec la détection *online*.

Ma, Sigal & Sclaroff (2016) proposent une combinaison CNN + RNN. À chaque image de la vidéo, le réseau émet donc une prédiction. Une fonction de perte favorise une confiance strictement croissante au fur et à mesure que se déroule l'action. Cela améliore les performances et les fiabilise. Dave, Russakovsky & Ramanan (2017) réalisent une prédiction présente et future pour chaque image de la vidéo. À l'entraînement, la prédiction future est confrontée à l'image suivante.

Formellement, la détection *online* d'activités est un domaine relativement récent. De Geest, Gavves, Ghodrati, Li, Snoek & Tuytelaars (2016) établissent ses enjeux et proposent une base de données axée autour de cette thématique. Ils proposent ensuite un réseau à deux flux

(De Geest & Tuytelaars, 2018). Le premier s'intéresse aux caractéristiques locales, le deuxième s'intéresse aux globales. Gao, Yang & Nevatia (2017c) proposent d'utiliser des caractéristiques extraites d'un CNN 3D préentraîné sur des blocs adjacents d'une dizaine d'images vidéo. Elles sont utilisées en entrée d'un autoencodeur récurrent. Le réseau est utilisé pour de la prédiction anticipée, mais peut aussi bien être utilisé pour de la détection *online*. Le tout est entraîné par renforcement. Shou, Pan, Chan, Miyazawa, Mansour, Vetro, Nieto & Chang (2018) se concentrent sur une détection précise du début d'une action à l'aide de réseaux génératifs.

Xu, Gao, Chen, Davis & Crandall (2019) proposent une architecture récurrente, Figure 1.9, modélisant les dépendances spatiotemporelles et avec un module de prédiction dans le futur, ce qui a pour heureuse conséquence d'améliorer les prédictions présentes. À chaque pas de temps t, le vecteur d'état caché h_t de la cellule *Temporal Recurrent Network* (TRN) est utilisé en entrée d'un décodeur temporel. Ce dernier émet des prédictions pour les classes futures. Il est composé d'un LSTM, qui prend en entrée un vecteur d'état caché interne $\tilde{h}_t^{l_d-1}$ et la prédiction future précédente $\tilde{r}_t^{l_d-1}$. Les vecteurs d'état caché internes sont accumulés pour émettre un vecteur de caractéristiques futures $\tilde{x}_t$. Le vecteur $\tilde{x}_t$ et le vecteur issu de l'extracteur de caractéristiques x_t sont concaténés, et utilisés comme entrée du module *Spatiotemporal Accumulator* (STA), qui est un LSTM.

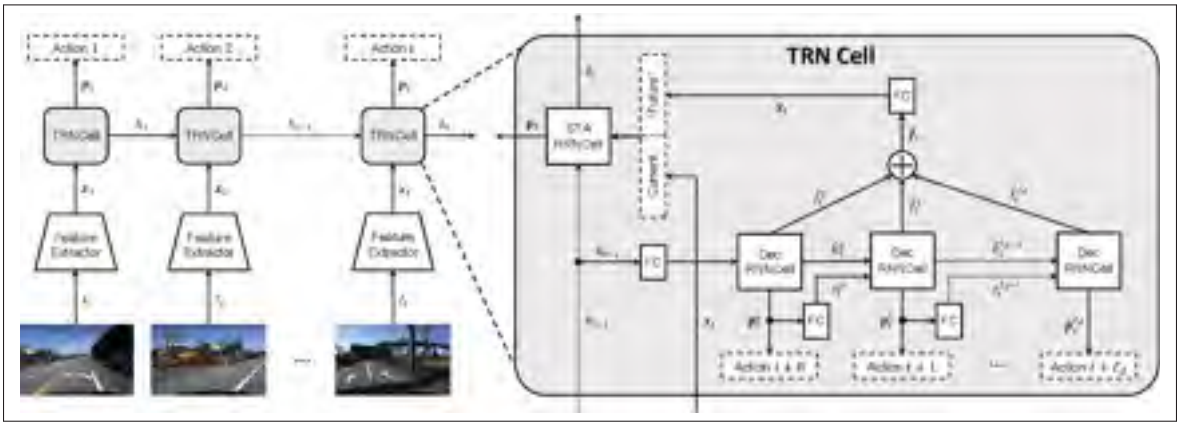


Figure 1.9 Modèle récurrent temporel prédictif. TRN : *Temporal Recurrent Network*. STA : *Spatiotemporal Accumulator*. FC : *Fully connected*
Tirée de Xu *et al.* (2019)

Ces différents travaux utilisent des modèles préentraînés pour l'extraction de caractéristiques, mais n'ajustent pas les poids de ceux-ci. Dans nos travaux, nous utilisons des architectures similaires, mais avec une attention particulière portée à l'entraînement. Nous proposons chapitre 5 une architecture réutilisant notre module infrarouge présenté chapitre 3 pour la détection d'activités *online*.

1.7 Résumé

Les taux de suicide sont plus importants en prison par rapport à la population générale (Fazel *et al.*, 2011). La méthode la plus courante est la pendaison, 19 fois plus probable d'être fatale que le sectionnement (McKee, 1998). Ce dernier, moins courant, reste problématique.

Il existe trois types de plaies auto-infligées : le poignardement, l'incision et l'hésitation (Krywanczyk & Shapiro, 2015). Elles sont principalement dirigées sur le haut du corps (Brunel *et al.*, 2010 ; Karlsson, 1998 ; Krywanczyk & Shapiro, 2015 ; Racette *et al.*, 2008 ; Vassalini *et al.*, 2014). Lors d'une tentative, on en compte plus d'une dans au moins 66% des cas (Brunel *et al.*, 2010 ; Karger *et al.*, 2000 ; Vassalini *et al.*, 2014). Les outils utilisés sont principalement, des rasoirs et des couteaux (Ersen *et al.*, 2017 ; Fujioka *et al.*, 2012 ; Karger *et al.*, 2000 ; Karlsson, 1998 ; Vassalini *et al.*, 2014).

Les méthodes de prévention efficaces misent sur la création d'une communauté sociale pour les internes et la formation d'un personnel psychiatrique (Cramer *et al.*, 2017 ; Hayes, 2013 ; Marzano *et al.*, 2016). En revanche, aucun outil de détection temps réel ne s'est démocratisé (Hayes, 2013).

Des méthodes prédictives à l'aide de données sociodémographiques et d'échelles psychiatriques par apprentissage machine existent, mais sont insuffisantes (Barros *et al.*, 2017 ; Delgado-Gomez *et al.*, 2012 ; Oh *et al.*, 2017 ; Walsh *et al.*, 2017). Quand bien même, celles-ci ne pourraient jamais prédire l'heure exact d'une tentative, motivant le développement d'outils de détection temps réel.

La reconnaissance d'activités humaines à l'aide de caméras RGB+D est un domaine de recherche prometteur. Il n'existe pas de base de données contenant des simulations de tentatives de suicide, mais des bases de données conséquentes avec des actions du quotidien sont utilisées comme référence (Liu, Hu, Li, Song & Liu, 2017a ; Shahroudy *et al.*, 2016). Pour une application de sécurité, la vidéo infrarouge est un flux intéressant, délaissé par la littérature, en plus du squelette humain 3D.

La reconnaissance d'activités humaines étudie une séquence dans son ensemble. La reconnaissance anticipée permet d'émettre une prédiction en analysant uniquement une sous-portion. L'apprentissage distillé est largement utilisé (Ke *et al.*, 2019 ; Pang *et al.*, 2019 ; Wang *et al.*, 2019). Mais la sous-portion est tout de même étudiée dans son entièreté, ce qui nécessite de connaître à l'avance sa durée. Il s'agit donc d'un paradigme *offline*.

Parallèlement, la détection d'activités humaines traite des séquences contenant plusieurs actions, sans renseignement concernant leur durée et leur position temporelle. En détection *offline*, Chao *et al.* (2018) détiennent l'état de l'art. Mais la séquence est à nouveau étudiée dans son entièreté avant d'émettre des prédictions. En détection *online*, la séquence est évaluée au fur et à mesure qu'elle se déroule. Les architectures de référence utilisent un CNN préentraîné et un RNN (Gao *et al.*, 2017c ; Ma *et al.*, 2016 ; Xu *et al.*, 2019). Les poids du CNN sont gelés lors de l'entraînement. Les mettre à jour demande certes plus de ressources, mais pourrait mener à de meilleurs résultats.

Ces différentes branches de recherches sont étudiées séparément dans la littérature. Néanmoins, elles sont complémentaires. À notre connaissance, il n'existe pas de cadre proposant de les relier entre elles.

CHAPITRE 2

DÉMARCHE DE TRAVAIL ET ORGANISATION DU DOCUMENT

Ce chapitre présente l'organisation du travail qui va permettre de répondre aux problématiques et axes de recherche dégagés à la section 1.7. Les objectifs spécifiques sont tout d'abord détaillés puis la méthodologie oeuvrant à les réaliser. Les résultats conduisent à la rédaction d'un article de journal (chapitre 3) autour de la reconnaissance d'activités humaines à l'aide de caméras RGB+D. L'article ne couvrant pas toutes les thématiques abordées, nous développons chapitre 5 une stratégie permettant de faire le lien entre reconnaissance anticipée et détection *online* d'activités humaines dans un article de conférence.

2.1 Objectifs spécifiques

L'objectif du travail est de développer une preuve de concept permettant la détection de comportements dangereux en temps réel à l'aide de caméras RGB+D. Pour ce faire, une attention particulière est portée sur les flux de données insensibles aux conditions d'éclairage, à savoir, l'infrarouge et les coordonnées 3D des articulations du squelette humain. Aucune base de données de simulation de comportements suicidaires n'existe. À la place, nous appuyons notre preuve de concept sur de larges bases de données publiques générales (Liu *et al.*, 2017a ; Shahroudy *et al.*, 2016). Un modèle performant sur une base de données généraliste devrait, par transfert d'apprentissage, fonctionner pour une tâche plus spécifique.

La littérature scientifique s'intéresse depuis quelques années aux applications rendues possibles par les caméras RGB+D. Il existe une progression naturelle entre reconnaissance et détection d'activités en temps réel. De ce fait, nous fixons comme objectifs :

- évaluer les caméras RGB+D potentielles pour un prototype (section 2.4),
- développer une architecture profonde d'apprentissage machine pour la reconnaissance d'activités humaines, pouvant fonctionner dans l'obscurité. L'accent est mis sur des actions utilisant les mains et des objets (article de journal du chapitre 3) ;

- proposer une architecture de reconnaissance anticipée (article de conférence du chapitre 5),
- faire le lien entre reconnaissance anticipée et détection d'activités *online* par apprentissage distillé (article de conférence du chapitre 5).

2.2 Méthodologie et approche de recherche

Pour les différents domaines de recherche évoqués, il existe différentes bases de données publiques permettant aux chercheurs de comparer leurs travaux. Un article scientifique accompagne généralement ces bases de données. Celui-ci propose un protocole d'évaluation standard à la base de données. Nous suivons donc les protocoles proposés par ces articles ainsi que les métriques d'évaluation.

L'apprentissage machine moderne requiert des quantités phénoménales de données. Un paradigme existe quant à leur utilisation pour la conception et l'entraînement d'une architecture. Une base de données est divisée en trois ensembles. Un ensemble d'entraînement, généralement le plus conséquent, contient les données sur lesquelles un modèle va être entraîné. Un ensemble de validation est utilisé pour évaluer les performances du modèle sur des données qu'il n'a jamais étudiées. Il permet d'affiner les valeurs des hyperparamètres. L'ensemble de validation est également utilisé pour évaluer le compromis biais-variance. En d'autres termes, si la métrique d'évaluation de l'ensemble de validation se dégrade au cours de l'entraînement alors que sur l'ensemble d'entraînement, elle continue de progresser, le réseau surapprend. Il convient alors d'arrêter l'entraînement à ce moment-là. Enfin, un ensemble de test est utilisé en phase finale pour évaluer les performances du réseau sur de nouvelles données.

Détaillée avec plus de précision chapitres 3 et 5, la base de données utilisée pour l'élaboration de notre architecture s'appelle NTU RBG+D (Shahroudy *et al.*, 2016). Il s'agit de la plus importante à ce jour, à notre connaissance. Les acquisitions sont réalisées à l'aide d'une *Kinect 2* (Zhang, 2012). Elle contient 60 classes allant d'actions quotidiennes banales (lire, écrire) à des symptômes médicaux (vomir, tomber). Elle présente 80 dispositions différentes contenant 3 caméras, 40 intervenants pour un total de 56 880 séquences.

De même, détaillée chapitre 5, nous utilisons la base de données *Peking University Multi-Modality Dataset* (PKU-MMD) de détections d'activités humaines. À notre connaissance, il s'agit de la seule contenant des séquences infrarouges provenant d'une *Kinect 2* (Zhang, 2012). Elle compte 1,076 séquences de quelques minutes, totalisant près de 20,000 actions réparties en 51 classes, pour 66 sujets.

Pour les deux bases de données retenues, il existe deux protocoles d'évaluation. Un premier sépare l'ensemble d'entraînement et de test entre sujets : *cross-subject* (CS). Un deuxième divise les ensembles selon la caméra étudiée : *cross-view* (CV). Selon les recommandations de Shahroudy *et al.* (2016), nous échantillonnons 5% de l'ensemble d'entraînement comme ensemble de validation. Nous utilisons ce dernier pour affiner les hyperparamètres et pour arrêter l'entraînement de façon anticipée.

2.3 Choix du langage de programmation et des librairies de développement

Les codes sont développés en Python version 3. Les librairies notables utilisées sont les suivantes :

- **PyTorch :** bibliothèque d'apprentissage machine. Permet de manipuler des tenseurs (des tableaux multidimensionnels) et d'automatiser l'optimisation par descente de gradients (Paszke, Gross, Chintala, Chanan, Yang, DeVito, Lin, Desmaison, Antiga & Lerer, 2017) ;
- **NumPy :** librairie permettant d'effectuer des opérations sur des tableaux multidimensionnels,
- **h5py :** permet de créer et lire des fichiers au format binaire HDF5 (Folk, Heber, Koziol, Pourmal & Robinson, 2011). Utilisé principalement pour convertir les séquences des bases de données en un format plus facilement utilisable en Python ;
- **ffmpeg-python :** emballage logiciel permettant d'interpréter les données vidéos en Python.

L'ensemble des codes ainsi que les instructions pour répliquer les résultats sont détaillés annexe I.

2.4 Choix matériels

À ce jour, plusieurs caméras RGB+D sont envisagées pour un futur prototype. Elles présentent différents avantages et inconvénients. Deux fabricants sont retenus : *Intel* et *Microsoft*.

La famille *Intel RealSense* (Keselman *et al.*, 2017) propose des capteurs grand public. Leur force réside dans leur moindre coût et le large champ de vision de la version D435. En effet, un champ de vision proche de 90° permet de limiter les angles morts d'une pièce. Néanmoins, un défaut majeur de la famille *RealSense* est l'absence de prise en charge de l'estimation des coordonnées 3D du squelette humain du kit de développement. À notre connaissance, le seul kit de développement externe offrant cette caractéristique, *Nuitrack SDK* (Nuitrack™, 2020), s'avère peu performant et empêche l'accès au flux infrarouge simultanément.

La famille *Kinect* (Zhang, 2012) intègre des algorithmes d'estimation des coordonnées 3D en temps réel (Shotton *et al.*, 2011) dans son kit de développement. La *Kinect 2* est une référence et est utilisée par l'écrasante majorité des bases de données publiques (Wang *et al.*, 2018a). Mais sa production n'est plus assurée et elle est trop volumineuse pour être envisagée comme caméra de sécurité. En 2019, *Microsoft* sort la dernière version de la *Kinect* : la *Kinect Azure*. Véritable condensé de prouesses techniques, la précision des coordonnées 3D est imbattable. En contrepartie, les ressources minimales exigées par la caméra sont considérables et l'estimation des coordonnées nécessite une carte graphique. Actuellement, les ordinateurs embarqués avec une carte graphique puissante ne sont pas légion.

Le fait que la *Kinect Azure* requiert une carte graphique n'est pas rédhibitoire. En effet, les architectures profondes imposent également cette contrainte. La famille de micro-ordinateurs embarquant une carte graphique retenue est la gamme *Jetson* de *Nvidia*. Mais à ce jour, la *Kinect Azure* n'est pas encore compatible.

À la lumière de ces réflexions, il apparaît que la solution la plus prometteuse est d'utiliser la *Kinect Azure*. Mais les ressources nécessaires sont considérables et la faisabilité n'est pas encore assurée. Les coordonnées 3D des articulations sont une source de données puissante, mais

pas forcément nécessaire. Si seule la vidéo infrarouge est utilisée, les caméras *RealSense* sont amplement suffisantes. En effet, la résolution spatiale des flux vidéo importe peu puisqu'ils sont nécessairement redimensionnés en entrée de l'architecture profonde.

Initialement, le flux infrarouge était envisagé comme source complémentaire aux coordonnées 3D. Mais l'utiliser comme unique source d'information devient une thématique intéressante afin d'envisager différents systèmes. Le Tableau 2.1 résume les différents avantages et inconvénients des caméras retenues.

Tableau 2.1 Comparatif des caméras retenues

	RealSense D415	**RealSense D435**	**Kinect 2**	**Kinect Azure**
Champ de vision max.	63.4°x40.4°	85.2°x58°	70.6° x 60°	120°x120°
Squelette intégré	X	X	✓	✓
Compatible Jetson	✓	✓	✓	X
Prix	149$	179$	X	399$
Ressources requises	Faibles	Faibles	Faibles	Hautes

2.5 Présentation des articles

Nos travaux sont répartis en deux articles. Chapitre 3, un article de journal a été soumis à *Institute of Electrical and Electronics Engineers* (IEEE) *Access*. Chapitre 5, un article de conférence a été soumis à *European Conference on Computer Vision* (ECCV).

2.5.1 Infrared and 3D skeleton feature fusion for RGB-D action recognition

La recherche actuelle en reconnaissance d'activités humaines à l'aide de caméras RGB+D explore volontiers les flux RGB, de profondeur ou encore les articulations 3D du squelette humain. Mais la vidéo infrarouge reste inexplorée (Wang *et al.*, 2018a). Nous supposons qu'il existe trois raisons à cela. Premièrement, il existe plusieurs bases de données publiques d'activités humaines enregistrées à l'aide de caméras RGB+D, mais NTU RGB+D et PKU-MMD sont les seules à notre connaissance à proposer ce flux. Deuxièmement, les flux RGB et infrarouge ont une représentation similaire. Le flux infrarouge est cependant plus bruité et représenté en

niveaux de gris. Il est donc potentiellement moins riche en information ce qui justifie amplement un intérêt majoritaire pour la vidéo RGB. Troisièmement, la littérature scientifique n'a pas pour vocation première d'être directement applicable. Par conséquent, il est naturel que l'impossibilité d'utiliser la vidéo RGB dans le noir ne soit pas prise en compte.

Par conséquent, étudier l'infrarouge est une opportunité de proposer un nouveau candidat parfaitement viable comme le prouveront nos résultats. L'article de journal présenté au chapitre 3 propose un réseau de neurones profond utilisant les articulations 3D du squelette et le flux infrarouge. L'architecture proposée est modulaire. Un premier module étudie les séquences infrarouges et extrait un vecteur de caractéristiques. Un deuxième module étudie les séquences d'articulations 3D et extrait un autre vecteur de caractéristiques. Un troisième et dernier module concatène ces deux vecteurs et les étudie conjointement. Les résultats obtenus battent l'état de l'art sur le protocole CS, considéré comme plus difficile.

2.5.2 Bridging the gap between Human Action Recognition and Online Action Detection with knowledge distillation on infrared videos

La reconnaissance d'activités humaines n'est pas suffisante pour être déployée en temps réel. Nous proposons un cadre de travail permettant de faire le lien entre reconnaissance et détection *online*, en passant par la prédiction anticipée.

Dans l'article de conférence présenté chapitre 5, une stratégie de prétraitement des séquences infrarouges différente de celle présentée dans l'article (chapitre 3) est proposée. Celle-ci est utilisable dans un contexte *online*. Le module infrarouge présenté chapitre 3 est utilisé dans un contexte de reconnaissance anticipée *offline*. Puis les connaissances apprises sont distillées à un réseau étudiant, franchissant la frontière entre *offline* et *online*.

2.6 Présentation des annexes

Annexe I, les codes et un manuel d'utilisation pour les utiliser sont présentés. Annexe II, un tutoriel de prise en main du réseau entraîné est disponible.

CHAPITRE 3

INFRARED AND 3D SKELETON FEATURE FUSION FOR RGB-D ACTION RECOGNITION

Alban Main de Boissiere[1], Rita Noumeir[1]

[1] Département de Génie Électrique, École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Article soumis à la revue « IEEE Access », février 2020.

Abstract : For skeleton-based action recognition from depth cameras, distinguishing object-related actions with similar motions is a difficult task. The other available video streams (RGB, infrared, depth) may provide additional clues, given an appropriate feature fusion strategy. We propose a modular network combining skeleton and infrared data. A pre-trained 2D convolutional neural network (CNN) is used as a pose module to extract features from skeleton data. A pre-trained 3D CNN is used as an infrared module to extract visual features from videos. Both feature vectors are then fused and exploited conjointly using a multilayer perceptron (MLP). The 2D skeleton coordinates are used to crop a region of interest around the subjects for the infrared videos. Infrared is favored over RGB, as it is less affected by illumination conditions and usable in the dark. We are the first to combine infrared and skeleton data. We evaluate our method on the NTU RGB+D dataset, the largest dataset for human action recognition from depth cameras. We perform extensive ablation studies. In particular, we show the strong contributions of our cropping strategy, data augmentation and pre-training on action classification accuracy. We also test various feature fusion schemes. Of all, feature average on an element-wise level yields the best results. Our method achieves state-of-the-art performances on NTU RBG+D.

3.1 Introduction

Human action recognition is the task of recognizing an activity performed by one or more subjects inside a segmented sequence. Recent years have witnessed successful deep architectures (Crasto, Weinzaepfel, Alahari & Schmid, 2019 ; Du, Wang & Wang, 2015b ; Kim & Reiter,

2017 ; Shahroudy, Ng, Gong & Wang, 2017 ; Tu, Xie, Qin, Poppe, Veltkamp, Li & Yuan, 2018 ; Yan, Xiong & Lin, 2018) with promising results on benchmark datasets (Carreira & Zisserman, 2017 ; Shahroudy *et al.*, 2016).

Consumer-grade depth cameras such as Intel RealSense (Keselman *et al.*, 2017) and Microsoft Kinect (Zhang, 2012) coupled with advanced human pose estimation algorithms (Shotton *et al.*, 2011) have allowed 3D skeleton data to be obtained in real time. Key joints of the human body are extracted to a 3D space, providing a high-level representation of an action. Skeleton data are robust to surrounding environment, illumination variations and may be generalized to various viewpoints (Aggarwal & Xia, 2014 ; Han, Reily, Hoff & Zhang, 2017 ; Li, Hou, Wang & Li, 2017b,1 ; Presti & La Cascia, 2016 ; Wang *et al.*, 2018a). Earlier works have indicated that key joints are powerful descriptors for human motion (Johansson, 1973). The low dimensionality and high representation power make skeleton data a prime input for action recognition tasks.

Opening the door for new action recognition algorithms, those are broadly categorized into RGB and 3D skeleton approaches. However, it has been demonstrated that visual and skeleton inputs can work in symbiosis (Rahmani & Bennamoun, 2017). Actions with similar body motion, such as writing versus typing on a keyboard, prove difficult to classify with skeleton data only. In this respect, skeleton data might benefit from the visual clues of RGB streams.

Depth cameras offer four different data streams : RGB, depth, infrared (IR) videos and 3D skeleton. To our knowledge, infrared videos from depth cameras have never been used as an input source for action recognition. We argue that the lack of large scale datasets proposing IR videos in addition to the other streams is in part responsible. Moreover, RGB and IR images are quite similar, the former offering a richer representation of a scene therefore making it a better candidate. However, IR is usable in the dark, which is viable for security applications when skeleton data are insufficient. The recent introduction of large scale datasets like NTU RGB+D (Shahroudy *et al.*, 2016) and PKU-MMD (Liu *et al.*, 2017a) containing IR videos motivates the evaluation of methods using this stream. Video understanding is a well-studied computer vision task. But modeling spatiotemporal features and long-term dependencies remains an issue.

Another challenge in video action classification is the volume of information. To reduce the complexity of the videos, downscaling the frames is often employed but also comes with a decrease in the quality of the information. Moreover, discriminating clues may only occur in a small portion of the frames, becoming undetectable in the process (Tu, Li, Zhang, Dauwels, Li & Yuan, 2019). An alternative proposal is to focus on regions of interest. Visual attention models are capable of focusing on important cues and disregard other areas (Cho, Courville & Bengio, 2015 ; Mnih, Heess, Graves et al., 2014 ; Sharma, Kiros & Salakhutdinov, 2015).

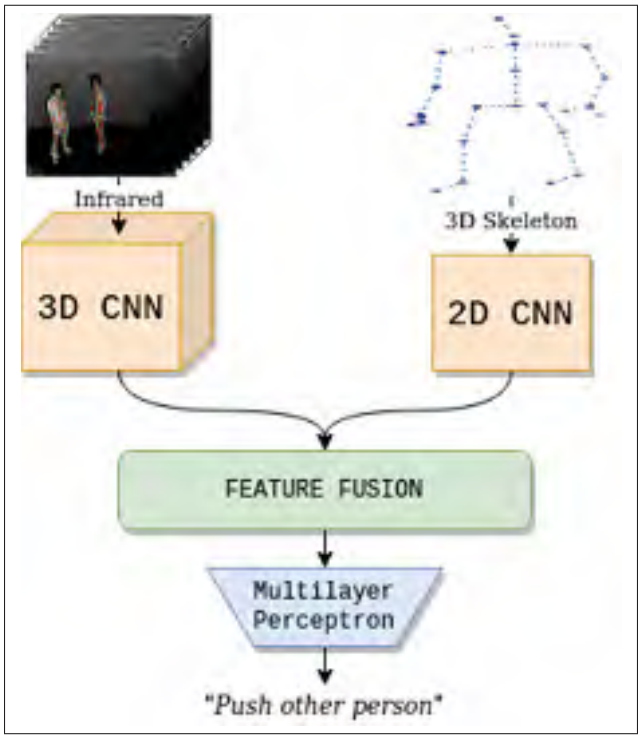


Figure 3.1 Our model uses a 2D CNN for pose data and a 3D CNN for IR sequences. Features from both modules are then fused and studied conjointly via an MLP. Training is done in end-to-end fashion

In this work, we intend to address the difficulty of differentiating actions with similar motions with an additional visual stream insensible to illumination conditions. Furthermore, we evaluate the potential of IR videos as a standalone source. We propose a model fusing video and pose data (FUSION). Pose has a double purpose. It is used as an input stream in its own right and also

conditions the IR sequences, providing a crop around the subjects, facilitating the classification. The general outline of the network is illustrated Fig. 3.1.

The pose network is an 18-layer ResNet (He *et al.*, 2016) taking as input the entire skeleton sequence. The sequence is mapped to an RGB image which is then rescaled to fit the input size of the CNN. The IR network is a ResNet (2+1)D (R(2+1)D) (Tran, Wang, Torresani, Ray, LeCun & Paluri, 2018) where a fixed number of random frames taken from evenly spaced subsequences are used as inputs. The features of each module are then fused before proposing a final classification with a multilayer perceptron (MLP).

Our main contributions are as follows.

- We demonstrate the importance of IR streams from depth cameras for human action recognition.
- We propose a fusion network taking skeleton and IR sequences as inputs. Utilizing those two steams conjointly has never been attempted before.
- We perform extensive ablation studies. We isolate different modules of our model and study their individual representation power. We also evaluate the importance of data augmentation, transfer learning, 2D-skeleton conditioned IR sequences, IR sequence length and various feature fusion strategies on the accuracy score.
- We achieve state-of-the-art results compared to methods using different streams.

Codes, documentation and supplementary materials can be found on the project page. [1]

3.2 Related Work

3.2.1 Skeleton-based approaches

Human action recognition has received a lot of attention due to its high-level representation and powerful discriminating nature. Traditional approaches focus on handcrafted features (Hussein,

[1] https://github.com/adeboissiere/FUSION-human-action-recognition

Torki, Gowayyed & El-Saban, 2013; Vemulapalli, Arrate & Chellappa, 2014; Wang, Liu, Wu & Yuan, 2012). These could be the dynamics of joint motion, the covariance matrix of joint trajectories (Hussein *et al.*, 2013) or the representation of joints in a Lie group (Vemulapalli *et al.*, 2014). Design choices prove challenging and result in suboptimal results. Recent deep-learning methods report improved accuracy. There exist three main frameworks : sequence-based models, image-based models, and graph-based models.

Sequence models exploit skeleton data as time series of key joints which are then fed to recurrent neural networks (RNN) (Du *et al.*, 2015b; Lee, Kim, Kang & Lee, 2017; Liu, Shahroudy, Xu & Wang, 2016; Shahroudy *et al.*, 2016; Song, Lan, Xing, Zeng & Liu, 2017; Wang & Wang, 2017; Zhang, Lan, Xing, Zeng, Xue & Zheng, 2017). Part-aware long short-term memory (LSTM) RNN (Shahroudy *et al.*, 2016) uses different memory cells for different regions of the body, then fuses them for the final classification. Similarly in (Du *et al.*, 2015b), a bidirectional RNN studies separate body parts individually in earlier levels and conjointly deeper on. In an effort to model simultaneously time and spatial dependencies, Liu *et al.* propose a 2D recurrent model (Liu *et al.*, 2016). Recurrent models are now part of the early deep learning efforts for skeleton-based action recognition. Vastly improving upon the results of the traditional methods, they remain insufficient. The sequence length has to be fixed during training which is not ideal and requires a sampling strategy. Moreover, sequence models tend to be much slower than their image-based counterparts.

Image models represent skeleton data as 2D images which are then used as inputs for convolutional neural networks (CNN) (Du, Fu & Wang, 2015a; Ke, Bennamoun, An, Sohel & Boussaid, 2017b; Kim & Reiter, 2017; Li, Dai, Cheng, Chen, Lin & He, 2017a; Li *et al.*, 2017b,1; Liu, Liu & Chen, 2017b; Wang, Li, Hou & Li, 2016). An intuitive method is to assign the x, y and z coordinates of a skeleton sequence to the channels of an RGB image (Du *et al.*, 2015a; Li *et al.*, 2017a). Each joint corresponds to a row and each frame to a column, or inversely. Pixel intensity is then normalized between 0 and 255 based on maximal coordinates value of the dataset (Du *et al.*, 2015a) or sequence (Li *et al.*, 2017a). Other works utilize the relative coordinates between joints to generate multiple images (Ke *et al.*, 2017b). Similarly, some works project the 3D

coordinates on orthogonal 2D planes (Li *et al.*, 2017b,1) and encode the trajectories into a hue, saturation, value (HSV) space (Wang *et al.*, 2016). A pre-trained model over ImageNet (Deng *et al.*, 2009) is leveraged. A similar approach is used in (Hou, Li, Wang & Li, 2016). More recent works focus on view-invariant transformations (Ke, An, Bennamoun, Sohel & Boussaid, 2017a ; Liu *et al.*, 2017b) or networks (Zhang, Lan, Xing, Zeng, Xue & Zheng, 2019) with improved results. In (Kim & Reiter, 2017), a temporal convolutional network is deployed with interpretability of the results as a major objective. CNNs are able to learn from entire sequences rather than sampled frames. The image generated from the skeleton sequence is resized to accommodate the fixed input shape of the CNN. This means an entire sequence can be used at once, which is an advantage compared to recurrent methods.

Graph neural networks have received a lot of attention as of late due to their effective representation of skeleton data (Xu, Hu, Leskovec & Jegelka, 2018). There exist two main graph model architectures : graph neural networks (GNN), which combine graph and recurrent networks, and graph convolutional networks (GCN), which aim to generalize traditional convolutional networks. From this architecture derives two types of GCNs : spectral and spatial. Spatial GCNs leverage the convolution operator for each node using its nearest neighbors (Simonovsky & Komodakis, 2017). Yan *et al.* (Yan *et al.*, 2018) make the best of the graph representation to learn both spatial and temporal features. Li *et al.* generalize the graph representation to actional and structural links (Li, Chen, Chen, Zhang, Wang & Tian, 2019). In (Si, Chen, Wang, Wang & Tan, 2019), a temporal attention mechanism is adopted to enhance the classification while exploring the co-occurrence relationship between spatial and temporal domains. In (Shi, Zhang, Cheng & Lu, 2019b), the length and direction of bones are used in addition to joint coordinates while adapting the topology of the graph. Shi *et al.* represent skeleton data as a directed acyclic graph based on kinematic dependencies of joints and bones (Shi, Zhang, Cheng & Lu, 2019a). GCNs report the current state-of-the-art results on benchmark datasets. However, carefully designed CNNs show comparable results (Zhang *et al.*, 2019). Also, CNNs can be pre-trained on other large scale datasets which actually improves the performances of image-based skeleton action recognition

models (Zhang *et al.*, 2019). To our knowledge, an ImageNet (Deng *et al.*, 2009) style transfer learning is impractical for GCNs.

3.2.2 RGB-based video classification

Traditional approaches focus on handcrafted features in the form of spatiotemporal interest points. Among those, improved Dense Trajectories (iDT) (Wang & Schmid, 2013), which uses estimated camera movements for feature correction, is considered the state of the art. After the widespread use of deep learning on single images, many attempts have been made to propose benchmarks for video classification.

Soon after (Wang & Schmid, 2013), two breakthrough papers (Karpathy, Toderici, Shetty, Leung, Sukthankar & Fei-Fei, 2014 ; Simonyan & Zisserman, 2014) would form the backbone of future efforts. In (Karpathy *et al.*, 2014), Karpathy *et al.* explore different ways of fusing temporal information using pre-trained 2D CNNs. In (Simonyan & Zisserman, 2014), handcrafted features, in the form of optical flow, are used symbiotically with the raw video. Two parallel networks compute spatial and temporal features. A few drawbacks include the inability to effectively capture long-range temporal information and the heavy calculations required to compute optical flow.

Later research propositions fall into five frameworks.

- 2D CNN followed by RNN network (Donahue, Anne Hendricks, Guadarrama, Rohrbach, Venugopalan, Saenko & Darrell, 2015).
- 3D CNN (Crasto *et al.*, 2019 ; Tran, Bourdev, Fergus, Torresani & Paluri, 2015 ; Yao, Torabi, Cho, Ballas, Pal, Larochelle & Courville, 2015).
- Two-Stream 2D CNN (Feichtenhofer, Pinz & Zisserman, 2016).
- 3D-Fused Two-Stream (Feichtenhofer *et al.*, 2016).
- Two-Stream 3D CNN (Carreira & Zisserman, 2017 ; Tu *et al.*, 2018).

Heavy networks and computations of handcrafted features, as well as the absence of a benchmark for long-term temporal features, remain an issue. In (Tran *et al.*, 2018), Tran *et al.* explore different forms of spatiotemporal convolutions and their impact on video understanding. A (2+1)D convolution block separating spatial and temporal filters allows for a greater nonlinearity compared to a standard 3D block with an equivalent number of parameters, as illustrated Fig. 3.2. Separating convolutions yields state-of-the-art results on benchmark datasets such as Sports-1M (Karpathy *et al.*, 2014), Kinetics (Carreira & Zisserman, 2017), UCF101 (Soomro, Zamir & Shah, 2012) and HMDB51 (Kuehne, Jhuang, Garrote, Poggio & Serre, 2011).

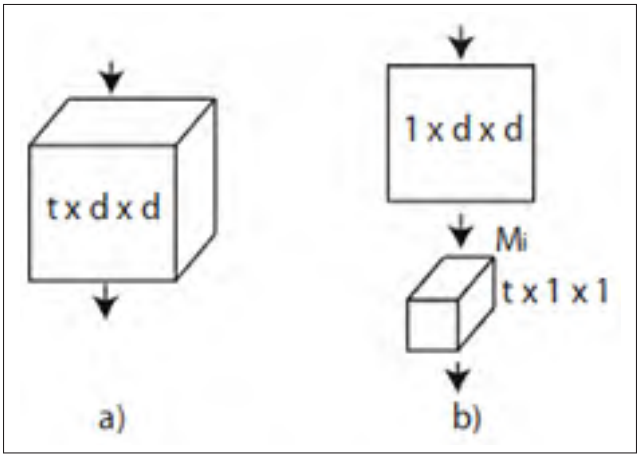


Figure 3.2 a) A standard 3D convolution operator. b) A factorized (2+1)D convolution operation with an additional non-linear activation function in between Taken from Tran *et al.* (2018)

3.2.3 Mixed inputs action recognition

Depth cameras provide different streams, or in other words, different representations of the same action. Some works have attempted to improve classification by combining streams. It can be argued that skeleton-based approaches prove most effective at discriminating actions with broad movements. However, for actions involving similar joint positions and trajectories, such as reading vs. playing on a phone, skeleton-based models do not perform as well. Visual streams can provide important cues such as the type of object held. RGB and depth streams have

been studied extensively. However, to our knowledge, we are the first to use IR data from depth cameras for action recognition.

In (Hu, Zheng, Pan, Lai & Zhang, 2018b; Shahroudy *et al.*, 2017; Wang, Li, Wan, Ogunbona & Liu, 2018b) the complementary role of RGB and depth is demonstrated. In (Zolfaghari, Oliveira, Sedaghat & Brox, 2017), pose, motion, and raw RGB images are inputted in 3 parallel 3D CNNs. Although visual information greatly improves upon the pose baseline, results are comparable with the then state-of-the-art methods using only skeleton data. Pose data can be utilized to extract regions of interest around joints or body parts (Baradel, Wolf & Mille, 2017; Hou, Wang, Wang, Gao & Li, 2017; Rahmani & Bennamoun, 2017). In (Rahmani & Bennamoun, 2017), human-object interactions are modeled using both skeleton and depth data. An end-to-end network is proposed to learn view-invariant representations of skeleton data and held objects. Once again, visual information increases the accuracy but the results do not justify the complexity of a fusion approach compared to the other skeleton-only approaches of the time. The same year, Baradel *et al.* use RGB and skeleton data conjointly in a pertinent way (Baradel *et al.*, 2017). Pose information is used as an input but also conditions the RGB stream. The 3D skeleton data are projected onto the RGB sequences to effectively extract crops around the hands of the subject, serving as another input. The RGB stream thus provides important clues about an object held and inter-subject interactions, significantly improving the results. This work shows that not all body parts need to be focused on, unlike the approach in (Rahmani & Bennamoun, 2017). But this requires as many streams as there are hands, which is memory inefficient. Furthermore, when the hands are close together, the information provided may be redundant. Alternatively in (Tu *et al.*, 2018), a region of interest is created using motion fields from RGB videos. An additional region from the body is extracted using motion saliency. The advantages of this method are that depth data are not required and the attention mechanism role of the saliency map. But for almost motionless actions, such as dropping an object or playing on a phone, the region extraction should not perform as well.

We propose a similar approach to (Baradel *et al.*, 2017; Rahmani & Bennamoun, 2017; Tu *et al.*, 2018) in which the 3D skeleton data provide a crop around the subjects, alleviating the need

for a spatial attention mechanism. A single crop is necessary, even when multiple subjects are interacting, which relaxes the memory needs.

3.3 Proposed Model

We design a deep neural network using skeleton and IR data, called "Full Use of Infrared and Skeleton in Optimized Network" (FUSION). The network consists of two parallel modules and an MLP. One module interprets skeleton data, the other IR videos. The features extracted from each individual stream are then fused using different strategies (average, sum, multiplication, max, convolution, concatenation). The MLP is used as the final module and outputs a probability density. The network is trained in end-to-end fashion by optimizing the classification score.

We note a skeleton sequence $S = \{S_{j,t,k}\}$ where j denotes a joint index, t a frame index and k a coordinate axis (X, Y and Z). We note $I = \{I_t\}$ a sampled IR sequence, as detailed section 3.3.2.3, where t is taken between $\{1, .., T\}$, with T the number of sampled frames.

In the following sections, we present the individual modules of our FUSION model : a 2D CNN as the pose module, a 3D CNN as the IR module and an MLP as the stream fusion module.

3.3.1 Pose module

A skeleton sequence requires careful treatment for optimal results. First, a skeleton sequence is normalized to be position invariant, meaning the distance between the subject and the camera is accounted for. The sequence is then transcribed to an RGB image, with multiple subjects interactions in mind. The handcrafted RGB image is then fed to a 2D CNN.

3.3.1.1 Prior normalization step

Each skeleton sequence is normalized by translating the global coordinate system of the camera to a local coordinate system corresponding to a key joint of the main subject. We choose the middle of the spine as the new origin. This is illustrated Fig. 3.3.

We adopt a sequence-wise normalization. In other words, the translation vector is computed for the first frame and applied to each subsequent frame, meaning the subject may move away from the new local coordinate system, as follows :

$$S' = S_{:,:,:} - S_{1,0,:}. \tag{3.1}$$

Where S' is the normalized skeleton sequence, $j = 1$ corresponds to the middle of the spine for the Kinect 2 skeleton (Zhang, 2012). The ":" notation signifies that all values are considered across this dimension.

3.3.1.2 Skeleton data to skeleton 2D maps

A skeleton sequence is mapped to an image similar to (Du *et al.*, 2015a), a skeleton map. Each coordinate axis, X, Y and Z, is attributed to each channel of an RGB image. Each key joint corresponds to a row while the columns represent the different frames.

We apply a dataset-wise normalization (Du *et al.*, 2015a). We note c_{min} and c_{max} the minimal and maximal values of the coordinates after the normalization step for the entire dataset. The pixels of the skeleton map are recalculated using a min-max strategy in the $[0, 1]$ range, as follows :

$$M = \frac{S' - c_{min}}{c_{max} - c_{min}}. \tag{3.2}$$

Where $M = \{M_{j,t,k}\}$ is the normalized skeleton map with k both the coordinate axis and the image channel.

To accommodate for the fixed input size of the 2D CNN, the skeleton map is resized to a standard size.

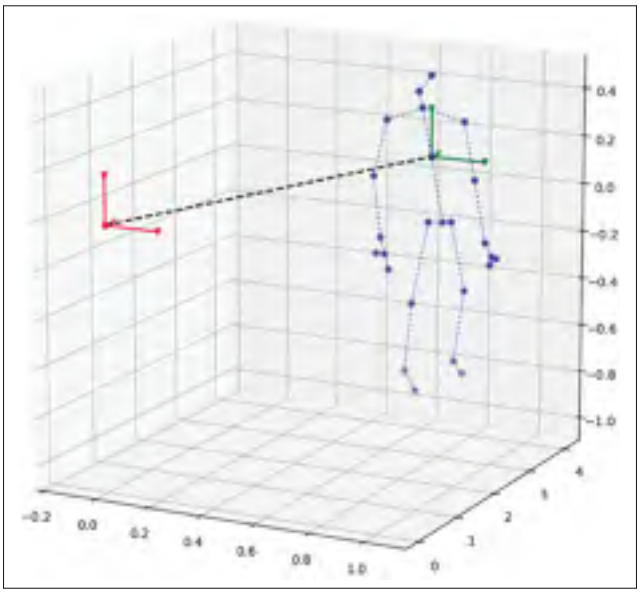


Figure 3.3 In red the coordinate system of the camera, in green the new coordinate system corresponding to the middle of the spine of the main subject for the first frame of the sequence, in blue the skeleton of the main subject, in black the translation vector

3.3.1.3 Multi-subject strategy

Our network is scalable to multiple subjects. We concatenate the different skeleton maps across the joint dimension. With J being the total number of joints, the first J rows correspond to the first subject, the subsequent J rows to subject 2, etc. We limit the number of subjects to two, corresponding to the maximum of the NTU RGB+D dataset (Shahroudy *et al.*, 2016). Nonetheless, this method may be generalized to a greater number of subjects. Should the skeleton sequence comprise only one subject, the J rows of the second subject are set to zero.

In case of multiple subjects, the coordinates of the latter are translated to the local coordinate system of the main subject (Fig. 3.4).

The advantages of our method are manifold. Firstly, this alleviates the need for individual networks for different subjects. Secondly, this representation allows for a second subject to still intervene if its skeleton is detected after the first frame. Thirdly, the distance information is kept as each subject coordinates are translated to the local coordinate system of the first subject.

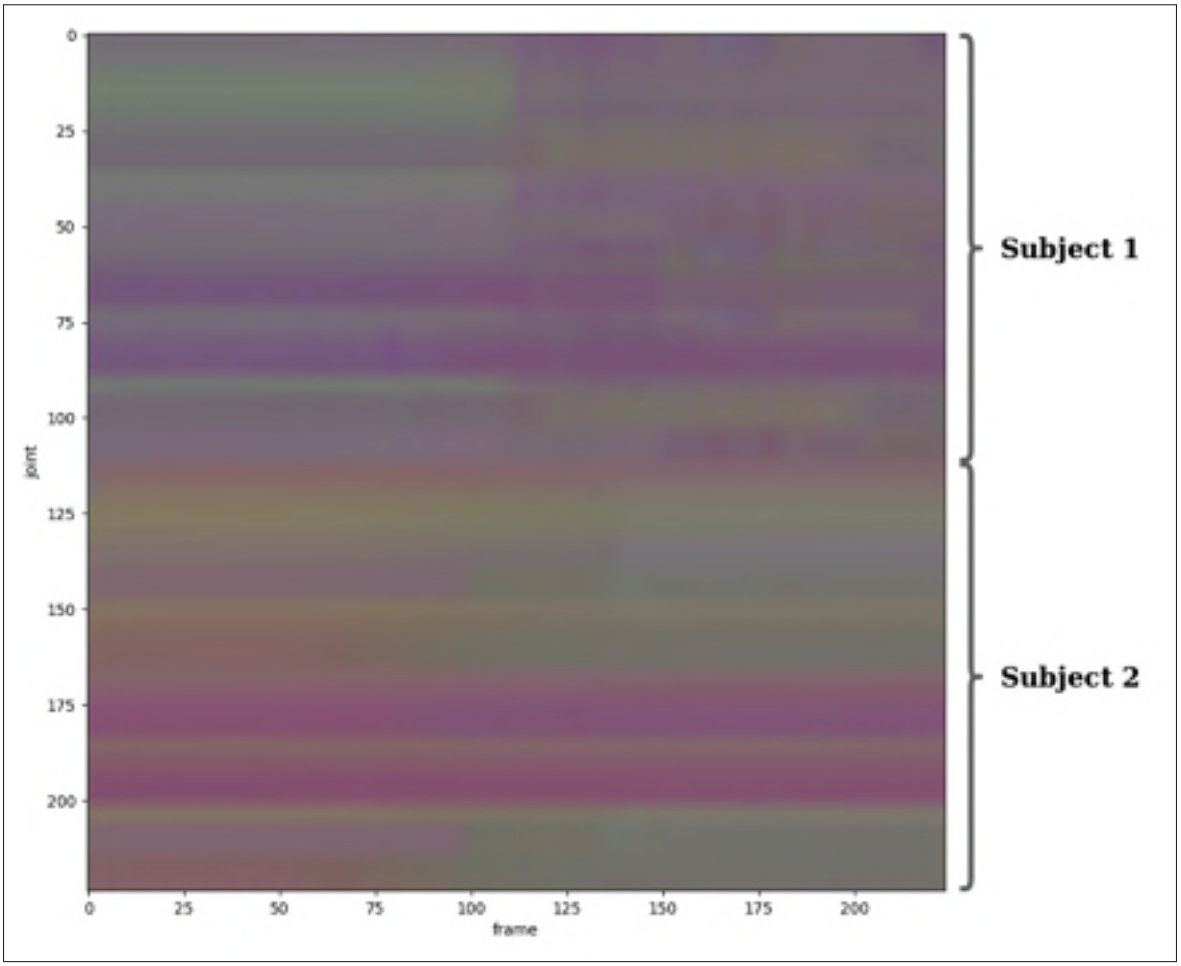


Figure 3.4 Skeleton map of two subjects. The joints of the two subjects are concatenated across a dimension, then stacked over time. The created image is reshaped to the fixed CNN input size

Lastly, the skeleton map is resized to a standard size to accommodate for the fixed input size of the pose module. This implies that the network is able to learn from raw sequences of different sizes.

3.3.1.4 CNN used

The transformed skeleton map is used as input. We use an existing CNN with pre-trained weights on ImageNet as we find this ameliorates the classification score even when the images

are handcrafted. We choose an 18-layer ResNet (He *et al.*, 2016) for its compromise between accuracy and speed.

We extract a pose feature vector s from the skeleton map M with the pose module f_S with parameters θ_S (3.3). Here, and for the rest of the paper, subscripts of modules and parameters refer to a module, not an index.

$$s = f_S(M|\theta_S) \tag{3.3}$$

3.3.2 IR module

The action performed by a subject is only a small region inside the frames of an IR sequence. The 2D skeleton data are used to capture the region of interest and virtually focus the attention of the network, with multiple potential subjects in mind. Because the IR module requires a video input with a fixed number of frames, a subsampling strategy is deployed. A 3D CNN is used to exploit the IR data.

3.3.2.1 Cropping strategy

Traditionally, 3D CNNs require a lot of parameters to account for the complex task of video understanding. Thus, the frames are heavily downscaled to reduce memory needs. In the process, discriminating information may be lost. In an action video of daily activities, the background provides little to no context. We would like our model to only focus on the subject as this is where the action happens. We argue that a crop around the subject provides ample cues about the action performed. Depth information, coupled with pose estimation algorithms, provides a turnkey solution for human detection. We propose a cropping strategy, shown Fig. 3.5 by a green parallelepiped, to virtually force the model to focus on the subject.

Given a 3D skeleton sequence projected on the 2D frames of the IR stream, we extract the maximal and minimal pixel positions across all joints and frames. This creates a fixed bounding

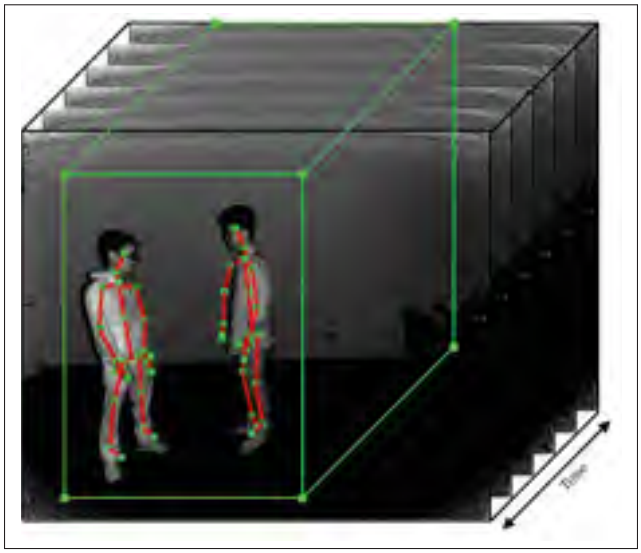


Figure 3.5 A fixed bounding box across the entire sequence is generated using the 2D skeleton information. The new sequence focuses attention on the subject rather than the background which provides little to no context
Adapted from Shahroudy *et al.* (2016)

box capturing the subject on the spatial and temporal domains. We empirically choose a 20 pixels offset to account for potential skeleton inaccuracy. The IR stream is padded with zeros should the box coordinates with the offset exceed the IR frame range.

The advantage of our method is as follows. Providing a crop around the region of interest reduces the size of the frames without decreasing the quality. The downscaling factor is thus less important and preserves a better aspect of the image. Furthermore, it alleviates the need for an attention mechanism as the cropping strategy may be seen as a hard attention scheme in itself. Also, the network does not have to learn information from the background, which is noise in our case, as it is reduced to a minimum.

3.3.2.2 Multi-subject strategy

The cropping strategy can be generalized to multiple subjects. The bounding box is enlarged to account for the other subjects. We take the maximal and minimal values across all joints, frames, and subjects.

For a given sequence, the bounding box is immobile regardless of the number of subjects. This allows keeping camera dynamics. We do not want to add confusion to the sequence by adding a virtual movement of the camera with a mobile bounding box.

3.3.2.3 Sampling strategy

Contrary to the pose network, a given IR sequence is not treated in its entirety. A 3D CNN requires a sequence with a fixed number of frames T. Choices must be made regarding the value of T and the sampling strategy. A potential approach would be to take adjacent frames in a sequence. But the subsequence might not be enough to correctly capture the essence of the action. Instead, we propose a scheme where the raw sequence is divided into T windows of equal duration similar to (Liu *et al.*, 2016), as illustrated Fig. 3.6. A random frame is taken from each window. A new sequence is created of length T. This is a form a data augmentation as a raw sequence may yield different results.

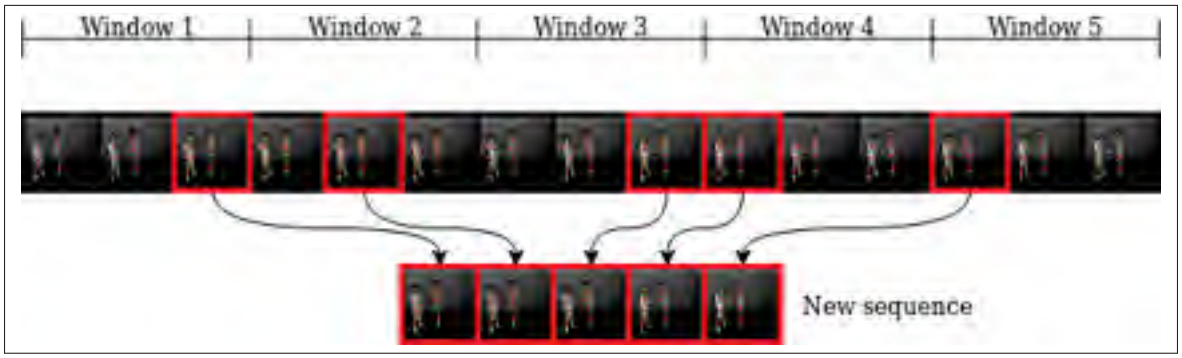


Figure 3.6 Each IR sequence is divided into a fixed number of windows of equal size. For each subdivision, a random frame is sampled. The concatenation of those frames is the input for the IR module

3.3.2.4 3D CNN used

The new sampled sequences are used as inputs for the 3D CNN. We use an 18-layer deep R(2+1)D network (Tran *et al.*, 2018) pre-trained on Kinetics-400 (Carreira & Zisserman, 2017). R(2+1)D is an elegant network which revisits 3D convolutions. Tran *et al.* showed factoring

spatial and temporal convolutions yields state-of-the-art results on benchmark RGB action recognition datasets. Separating spatial and temporal convolutions with a nonlinear activation function in between allows for a more complex function representation with the same number of parameters.

We extract a stream feature vector i from the sampled IR sequence I with the IR module f_{IR} with parameters θ_{IR}, as follows :

$$i = f_{IR}(I|\theta_{IR}). \tag{3.4}$$

3.3.3 Stream fusion

Both pose and IR modules output their own feature vectors. An MLP serves as the final module and returns a probability distribution for each action class in a dataset.

Features of both streams are fused using different schemes (average, sum, multiplication, max, convolution, concatenation). The MLP consists of three layers with batch normalization (Ioffe & Szegedy, 2015) before computation. The *ReLU* activation function is used for all neurons. Lastly, a *softmax* activation function is deployed to normalize the last layer's output into a probability distribution.

The class probability distribution y is outputted by the MLP f_{MLP} with parameters θ_{MLP} (3.5). Inputs i and s correspond to the feature vectors computed by the pose and IR modules.

$$y = f_{MLP}(i, s|\theta_{MLP}) \tag{3.5}$$

We tried a scheme where the pose and IR modules of our network would emit their own prediction. We would then average the predictions on a logits level with learned weights during the backpropagation step. However, this would lead to the network's final classification to be

attributed solely to one module or the other. Instead, we believe that an MLP allows for the features of the different streams to be interpreted conjointly.

3.4 Network Architecture

3.4.1 Architecture

3.4.1.1 Pose module

The pose network is an 18-layer deep ResNet (He *et al.*, 2016). The network takes as input a tensor of dimensions 3x224x224, where 3 corresponds to the RGB channels and 224 to the height and width of the image. The output, s, is a 1D vector of 512 features.

3.4.1.2 IR module

The IR network is an 18-layer deep R(2+1)D (Tran *et al.*, 2018). It takes as input a video of dimensions 3xTx112x112, where 3 corresponds to the RGB channels, T to the length of the sequence and 112 to the height and width of the image. The output, i, is a 1D vector of 512 features.

To be able to leverage the pre-trained R(2+1)D CNN, which is originally trained on RGB images, the IR frames, which are single-channel grayscale images, are duplicated.

3.4.1.3 Classification module

The classification module is an MLP network with three layers. The first layer expects a vector of 512 (average, sum, multiplication, max, convolution) or 1024 (concatenation) features and comprises 256 units. The second layer consists of 128 units. The last layer has as many units as there are different action classes in a dataset. Finally, the *softmax* function is used to normalize the predictions to a probability distribution. Batch normalization is applied before the layers. A

dropout scheme has been tested in place of batch normalization but was not found to be superior. The *ReLU* activation function is used for all layers except the last.

The entire network is detailed Fig. 3.7.

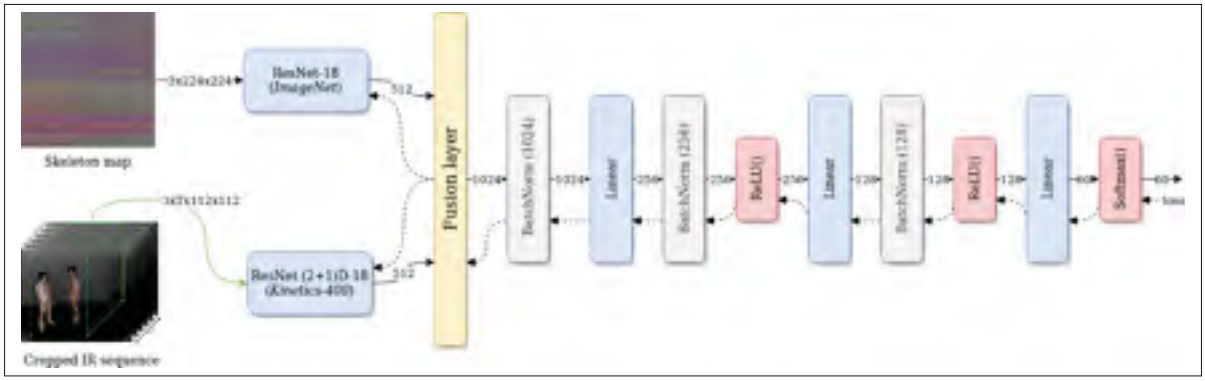


Figure 3.7 The full detailed model. The pose and IR modules output separate feature vectors. The two are fused (average, sum, multiplication, max, convolution or concatenation) and a final MLP outputs a class probability distribution. The pose network is a pre-trained *ResNet-18*. The IR network is a pre-trained network *R(2+1)D-18*

3.4.2 Data augmentation

To prevent overfitting and reinforce the generalization capabilities of our model, we perform data augmentation during training.

The skeleton sequences have limited viewpoints but their representation makes them excellent candidates for augmentation through geometric transformations. The skeleton sequences are enhanced by performing a random rotation around the X, Y and Z axis. For each sequence during training, we apply a random rotation between $-20°$ and $20°$ on each axis.

We approach IR data augmentation with the following scheme. For each sequence during training, we perform a horizontal mirroring transformation on the frames with a 50% chance probability. The two streams are augmented independently.

3.4.3 Training

The network is trained in end-to-end fashion by minimizing cross-entropy loss, meaning all the modules of our network are trained together. The pose network is pre-trained on the ImageNet dataset (Deng *et al.*, 2009). The IR network is pre-trained on the Kinetics-400 dataset (Carreira & Zisserman, 2017).

3.5 Experiments

We evaluate the performances of our proposed model on the NTU RGB+D dataset, the largest benchmark to date (Shahroudy *et al.*, 2016). We also perform extensive ablation studies to understand the individual contributions of our modules.

3.5.1 NTU RGB+D dataset

The NTU RGB+D dataset is the largest human action recognition dataset to date captured with a Microsoft Kinect V2 (Zhang, 2012). To our knowledge, it is also the only one including the IR sequences. It contains 60 different classes ranging from daily to health-related actions spread across 56,880 clips and 40 subjects. It includes 80 different views. An action may require up to two subjects. The various setups, views, orientations, result in a great diversity which makes NTU RGB+D a challenging dataset.

There are two benchmark evaluations for this dataset : Cross-Subject (CS) and Cross-View (CV). The former splits the 40 subjects into training and testing groups. The latter uses the samples acquired from cameras 2 and 3 for training while the samples from camera 1 are used for testing.

3.5.2 Experimental settings

For consistency, we do not modify the following hyperparameters across all experiments. We set the batch size to 16 which allows the model and a batch to fit on most high-end GPUs. Gradient clipping is used to avoid an exploding gradient issue. We set it to 10. Adam optimizer

(Kingma & Ba, 2014) is used to train the networks. A learning rate of 0.0001 is set and kept consistent during training.

The pose and IR modules each require a fixed input size. Skeleton maps are resized to 224x224 images. IR frames are resized to 112x112.

To assure consistency and reproducibility, we use a pseudorandom number generator fed with a fixed seed. Following (Shahroudy *et al.*, 2016), we sample 5% of the training set as our validation set.

3.5.3 Ablation studies

In this section, we isolate the pose and IR modules and study their individual contribution with regard to different parameters. Action classification accuracy on the NTU RGB+D dataset is used as the comparison metric. We evaluate the impact of transfer learning, data augmentation, pose conditioning of IR sequences and the number of frames T. Finally, we compare our results with the current state of the art.

3.5.3.1 Pose module

We evaluate the performances of our pose module as a standalone. The IR module does not intervene. We also adjust the input size of the classification MLP. Optimal results are achieved by combining pre-training with data augmentation. Table 3.1 shows the best results of the pose module on NTU RGB+D : 82.3% on CS and 89.5% on CV.

Tableau 3.1 Results of the pose module on NTU RGB+D dataset (accuracy in %)

Method	Pose	IR	CS	CV
Pose network	X	-	82.3	89.5

The CV benchmark is a much easier task, hence the better results compared to CS. The test actions are already seen during training but from a different point of view with a different camera. Although the different setups yield different joint position estimations for a given sequence (Zhang *et al.*, 2019), the geometric nature of skeleton data allows for a better generalization. This is not the case for the CS task as the test sequences are completely novel. Consequently, the following discussions will only address the CS benchmark.

The confusion matrix reveals the pose module's strong ability to correctly classify actions with intense kinetic movements. Actions such as sitting down, standing up, falling, jumping, staggering, walking toward or away from another subject are classified with over 95% accuracy. Unsurprisingly, actions with similar skeleton motions prove the most challenging. Writing is the trickiest, with 40% accuracy only and often mislabeled as writing or typing on a keyboard. The incorrectly classified actions fall under two categories : similar motion actions and object-related actions. We believe this will always be a limitation of pose-only networks.

3.5.3.2 Infrared module

The other part of the FUSION network, and arguably the most important contributor, is the infrared module. In a similar fashion as above, the input size of the MLP is adjusted while keeping the number of neurons equal. Optimal results are achieved with a pre-trained network, with data augmentation, on pose-conditioned inputs for a sequence length of $T = 20$. Table 3.2 shows the performance of the IR module as a standalone : 89.8% on CS and 94.1% on CV.

Tableau 3.2 Results of the IR module on NTU RGB+D dataset (accuracy in %)

Method	**Pose**	**IR**	**CS**	**CV**
IR network	-	X	89.8	94.1

The confusion matrix reveals a more balanced accuracy score over the different actions of the NTU RGB+D dataset. Some actions, such as touching another person's pocket or staggering,

prove more difficult to recognize for the IR module compared to the pose module. This reinforces our intuition that pose and visual streams are complementary. However, some object-oriented actions are still difficult to correctly discern. For instance, writing is more often than not mislabeled as playing with a phone. We propose two possible explanations. Firstly, the object information might be lost during the rescaling process, even with our cropping strategy in place. Secondly, the IR nature, grayscale and noisy, might not be clear enough to discern the object correctly. But other object-related actions such as dropping an object or brushing hair see an impressive improvement of over 10%.

3.5.3.3 Influence of feature fusion scheme

We test various deep feature fusion schemes : average (avg), sum, multiplication (mult), max, convolution (conv) and concatenation (conc). The average, sum, multiplication and max fusion schemes are done in element-wise fashion. The convolution scheme considers the i and s feature vectors as a 512x2 image. The features are convoluted by a 2D kernel of size $(1, 2)$. A new 1D feature vector with 512 new computed features is thus outputted. Table 3.3 shows the impact of the different fusion schemes on classification accuracy for the CS and CV benchmarks.

Tableau 3.3 Impact of fusion scheme on classification performances
(A : Augmented | P : Pre-trained | C : cropped inputs)
(accuracy in %)

Method	Pose	IR	CS	CV	Avg.
FUSION - CPA (conv)	X	X	91.4	94.6	93.00
FUSION - CPA (mult)	X	X	92.0	94.1	93.05
FUSION - CPA (conc)	X	X	91.6	94.5	93.05
FUSION - CPA (max)	X	X	91.9	94.3	93.10
FUSION - CPA (sum)	X	X	**92.3**	94.7	93.50
FUSION - CPA (avg)	X	X	92.0	**95.1**	**93.55**

The different schemes perform similarly, especially the convolution (93.00% average on CS and CV), multiplication (93.05%), concatenation (93.05%) and max (93.10%) strategies. More

convincingly, the sum (93.50%) and average (93.55%) schemes improve average accuracy by over 0.3% compared to the other schemes. It is reassuring to note that the sum and average scheme, which are alike, perform similarly. Nonetheless, regardless of the chosen scheme, results are systematically improved compared to the pose module (Table 3.1) and the IR module (Table 3.2).

3.5.3.4 Influence of pre-training

Pre-training a network is an elegant way to transfer a learned task to a new one. It has been shown to provide impressive results even on handcrafted images (Zhang *et al.*, 2019). Furthermore, it helps with the overfitting issue smaller datasets may demonstrate.

We evaluate the impact of this strategy on our network. Table 3.4 shows the effect of pre-training on the different modules.

Tableau 3.4 Impact of pre-training on classification performances
(A : Augmented | P : Pre-trained | C : cropped inputs)
(accuracy in %)

Method	Pose	IR	CS	CV
Pose module	X	-	78.7	85.1
Pose module - P	X	-	**80.7**	**87.0**
IR module	-	X	76.8	76.3
IR module - P	-	X	**84.0**	**84.6**
IR module - C	-	X	84.6	88.6
IR module - CP	-	X	**90.1**	**91.2**

The pose network enjoys a noticeable increase in accuracy of about 2% for both benchmarks (78.7% to 80.7% on CS). It is pre-trained on ImageNet, which consists of real-life images. The skeleton maps used as inputs are handcrafted. Even then, a pre-training scheme shows encouraging results.

The impact of pre-training on the IR module's accuracy is significant. For uncropped sequences, the accuracy increases by about 7% for both benchmarks (76.8% to 84.0% on CS). For cropped sequences, the gain is over 5% for the cross-subject benchmark (84.6% to 90.1%) and almost 3% for cross-view (88.6% to 91.2%).

The greater contribution of transfer learning for the IR module compared to the pose module might be explained by the greater resemblance of IR vs. RGB videos compared to handcrafted vs. real-life images. Nonetheless, such findings further emphasize the power of transfer learning.

3.5.3.5 Influence of data augmentation

Data augmentation consists of virtually enlarging the dataset, thus hopefully preventing overfitting and reducing variance between training and test sets. We perform augmentation for the different data streams. Table 3.5 shows the performances of data augmentation on the different modules with pre-trained networks. Overall, data augmentation yields favorable results.

Tableau 3.5 Impact of data augmentation on classification performances
(A : Augmented | P : Pre-trained | C : cropped inputs)
(accuracy in %)

Method	Pose	IR	CS	CV
Pose module - P	X	-	80.7	87.0
Pose module - PA	X	-	**82.3**	**89.5**
IR module - P	-	X	84.0	84.6
IR module - PA	-	X	**84.9**	**87.5**
IR module CP	-	X	**90.1**	91.2
IR module CPA	-	X	89.8	**94.1**
FUSION - CP (avg)	X	X	91.5	94.0
FUSION - CPA (avg)	X	X	**92.0**	**95.1**

The pose module alone enjoys an increase of about 2% accuracy for both benchmarks (80.7% to 82.3% on CS). The IR module alone seems to benefit more from data augmentation on the CV benchmark compared to the CS. For the CV benchmark, the increase is about 3% whether the

input sequence is cropped (91.2% to 94.1%) or not (84.6% to 87.5%). For the CS benchmark, the improvements are not significant. When the modules are fused using an element-wise average scheme, our FUSION network, data augmentation is favorable with an increase of 0.5% for the CS benchmark (91.5% to 92.0%) and 1.1% for the CV benchmark (94.0% to 95.1%). As the baseline results increase, the gains are expected to diminish.

3.5.3.6 Transfer learning vs. data augmentation

Transfer learning and data augmentation are two strategies to better generalize the performances of a network. Transfer learning leverages the learned parameters from another dataset while data augmentation virtually enlarges the current dataset. A small dataset might lead to overfitting which results in an increase in variance between the training and validation sets as the training error continues to lower.

Our model is able to reach a negligible training error, even with individual modules, showcasing an overfitting issue. Having studied the impacts on performances of both methods, transfer learning shows much better results. This might be explained by the already large size of the NTU RGB+D dataset mitigating the potential of data augmentation. Nonetheless, it is formidable how a model can yield vastly different performances based on the initialization of its parameters. The black-box nature of deep learning makes the interpretation of how a model learns difficult. Perhaps future works will focus on understanding the internal representation of a network to guide its learning rather than implementing evermore complex models.

3.5.3.7 Influence of pose-conditioned cropped IR sequences

In this section, we evaluate the impact of our cropping strategy, detailed section 3.3.2.1, on the performances of the IR module as a standalone. Table 3.6 shows a significant increase in performances.

Our baseline for this comparison, the IR module without transfer learning and data augmentation on uncropped sequences, reports unsatisfactory results (76.8% on CS). With transfer learning

Tableau 3.6 Impact of our cropping strategy on classification performances
(A : Augmented | P : Pre-trained | C : cropped inputs)
(accuracy in %)

Method	Pose	IR	CS	CV
IR module	-	X	76.8	76.3
IR module - C	-	X	**84.6**	**88.6**
IR module- PA	-	X	85.0	87.5
IR module - CPA	-	X	**89.8**	**94.1**

and data augmentation, we are able to increase the accuracy by almost 10% average for both benchmarks (76.8% to 84.9% on CS). However, we find that our cropping strategy alone reaps similar benefits (76.8% to 84.6% on CS). When combining all three strategies, we further ameliorate the classification score by about 5% (89.8% on CS). The average gain for both benchmarks is thus above 15%, which is considerable.

We demonstrate the power of a pragmatic approach. An identical model is able to perform significantly better thanks to careful design choices.

3.5.3.8 Influence of sequence length

Sequences of the NTU RGB+D dataset are at most a couple of seconds long. We study the impact of the length T of the new sampled IR sequence on classification performances of two networks : the IR module only and on the complete FUSION model. Both models are pre-trained and fed with augmented data. The IR sequences are pose-conditioned. Table 3.7 reports the impact of different values of T on the accuracy score.

As a general tendency, the greater the value of T, the better the results. Best results are achieved for $T = 20$, for three out of four scenarios (on CS : 89.8% for IR module only and 92.0% for FUSION). The exception happens for the IR module as a standalone on the CS benchmark where the optimal value is $T = 16$ (90.0%). However, the difference in accuracy is negligible. For the FUSION network, excellent results are achieved for a number of frames as little as $T = 8$ (90.5%

Tableau 3.7 Impact of IR sequence length on classification performances
(A : Augmented | P : Pre-trained | C : cropped inputs) (accuracy in %)

Method	Pose	IR	CS				CV			
			T=8	T=12	T=16	T=20	T=8	T=12	T=16	T=20
IR module - CPA	-	X	86.8	89.5	90.0	89.8	88.8	91.3	93.0	94.1
FUSION - CPA (avg)	X	X	**90.5**	**90.6**	**90.6**	**92.0**	**93.0**	**92.7**	**94.0**	**95.1**

on CS and 93.0% on CV). However, the results really shine with $T = 20$. But FUSION networks with a smaller value of T are much faster, showcasing a trade-off between speed and accuracy.

3.5.3.9 Comparison with the state of the art

We compare our FUSION model, using an average fusion scheme, with the state of the art (Table 3.8). We divide current methods into 5 different frameworks including handcrafted features, RNN-based methods, CNN-based methods, fusion methods, and GCN-based methods. Current best results are obtained using skeleton data only with GCNs. We achieve better results than the current state of the art on the CS benchmark (92.0%) with 2.1% accuracy increase. On the CV benchmark, results are comparable (95.1% for FUSION against 96.1% for DGNN (Shi *et al.*, 2019a)). We conclude to the efficacy of IR data to correctly interpret human actions.

We significantly improve upon current fusion methods, once again validating the complementary role of pose and visual data.

3.6 Conclusion

We propose an end-to-end trainable network using skeleton and infrared data for human action recognition. A pose module extracts features from skeleton data and an infrared module learns from videos. The 3D skeleton is used as an input source and also conditions the infrared stream, providing a crop around the subjects. The two stream features are then concatenated, and a final prediction is outputted. The pose and infrared modules report strong individual performances, which is greatly due to the power of transfer learning as they are both pre-trained on other large

Tableau 3.8 Comparison of our model to the state of the art
(A : Augmented | P : Pre-trained | C : cropped inputs) (accuracy in %)

Method	Pose	RGB	Depth	IR	CS	CV
Lie Group (Vemulapalli *et al.*, 2014)	X	-	-	-	50.1	82.8
HBRNN (Du *et al.*, 2015b)	X	-	-	-	59.1	64.0
Deep LSTM (Shahroudy *et al.*, 2016)	X	-	-	-	60.7	67.3
PA-LSTM (Shahroudy *et al.*, 2016)	X	-	-	-	62.9	70.3
ST-LSTM (Liu *et al.*, 2016)	X	-	-	-	69.2	77.7
STA-LSTM (Song *et al.*, 2017)	X	-	-	-	73.4	81.2
VA-LSTM (Zhang *et al.*, 2017)	X	-	-	-	79.2	87.7
TCN (Kim & Reiter, 2017)	X	-	-	-	74.3	83.1
C+CNN+MTLN (Ke *et al.*, 2017b)	X	-	-	-	79.6	84.8
Synthesized CNN (Liu *et al.*, 2017b)	X	-	-	-	80.0	87.2
3scale ResNet (Li *et al.*, 2017a)	X	-	-	-	85.0	92.3
DSSCA-SSLM (Shahroudy *et al.*, 2017)	-	X	X	-	74.9	-
(Rahmani & Bennamoun, 2017)	X	-	X	-	75.2	83.1
CMSN (Zolfaghari *et al.*, 2017)	X	X	-	-	80.8	-
STA-HANDS (Baradel *et al.*, 2017)	X	X	-	-	84.8	90.6
Coop CNN (Wang *et al.*, 2018b)	-	X	X	-	86.4	89
ST-GCN (Yan *et al.*, 2018)	X	-	-	-	81.5	88.3
DGNN (Shi *et al.*, 2019a)	X	-	-	-	89.9	**96.1**
Pose module - PA	X	-	-	-	82.3	89.5
IR module - CPA	-	-	-	X	89.8	94.1
FUSION - CPA (avg)	X	-	-	X	**92.0**	95.1

scale datasets. When working in symbiosis, the results are further ameliorated. We are the first to conjointly use pose and infrared streams. Our method improves the state of the art on the largest RGB-D action recognition dataset to date. Compared to other stream fusion approaches, our method requires less prepossessing and is more memory efficient.

Our work demonstrates the strong representational power of infrared data, which opens the door for applications where illumination conditions render RGB videos unusable. The complementary role of pose and visual streams is further illustrated, which is in line with previous work. Given the modular nature of our proposed network, future works could focus on more modern pose modules such as graph neural networks.

3.7 Acknowledgment

This work was supported by research funding from the Natural Sciences and Engineering Research Council of Canada, Prompt Québec, and an industrial funding from Aerosystems International Inc. The authors would also like to thank their collaborators from Aerosystems International Inc.

CHAPITRE 4

DISCUSSION DES RÉSULTATS

Dans ce chapitre, les résultats du chapitre 3 sont interprétés, mais nous proposons des axes de discussion différents à ceux développés dans l'article afin d'éviter des redondances. Les résultats sont confrontés aux objectifs initiaux du mémoire. L'infrarouge démontre réellement son pouvoir de différenciation, à tel point qu'il peut être envisagé comme flux d'information unique. Parallèlement, l'importance si prononcée d'un préentraînement motive une compréhension et une représentation plus poussée des caractéristiques extraites par le réseau.

4.1 Infrarouge comme flux unique ?

Comme développé section 2.4, la démonstration de la puissance de l'infrarouge seul pourrait justifier l'utilisation d'une caméra RGB+D de la famille *RealSense* qui souffre d'un kit de développement limité. Ces caméras sont moins chères et moins gourmandes en ressources que la dernière *Kinect Azure*, mais ne peuvent proposer flux infrarouge et squelette 3D conjointement. Or, c'est précisément la stratégie employée dans l'article.

4.1.1 Infrarouge seul

L'infrarouge est évalué comme flux unique en utilisant deux stratégies de prétraitement. La première consiste à utiliser les données brutes. La deuxième consiste à recadrer la vidéo autour des sujets afin de forcer virtuellement le réseau à se focaliser sur la zone d'intérêt. Effectivement, l'arrière-plan d'une séquence ne donne pas ou peu d'information sur l'action réalisée. La vidéo est recadrée à l'aide des coordonnées 2D du squelette humain.

Ici, et pour tout le chapitre, les résultats discutés seront ceux du protocole CS de la base de données NTU RGB+D, plus difficile que le protocole CV.

En utilisant l'infrarouge seul, les meilleurs résultats (en termes d'exactitude) sont de 85% sans prétraitement, et de 89.8% avec. La différence entre les deux stratégies est considérable,

d'autant plus qu'au fur et à mesure que le réseau se rapproche d'un sans faute, les derniers points deviennent plus difficiles à obtenir.

Pour des contraintes de mémoire matérielle, le module infrarouge demande de compresser les dimensions spatiales des vidéos à 112x112 pixels. En recadrant avant de redimensionner, l'image conserve plus d'informations, car moins compressée. Cela peut expliquer pourquoi les actions impliquant l'utilisation d'objets et des mouvements avec les mains sont mieux reconnues avec un prétraitement.

Néanmoins, le prétraitement requiert les coordonnées du squelette 2D, qui proviennent directement du squelette 3D. Mais, on pourrait imaginer un réseau de détection d'objets effectuant la même tâche. Ici, l'objet à détecter serait le ou les sujets performant l'action. Compte tenu de la vitesse et des performances des architectures de la famille YOLO (Redmon *et al.*, 2016), un tel dispositif devrait fonctionner correctement en temps réel.

En poussant l'idée encore plus loin, ce réseau de détection pourrait donner des informations contextuelles grâce aux autres objets présents (arme, corde, etc.). Une telle architecture dépasse le cadre de ce mémoire, mais est une piste de recherche intéressante.

4.1.2 Infrarouge et squelette 3D

En plus du module infrarouge avec prétraitement, l'introduction du module squelette améliore les résultats. Le modèle passe de 89.8% à 91.6% avec les deux modules. En effet, le squelette 3D est particulièrement puissant pour reconnaître des actions dynamiques telles que sauter, donner un coup de pied ou un coup de poing. Le module infrarouge seul est plus homogène en termes de performance sur les différentes classes.

4.1.3 Un gain de performance nécessaire ?

Le gain de performance, bien que notable, est relativement faible. Avec moins de 2% d'amélioration, il n'est pas exagéré de considérer que la performance ne justifie pas d'utiliser la dernière

Kinect Azure, qui, comparée à la famille RealSense, propose la détection du squelette 3D dans son kit de développement.

4.2 Compréhension de l'apprentissage et apprentissage compréhensif

4.2.1 Préentraînement et limites de l'apprentissage profond

Les gains de performances dus au préentraînement sont considérables, notamment lorsqu'on l'applique au module infrarouge. Avec prétraitement, l'exactitude sur l'ensemble de test passe de 84,6% à 90,1%, soit un gain de plus de 5%. Sans prétraitement, la précision passe de 76,8% à 84,0%, soit plus de 7% ! Un simple préentraînement suffit à dépasser l'état de l'art développé sur plusieurs années. Cela amène naturellement à se questionner sur de futures dynamiques de recherche en apprentissage profond.

Actuellement, la recherche en reconnaissance d'activités à l'aide de caméras RGB+D s'essouffle. Depuis l'introduction des réseaux convolutifs sur des graphes, les publications proposent des améliorations minimes. Par exemple, un nouveau module qui améliorerait la classification de moins d'un pour cent. De plus, compte tenu de la forte nature non déterministe de l'entraînement en apprentissage machine, de l'absence d'études statistiques sur les performances, il est légitime de questionner le véritable impact de ces contributions. Par exemple, comment conclure qu'un module est réellement efficace s'il n'améliore que très peu les performances ? En se basant sur une unique expérience, cela pourrait simplement être dû au hasard. La recherche appliquée en apprentissage profond manque de formalisme statistique pour réellement valider les diverses contributions.

L'apprentissage profond est une branche de recherche très empirique qui tient parfois plus de l'art que de la science. Cela est dû au phénomène de boite noire. En effet, entre la sortie et l'entrée d'un réseau, on ne comprend pas bien la représentation interne du réseau. Et pourtant, avec des gains aussi conséquents apportés par un préentraînement, cela semble prometteur. Un même réseau entraîné sur les mêmes données est capable de performances sensiblement

différentes uniquement en changeant l'initialisation de ses poids internes. Pour une application sensible, comme la sécurité, nous trouvons optimiste de laisser une trop grande autonomie à ce genre d'architecture.

4.2.2 Comprendre la représentation d'un réseau

Comprendre pourquoi un réseau émet un résultat est une tâche difficile. Elle l'est d'autant plus que les données en entrée sont abstraites pour l'humain. En effet, il est plus parlant de donner une représentation d'un réseau qui apprend sur des images plutôt que sur des données purement numériques, par exemple.

Actuellement, des techniques existent permettant de mieux comprendre l'apprentissage d'un CNN 2D. Selvaraju, Cogswell, Das, Vedantam, Parikh & Batra (2017) présentent une méthode appelée Grad-CAM. Cette méthode utilise les gradients issus de la dernière couche d'un réseau convolutif, créant ainsi une cartographie des caractéristiques saillantes de l'image en entrée comme le montre la Figure 4.1.

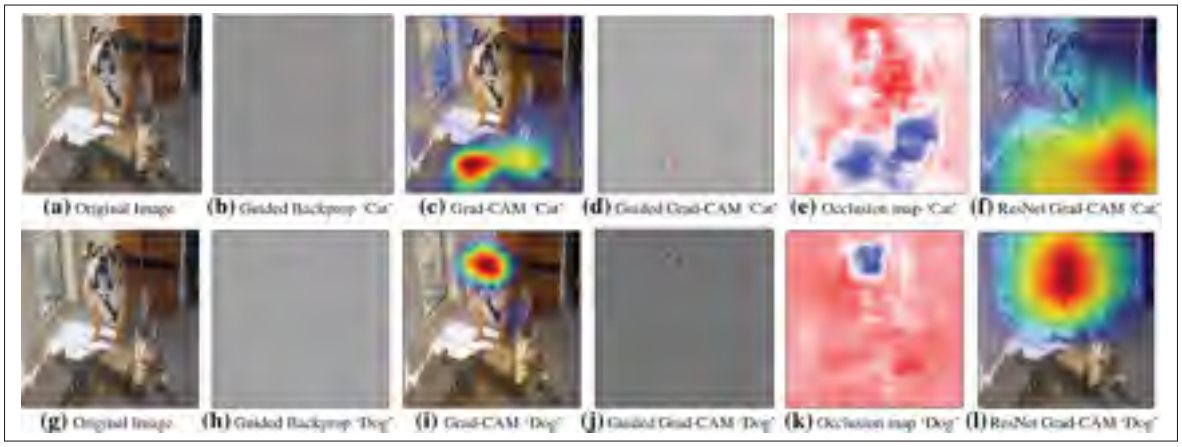


Figure 4.1 Grad-CAM
Tirée de Selvaraju *et al.* (2017)

Une autre méthode consiste à visualiser les poids des différents noyaux de convolutions et la représentation intermédiaire ainsi créée par chaque filtre. La compréhension est aisée pour les premières couches, mais devient bien plus abstraite pour les suivantes.

En reconnaissance d'activités à l'aide de vidéos RGB, Baradel, Wolf, Mille & Taylor (2018) proposent une modélisation de la rétine humaine qui s'intéresse à des zones d'intérêts de l'image. Cela permet de donner une compréhensibilité à ce que visualise le réseau.

Mais force est de constater que ces interprétations se font a posteriori de l'entraînement. L'autonomie, qui fait la force de l'apprentissage profond, est aussi un reproche qui peut lui être adressé. Sans comprendre pourquoi un réseau se comporte d'une certaine façon, il est assez optimiste de l'utiliser sereinement pour des applications importantes.

Autre domaine de recherche, l'apprentissage profond bayésien est saisissant de par ses forts fondements mathématiques et par l'introduction d'un degré d'incertitude. Actuellement, si le réseau présenté chapitre 3 émet la bonne prédiction ou se trompe, il est impossible de savoir pourquoi.

4.3 Vers la détection d'activités humaines *online*

Fort des résultats encourageants développés au chapitre 3, la transition naturelle est de progresser vers une prédiction anticipée, puis vers une détection *online* d'activités humaines. Dans cette section, nous discutons de la faisabilité de cette transition.

4.3.1 De reconnaissance à prédiction anticipée à détection *online*

Tel que présenté plus en détail dans un article de conférence au chapitre 5, il est possible de relier reconnaissance d'activités, à reconnaissance anticipée, puis à détection *online*.

Actuellement, la plupart des articles s'intéressent à la classification d'une séquence dans son entièreté. Sans modifier l'architecture proposée dans l'article du chapitre 3, il est possible d'utiliser en entrée des sous-portions de séquence. Néanmoins, malgré une classification sur une séquence partielle, la prédiction n'est toujours pas *online*. En effet, la fin de la séquence reste connue et est utilisée pour le prétraitement et l'échantillonnage, ce qui n'est pas le cas dans un cadre *online*.

En modifiant cette fois-ci l'architecture du réseau, il est possible de distiller les connaissances apprises par le réseau *offline* vers un réseau *online*, qui cette fois-ci n'a pas connaissance du début et de la fin d'une séquence. Pour ce faire, certains modules du réseau *offline* sont réutilisés, améliorant les performances et le temps nécessaire à l'entraînement.

4.3.2 Limite du prétraitement

Le prétraitement sur les séquences infrarouges détaillé chapitre 3 ne peut pas être utilisé dans un contexte *online*. En effet, la vidéo est recadrée à l'aide des coordonnées 2D du squelette. Les extrémités du cadre de l'action sont extraites à partir des coordonnées de la séquence entière. Dans un contexte *online*, cela correspond à des informations du futur qui ne sont pas disponibles. À la place, un recadrage est effectué à chaque nouvelle image de la vidéo (chapitre 5). Les performances de reconnaissance sont certes moins bonnes qu'avec le prétraitement *offline*, mais meilleures que sans.

CHAPITRE 5

BRIDGING THE GAP BETWEEN HUMAN ACTION RECOGNITION AND ONLINE ACTION DETECTION WITH KNOWLEDGE DISTILLATION ON INFRARED VIDEOS

Alban Main de Boissiere[1], Rita Noumeir[1]

[1] Département de Génie Électrique, École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Article soumis à la conférence « European Conference on Computer Vision », mars 2020.

Abstract : Action recognition, early prediction and online action detection are complementary disciplines that are often studied independently. Most online vision networks use a pre-trained feature extractor without proper fine-tuning. We propose a teacher-student framework between the above-mentioned disciplines and a novel architecture, Online Knowledge Distillation Action Detection network (OKDAD), which embeds online early prediction and online temporal segment proposal modules. Low interclass and high intraclass similarity are encouraged during offline early prediction teacher training, which improves performances. Knowledge distillation to the OKDAD network is assured via layer reuse and cosine similarity between teacher-student feature vectors. This significantly improves our baseline, but we conclude to the greater contribution of layer reuse over similarity learning. We evaluate our framework on infrared videos from two popular datasets, NTU RGB+D (action recognition) and PKU MMD (action detection). We achieve state-of-the-art results on both datasets.

5.1 Introduction

Computer vision branches out to many subfields that are often studied independently. In the video understanding domain, action recognition aims at classifying segmented sequences. On the other end, action detection embodies the ability to detect and classify multiple activities inside a longer, unsegmented sequence.

The emergence of consumer grade depth cameras (RGB+D) (Keselman *et al.*, 2017 ; Zhang, 2012) has sparked a new research dynamic in action understanding. Real-time pose estimation algorithms (Shotton *et al.*, 2011) are employed to extract 3D skeleton data from the depth stream. Additionally, RGB and infrared streams are also available. Except for the infrared, the various streams have been widely studied (Wang *et al.*, 2018a). In essence, the infrared and RGB representations are similar, with an advantage for the former. Infrared videos are represented on a gray scale and are noisy, which logically encourages the use of RGB data. But infrared is less impacted by illumination conditions and is usable in the dark, an important property for security applications.

Action recognition is usually conducted by analyzing a sequence in its entirety before emitting a prediction. Early efforts leveraged recurrent neural networks (RNN) to study skeleton data as temporal series (Liu *et al.*, 2016 ; Shahroudy *et al.*, 2016 ; Zhang *et al.*, 2017). This then shifted toward 2D convolutional neural networks (CNN) (Ke *et al.*, 2017b ; Kim & Reiter, 2017 ; Zhang *et al.*, 2019) and graph convolutional networks (GCN) (Yan *et al.*, 2018). But temporal normalization is always performed through sampling (Liu *et al.*, 2016 ; Shahroudy *et al.*, 2016 ; Zhang *et al.*, 2017), image mapping and rescaling (Ke *et al.*, 2017b ; Kim & Reiter, 2017 ; Zhang *et al.*, 2019) or global pooling (Yan *et al.*, 2018). As such, most action recognition methods are considered offline.

Early action prediction aims at recognizing a human activity before it is fully executed. The objective is therefore similar to the classification task of online action recognition. While some approaches are in line with the online paradigm (Ma *et al.*, 2016 ; Sadegh Aliakbarian, Sadat Saleh, Salzmann, Fernando, Petersson & Andersson, 2017), recent attempts tackle the problem by dividing a sequence into N shorter segments before evaluating a subset of these (Hu, Zheng, Ma, Wang, Lai & Zhang, 2018a ; Pang *et al.*, 2019 ; Wang *et al.*, 2019). In other words, the duration of the sequence is still used, which is considered offline.

Online action detection evaluates raw, unsegmented sequences containing multiple actions (De Geest *et al.*, 2016). The objective is to recognize an occurring activity and classify it frame

by frame, or by chunks of frames, as it happens. It differs from offline action detection where the sequence is studied in its entirety before temporal segments are proposed, then classified.

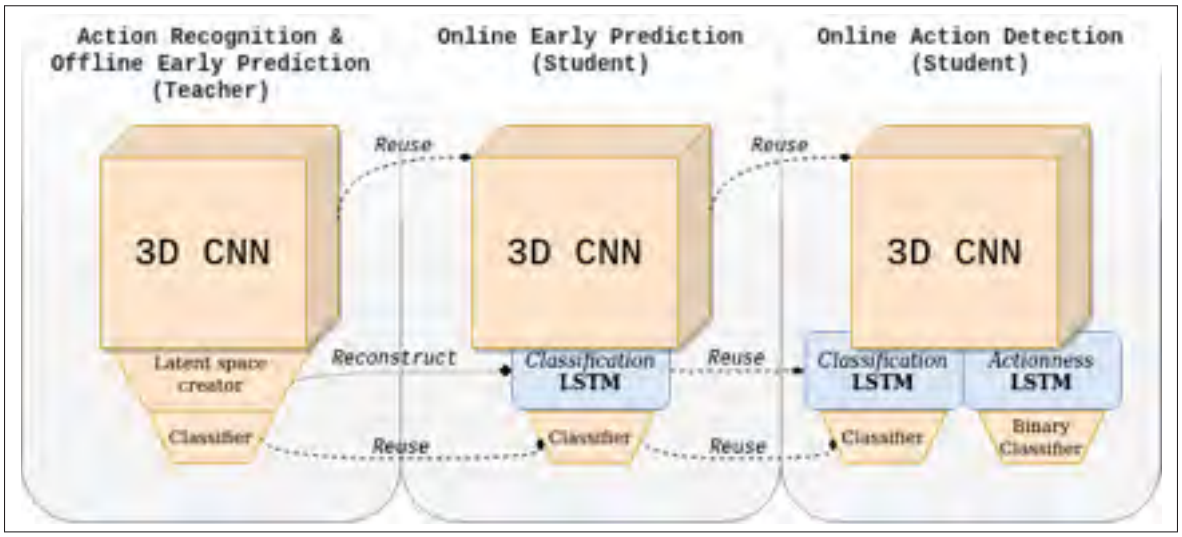


Figure 5.1 Action recognition to online action detection framework. A teacher is first trained. Its 3D CNN and classifier are reused by the students. Teacher feature vector reconstruction is encouraged via knowledge distillation. The online action detection student (OKDAD) embeds an additional temporal proposal module

There appears to be a natural progression from action recognition to online action detection. We propose a framework (Figure 5.1) to link the above-mentioned research fields, using a 3D CNN and long short-term memory (LSTM) RNNs as building blocks. We deploy an offline network to tackle both action recognition and early prediction (teacher). We use it to distill knowledge to an early prediction student network, which is online. Likewise for an online action detection student. The classification LSTM is used to recognize an ongoing action for both students. The "actionness" LSTM, used only for online action detection, considers a block of frames as containing an action or not. From here on out, we refer to actionness as the likelihood of a frame block being an action. We train our network on infrared videos from RGB+D cameras, encouraged by (Main de Boissiere & Noumeir, 2020).

In summary, our main contributions are as follows : 1) A novel framework for both RGB+D human action recognition and offline early prediction with infrared videos ; 2) Cosine similarity loss

terms, which improve teacher accuracy ; 3) A teacher-student knowledge distillation framework for offline to online early prediction based on cosine similarity ; 4) An online action detection architecture which builds upon the online student network ; 5) State-of-the-art results and extensive experiments on infrared videos from two popular benchmark datasets.

Project code will be publicly available upon acceptance of the paper. Video demonstrations and further illustrations are available in the supplementary materials.

5.2 Related work

5.2.1 Human action recognition

Pioneer approaches for video action recognition used handcrafted spatiotemporal features, such as scale-invariant feature transform, histogram of oriented gradients, and improved Dense Trajectories, which are still competitive (Wang & Schmid, 2013).

Recent efforts shifted toward deep learning. In (Karpathy *et al.*, 2014), different temporal fusing schemes are explored, with 2D CNNs as spatial feature extractors. In (Simonyan & Zisserman, 2014), a two-stream network models spatiotemporal features via RGB images and optical flow. Temporal dependencies may also be modeled via recurrent networks (Donahue *et al.*, 2015 ; Yue-Hei Ng, Hausknecht, Vijayanarasimhan, Vinyals, Monga & Toderici, 2015). A 2D CNN outputs a feature vector for each frame, or group of frames, which is then fed to an LSTM network. In (Donahue *et al.*, 2015), the CNN is pre-trained and frozen during training, which might not be ideal as the CNN cannot learn in the context of the video. Another family of networks is 3D CNNs (Carreira & Zisserman, 2017 ; Tran *et al.*, 2015,1 ; Xie, Sun, Huang, Tu & Murphy, 2018). The major drawback is the number of trainable parameters. In (Tran *et al.*, 2018), an architecture called ResNet (2+1)D (R(2+1)D) uses skip connections as its 2D counterpart (He *et al.*, 2016). This leads to fewer parameters to optimize while retaining state-of-the-art performances. Additionally, spatial and temporal convolutions are separated by nonlinear activation functions to allow for a more complex representation.

Skeleton data are powerful (Johansson, 1973). But superiority against video data is unclear ; rather, they seem to be complementary (Main de Boissiere & Noumeir, 2020 ; Wang *et al.*, 2018a). First modern deep learning attempts gravitated toward various forms of RNNs (Liu *et al.*, 2016 ; Shahroudy *et al.*, 2016 ; Zhang *et al.*, 2017). Then skeleton to 2D image mapping with 2D CNNs yielded better results (Ke *et al.*, 2017b ; Kim & Reiter, 2017 ; Zhang *et al.*, 2019). Nowadays, graph convolutional networks hold the state of the art (Shi *et al.*, 2019a ; Yan *et al.*, 2018).

5.2.2 Early action prediction

Early action prediction follows the paradigm of action recognition, but on partially observed sequences. First attempts used handcrafted features in the form of representation of visual words (Kong, Kit & Fu, 2014), hierarchical movemes (Lan, Chen & Savarese, 2014) and histogram of spatiotemporal features (Ryoo, 2011).

Deep learning approaches can be separated into two categories : online and offline. Online approaches do not use the duration of a sequence as part of a preprocessing step. In (Ma *et al.*, 2016), a CNN+LSTM network is used. A novel loss is introduced which encourages a non-monotonic ascending prediction score across time. In (Sadegh Aliakbarian *et al.*, 2017), a CNN+LSTM network is used for very early prediction, but can at most study 50 frames. A similar architecture is used in (Kong, Gao, Sun & Fu, 2018), but is offline because of the bidirectional nature of the RNN.

The offline efforts either preprocess the temporal domain (Ke *et al.*, 2019), (Wang *et al.*, 2019), or the architecture uses information of the future (Ke *et al.*, 2019 ; Kong *et al.*, 2018 ; Pang *et al.*, 2019). In (Ke *et al.*, 2019), partial and full sequences are confronted in an adversarial learning context on handcrafted skeleton images. In (Wang *et al.*, 2019), a bidirectional CNN+LSTM teacher distills its information to a unidirectional student network. The architecture could be online, but a temporal normalization step is first performed.

5.2.3 Action detection

Action detection analyzes raw sequences and outputs temporal segments containing activities with a class prediction. The online framework emits a prediction frame by frame, or by short blocks of frames, without future context. In an offline scenario, a sequence is studied in its entirety before outputting predictions.

5.2.3.1 Offline action detection

Early efforts focused on handcrafted features and sliding windows of different sizes (Caba Heilbron *et al.*, 2016 ; Ni *et al.*, 2016 ; Wang *et al.*, 2014 ; Yuan *et al.*, 2016). Deep learning attempts follow the framework of the R-CNN family for object detection in images (Girshick, 2015 ; Girshick *et al.*, 2014 ; Ren *et al.*, 2015). The network outputs temporal proposals, ranks them, then classifies them. The architectures presented in (Chao *et al.*, 2018 ; Dai *et al.*, 2017 ; Gao *et al.*, 2017a,1 ; Xu *et al.*, 2017) combine those tasks in end-to-end fashion. In (Song, Lan, Xing, Zeng & Liu, 2018), proposal precedes classification via spatiotemporal attention LSTMs on skeleton data.

Closer to the online efforts, some architectures study sequences in a single stream flow. In (Buch *et al.*, 2017a) and (Buch *et al.*, 2017b) a 3D CNN+RNN architecture studies videos by chunks of δ frames. The network in (Buch *et al.*, 2017a) could be used in real time, but is limited by a fixed maximum proposal size and demanding post-processing.

5.2.3.2 Online action detection

De Geestet *et al.* outlined the challenges of online action detection (De Geest *et al.*, 2016) and later proposed a two-stream LSTM architecture (De Geest & Tuytelaars, 2018). The first stream interprets the input, the other the temporal dependencies between actions. In (Gao *et al.*, 2017c), a Reinforced Encoder-Decoder (RED) network uses a CNN feature extractor with an LSTM. The network is designed for anticipation, but can be used for online detection. In (Shou *et al.*, 2018), precise start time of an action is emphasized with adversarial networks. In (Xu *et al.*,

2019), a temporal recurrent network (TRN) is introduced with a prediction module. Using a pre-trained feature extractor yields good results in (Buch *et al.*, 2017a ; Gao *et al.*, 2017c ; Xu *et al.*, 2019), but we believe fine-tuning a network in the context of its new task leads to improved performances, as done in (Luo, Hsieh, Jiang, Carlos Niebles & Fei-Fei, 2018)

5.2.4 Knowledge Distillation

Knowledge distillation regroups techniques which aim to transfer the knowledge of a large pre-trained network to a smaller one. The concept was introduced in (Buciluǎ, Caruana & Niculescu-Mizil, 2006). Popularized in (Hinton, Vinyals & Dean, 2015), "softmax temperature" is proposed. The student network learns from both the ground truth, and smoothed soft labels from the teacher. Minimizing the mean square error (MSE) between student and teacher outputs is a possibility (Romero, Ballas, Kahou, Chassang, Gatta & Bengio, 2014). In (Yim, Joo, Bae & Kim, 2017), knowledge distillation shows a faster learning time for the student network, which eventually outperforms the teacher. In (Wang *et al.*, 2019), the loss function minimizes maximum mean discrepancy (MMD) between teacher and student for early action prediction. Our approach borrows elements from both transfer learning and knowledge distillation. Also, we shift the focus from offline teacher to online student. But, because we do not only reuse a network and attempt to distill the teacher's representation, we believe it fits into the knowledge distillation paradigm.

5.3 Action recognition to online action detection framework

We tackle multiple video understanding tasks, from human action recognition to online action detection, and propose a framework to link those together. An offline teacher is deployed for action recognition and early prediction. During training, intraclass cosine similarity and interclass distance are encouraged. Knowledge distillation is employed to transfer the representation of the teacher to an online early prediction student network. Distillation is done via reuse of layers and cosine similarity between teacher and student feature vectors at different time progressions. The student network is then transposed to an online action detection task. We introduce a sigmoid-weighted temporal loss to train the online networks.

5.3.1 Preprocessing

With online action detection as our final objective, we implement an online cropping strategy on the infrared frames, inspired by (Main de Boissiere & Noumeir, 2020). It is used across the different networks. Also, the teacher and student networks will study sequences differently. The teacher requires a temporal normalization step, while the students do not, which leads to different sampling strategies.

5.3.1.1 Cropping strategy

Because 3D CNNs embed a lot of trainable parameters, the video frame resolution is heavily downscaled to compensate. This results in a non-negligible loss of information. For example, a small object might become indistinguishable.

In a human action recognition context, the background provides little to no context regarding the activity performed. Thus, the projected 3D skeleton coordinates on the infrared frames can be used to create a region of interest around the subject(s). As such, we extract the maximal and minimal pixel values across all joints at each time frame to capture the subject in a bounding box (Figure 5.2).

Because a frame is cropped before the resizing operation, the downscaling factor is less important, resulting in less information loss. However, because the bounding box is recalculated for every frame, it moves across time. This results in a fake camera movement and different scaling factors across time, as shown Figure 5.2.

5.3.1.2 Sampling strategies

We define an offline and an online sampling strategy for the teacher and online networks respectively.

For action recognition, an action sequence is divided into T^{off} subwindows of equal sizes, as done in (Main de Boissiere & Noumeir, 2020). A random frame is sampled from each, creating

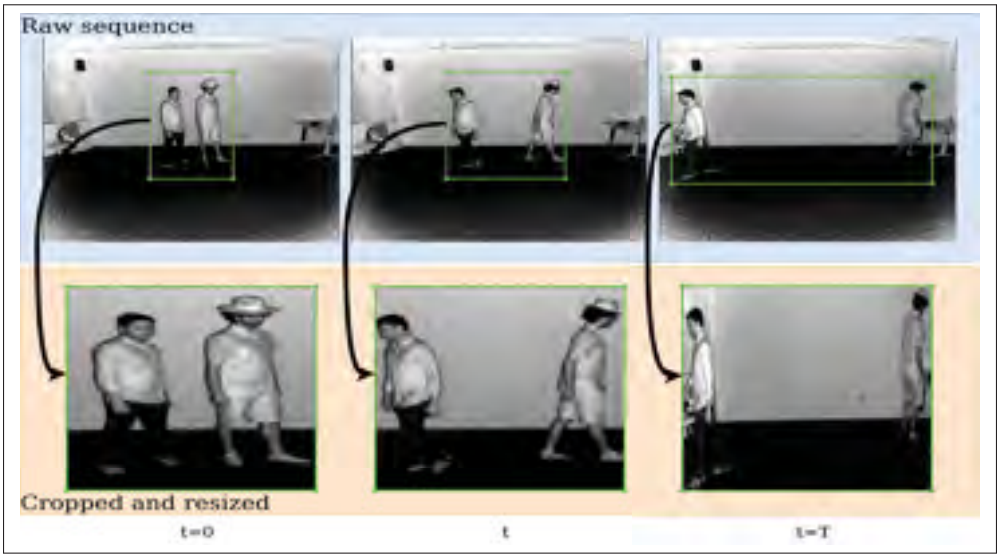


Figure 5.2 Online preprocessing on infrared videos. A crop around the subjects is performed using 2D skeleton data

a normalized sequence of size T^{off} which is fed to the teacher. For the early prediction task, the total number of frames N of a sequence is adjusted depending on the observation ratio r, i.e., the percentage of the sequence considered : $N_{partial} = \text{floor}(rN)$. For instance, for a sequence of $N = 100$ frames and an observation ratio of $r = 80\%$, we will consider the $N_{partial} = 80$ first frames. The same strategy is then used with the adjusted number of frames. If T^{off} is greater than $N_{partial}$, the first $N_{partial}$ frames are considered. The sequence is padded with black frames to reach a size T^{off}. For example, if T^{off} is set to 15, and $N_{partial} = 10$, then 5 back frames are added to get a normalized sequence of size T^{off}.

For the online networks, the sequences are studied in their entirety. The frame rate is reduced by a factor s, meaning we keep one frame every s frames.

5.3.2 Network architectures

The building blocks for our architectures are a 3D CNN, LSTMs and fully connected layers. They are reused across the different networks which improves both results and learning times.

5.3.2.1 Offline teacher

We use an 18-layer deep ResNet (2+1)D as our backbone (Xie *et al.*, 2018). The network is pre-trained on Kinetics-400 (Carreira & Zisserman, 2017). Our offline early prediction framework is summarized Figure 5.3. The R(2+1)D network outputs a feature vector of size 512 which we then normalize with a 1D batch normalization layer (Ioffe & Szegedy, 2015). Batch normalization allows for all feature vectors to have roughly the same Euclidean norm. As such, minimizing intraclass cosine similarity should also reduce the MSE between feature vectors. We call x^t the normalized teacher feature vector. From here on out, the superscript t refers the teacher, the subscript t refers to time. A final fully connected layer, the classifier, is used to output a softmax-normalized class probability distribution.

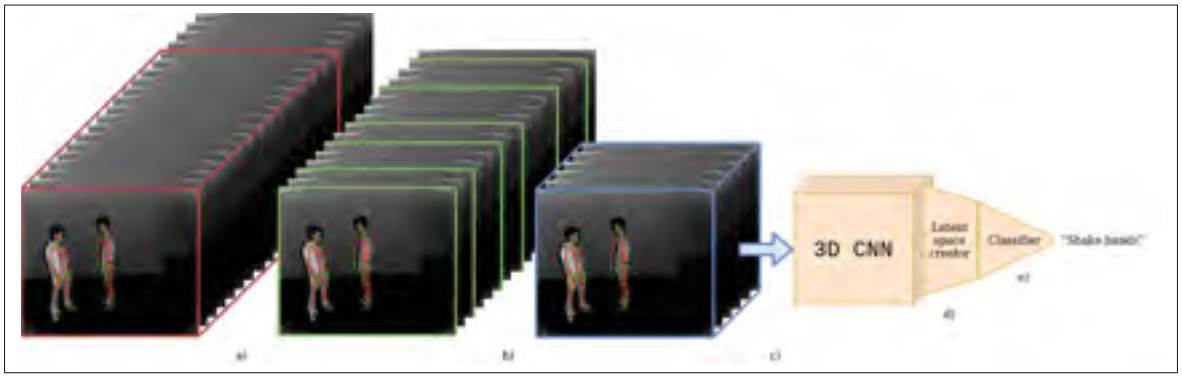


Figure 5.3 Teacher framework. a) A random observation ratio is used during training. b) T^{off} frames are sampled from T^{off} subwindows of even sizes. c) The normalized sequence is fed to a ResNet (2+1)D. d) The feature vector is normalized by a 1D batch normalization layer. e) The normalized teacher feature vector x^t is used for prediction

5.3.2.2 Online early prediction student

The online early prediction student network reuses the same R(2+1)D as the teacher but adds a classification LSTM network on top (Figure 5.4). For early prediction alone, it does not embed the actionness LSTM. A sequence is divided into T^{on} subsequences of δ frames, with $T^{on} = \text{ceil}(\frac{N}{\delta})$. For instance, with $\delta = 5$, a sequence of $N = 98$ frames, then $T^{on} = 20$. Each subsequence is fed to the R(2+1)D network. At each time frame $t \in \{1, .., T^{on}\}$, the computed

feature vector is used as input for the LSTM. The classification LSTM hidden vector h_t^c will then be fed to a "reconstruction" fully connected layer. From here on out, the superscript c refers to the classification LSTM, recall that the subscript t refers to time. We call the outputted vector x_t^c. During training, the goal will be to approximate x_t^t with x_t^c. Here, x_t^t is the feature vector outputted by the teacher with $N_{partial} = ts\delta$.

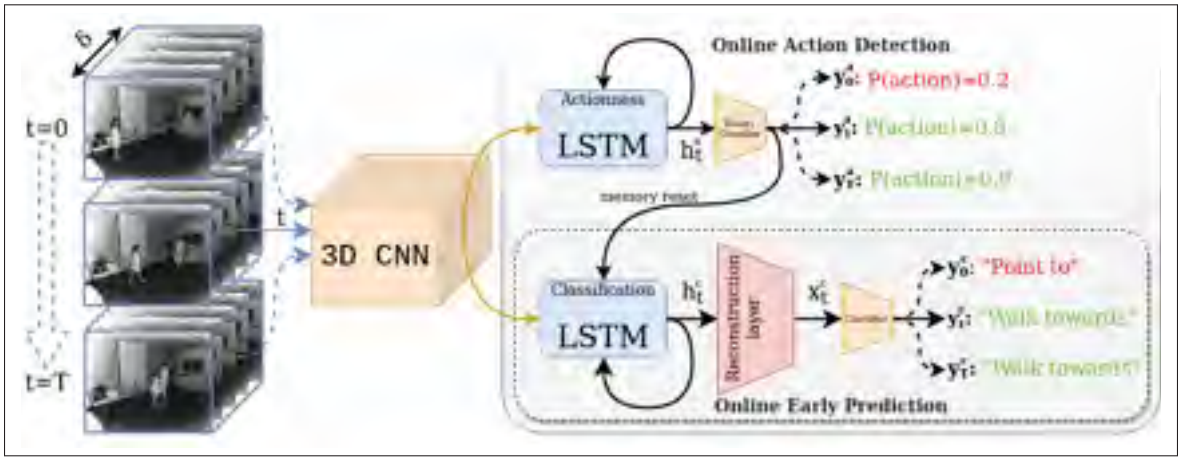


Figure 5.4 Student networks. A sequence is broken down into chunks of δ frames fed to a 3D CNN. The actionness LSTM proposes temporal segments. The classification LSTM outputs a prediction when an action is being detected. The network can be used for online early action prediction with the classification LSTM only

5.3.2.3 OKDAD student

We introduce the Online Knowledge Distillation Action Detection (OKDAD) network (Figure 5.4). It builds upon the previous network with an additional actionness LSTM layer. The actionness module has two roles. Firstly, it proposes temporal segments as the sequence happens. Secondly, it resets the cell and hidden state vectors of the classification LSTM. Intuitively, it forces the classification LSTM to forget about the past once an action is over. As such, when a new action is discovered, the classification LSTM is reduced to an online early prediction task.

The hidden state vector of the actionness LSTM h_t^a (the superscript a refers to the actionness LSTM) is fed to a fully connected layer. Its output is followed by the sigmoid function, to predict the probability that the current frame block contains an action. With y_t^a the actionness

probability, the classification LSTM is updated as follows, with c^c the cell state vector of the classification LSTM :

$$h^c_{t-1} := y^a_t * h^c_{t-1} \tag{5.1}$$

$$c^c_{t-1} := y^a_t * c^c_{t-1} \tag{5.2}$$

5.3.3 Knowledge distillation

5.3.3.1 Teacher loss

Recall that we aim to distill teacher knowledge to an online action detection network. We hypothesize that the reconstruction task of the student will be easier if the teacher's feature vectors have low intraclass and high interclass distance. We introduce loss terms based on cosine similarity, as we believe it is more appropriate than MSE for that task. In essence, it is more practical to "push apart" vectors of different classes with cosine similarity. Additionally, because the batch normalization layer implies that feature vectors have roughly the same Euclidean norm, encouraging cosine similarity for vectors of same classes should also reduce MSE, without explicitly penalizing it. As such, we propose a novel loss function :

$$\begin{aligned}\text{loss} = r^\gamma [& \frac{1}{|B|} \sum_{i=1}^{|B|} H(\hat{y}_i, y^t_i) \\ & + \alpha \frac{1}{\sum_{i=1}^{|B|} \sum_{j=i+1}^{|B|} \mathbb{1}_{\hat{y}_i = \hat{y}_j}} \sum_{i=1}^{|B|} \sum_{j=i+1}^{|B|} -\mathbb{1}_{\hat{y}_i = \hat{y}_j} \ln\left(\frac{\cos(\angle(x^t_i, x^t_j)) + 1}{2}\right) \\ & + \beta \frac{1}{\sum_{i=1}^{|B|} \sum_{j=i+1}^{|B|} \mathbb{1}_{\hat{y}_i \neq \hat{y}_j}} \sum_{i=1}^{|B|} \sum_{j=i+1}^{|B|} -\mathbb{1}_{\hat{y}_i \neq \hat{y}_j} \ln\left(1 - \frac{\cos(\angle(x^t_i, x^t_j)) + 1}{2}\right)]\end{aligned} \tag{5.3}$$

With $\angle$ the angle between two vectors, $\hat{y}$ the ground truth, y^t the teacher prediction, x^t the normalized teacher feature vector, H the cross-entropy loss, r the observation ratio, $|B|$ the number of sequences in a batch, α, β and γ three hyperparameters.

The loss consists of three terms. The cross-entropy loss term encourages correct predictions with high confidence. The similarity loss term reduces cosine similarity between same-class vectors in a batch. The distance loss term decreases cosine similarity between different-class vectors in a batch. Hyperparameters α and β weigh the importance of similarity and distance loss terms respectively. Hyperparameter γ, taken from $]0, +\infty[$, sets the importance of smaller observation ratio. The smaller the value, the more penalty is applied.

5.3.3.2 Online early prediction student loss

We want our student network to mimic the teacher feature vector x_t^t generated at different time steps $t \in \{1, .., T^{on}\}$. We propose a novel loss function that allows the network to learn autonomously while being guided when the teacher outputs a correct prediction :

$$\text{loss} = \frac{1}{|B|} \sum_{i=1}^{|B|} L_s(\hat{y}_i, y_i^c, x_i^t, x_i^c) \tag{5.4}$$

$$\text{loss} = \frac{1}{|B|} \sum_{i=1}^{|B|} \left[\sum_{t=1}^{T_i^{on}} \sigma_{i,t} [H(\hat{y}_{i,t}, y_{i,t}^c) - \eta(1 - \epsilon_{i,t}) \ln(\frac{\cos(\angle(x_{i,t}^t, x_{i,t}^c)) + 1}{2})] \right] \tag{5.5}$$

Here, ϵ_t is the teacher error, such that $\epsilon_t = 1 - P(y_t^t = \hat{y})$ where $P(y_t^t = \hat{y})$ is the softmax-generated probability from the teacher for the correct label. Weighted arithmetic mean coefficients are taken evenly from the sigmoid function in the $[-2, 2]$ range : $\sigma_t = \frac{2}{T^{on}} \sigma(4 \frac{t-1}{T^{on}-1} - 2)$ and $\sum_{t=1}^{T^{on}} \sigma_t = 1$. The network is thus less penalized on early outputs. Notice that T^{on} is variable. If $T_i^{on} = 1$, then we take $\sigma_{i,t} = 1$. The hyperparameter η weighs the contribution of the teacher.

The loss consists of two terms. The first one dynamically penalizes classification based on total sequence length. The second one guides the student reconstruction vector to the teacher normalized feature vector. To further encourage similar representations, the teacher classification layer is reused by the student, and frozen during training.

5.3.3.3 OKDAD student loss

During training of the OKDAD network, two tasks are optimized. The first is the temporal segment propositions. It is assured by minimizing the binary cross-entropy H between predicted and true actionness at each frame bloc. The second is early prediction for each action in a studied sequence, with A a set of actions and $|A|$ its cardinality. To do so, our sigmoid weighted temporal loss L_s is reused. In other words, we consider each instance of an action in a long sequence as an individual online early prediction task. We propose the following novel loss function :

$$\text{loss} = \frac{1}{|B|}\sum_{i=1}^{|B|}\left[\frac{1}{T}\sum_{t=1}^{T} H(\hat{y}^a_{i,t}, y^a_{i,t}) + \sum_{a=1}^{|A_i|} L_s(\hat{y}^c_{i,a}, y^c_{i,a}, x^t_{i,a}, x^c_{i,a})\right] \tag{5.6}$$

To summarize, the progression from action recognition to online action detection is as follows. First, an action detection dataset is reduced to an offline early prediction task which a teacher network performs. The knowledge is distilled to an online action detection network, which we showed can be considered as an online early prediction task doubled with temporal action proposals.

5.4 Experiments

We test our method on infrared videos from RGB+D human action datasets : NTU RGB+D (Shahroudy *et al.*, 2016) and PKU-MMD (Liu *et al.*, 2017a). To the best of our knowledge, they are the only RGB+D datasets proposing the infrared stream. NTU RGB+D is a human action recognition dataset on which we evaluate our offline teacher and the classification performances

of our OKDAD architecture. PKU-MMD is a human action detection dataset, on which we evaluate both the temporal action proposal and classification tasks of OKDAD.

We focus our efforts on the online classification task. We believe a strong online early prediction network should naturally perform well for temporal action propositions, as it can be seen as a binary classification task.

5.4.1 Implementation details

An R(2+1)D network (Xie *et al.*, 2018) is used across all networks. It outputs a 1D feature vector of size 512. For the students, the classification and actionness RNNs are single LSTMs with 2048 and 1024 features in the hidden state respectively. The reconstruction layer is a single linear layer with an input size of 2048 and an output size of 512.

During teacher training, we sample the sequence ratio r uniformly between $[r_{min}, 1]$ with a 0.5 probability. Otherwise, the ratio is set to 1. We use $r_{min} = 0.025$ and use at least one frame. This favors training on entire sequences and means we study at least 2.5% of a sequence. For best results, we use $\alpha = 1$, $\beta = 0.5$ and $\gamma = \frac{2}{3}$. We set $T^{off} = 15$ and train with a batch size of 16 and a learning rate of $1e^{-4}$.

For online early prediction student training, we sample every s frames from the entire sequences, with $s = 3$. Each subsequence contains $\delta = 5$ frames, as such $s\delta = T^{off} = 15$. We fix $T^{on} = \text{ceil}(\frac{N_{max}}{s\delta})$ for all sequences, with N_{max} the maximum number of frames in a sequence across the entire dataset. Thus, every sequence can be studied in its entirety while training by batches. Shorter sequences are padded with black frames. Because the student loss (equation 5.4) is dynamic, the predictions for the black frames are not penalized. We freeze the first 17,137,693 trainable parameters of the R(2+1)D network and fine-tune the subsequent 14,162,432. As such, the 3D CNN backbone can shift its purpose from global to local feature extractor. The classifier, reused from the teacher, is also frozen. We set hyperparameter $\eta = 1$, batch size to 32 and learning rate to $1e^{-3}$.

For OKDAD training, we keep s, δ, η, batch size, learning rate and frozen parameters identical as above. We randomly sample subsequences of $T = 40$ frame blocs from a long sequence, which equals to 20 seconds. When evaluating the offline action detection performances of OKDAD, predicted actionness with probabilities over 0.75 are considered positive. Temporal segments are created from adjacent positive predictions. Also, sequences are evaluated in their entirety to mimic a real-time scenario.

For the online networks, we calculate the offline classification accuracy by weighing the predictions at different time steps with the same sigmoid weights from equation 5.4, with T^{on} adapted to the observation ratio.

Adam optimizer (Kingma & Ba, 2014) is used systematically across all training.

5.4.2 NTU RBG+D human action recognition dataset

Tableau 5.1 Early prediction results on NTU-RGB+D (accuracy in %)

Observation ratio	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**	**Avg.**
KNN (Hu *et al.*, 2018a)	7.4	9.5	12.2	16.0	20.8	25.9	30.8	34.4	36.1	37.0	23.01
RankLSTM (Ma *et al.*, 2016)	11.5	16.4	25.6	37.7	47.9	55.9	60.9	64.4	66.0	65.9	45.22
DeepSCN (Kong, Tao & Fu, 2017)	16.8	21.4	30.5	39.9	48.7	54.6	58.1	60.1	60.0	58.6	44.87
MSRNN (Hu *et al.*, 2018a)	15.1	20.3	29.5	41.3	51.6	59.1	63.9	67.3	68.8	69.2	48.61
PTSL (Wang *et al.*, 2019)	27.8	**35.8**	46.2	58.4	67.4	73.8	77.6	80.0	81.4	82.0	63.04
DBNet (Pang *et al.*, 2019)	**27.9**	33.3	47.2	56.9	68.5	74.5	78.5	80.5	81.6	81.5	63.04
Teacher	10.5	29.5	**49.9**	**66.7**	**76.0**	**81.1**	**83.9**	**85.4**	**86.3**	86.4	**65.57**
Student	19.3	28.1	38.6	55.5	67.9	75.5	80.6	83.7	85.5	**86.6**	62.13

NTU RGB+D is a state-of-the-art human action recognition dataset (Shahroudy *et al.*, 2016). It contains 60 action classes, from daily activities to medical conditions, spread across 56,880 clips, 40 subjects and 80 views. Each action contains up to two subjects. Captured from different views and setups, the great diversity makes NTU RGB+D a challenging dataset. Accuracy is used as the evaluation metric.

There are two benchmark evaluations : cross-subject (CS) and cross-view (CV). Following previous early prediction works on this dataset (Pang *et al.*, 2019), (Wang *et al.*, 2019), we only

consider the CS benchmark, which splits training and test sets across different subjects. We sample 5% of our training set as our validation set, as in (Shahroudy *et al.*, 2016).

The results of our networks for different ratios are presented Table 5.1. The strong performances of the teacher are outlined. For ratios greater than 20%, the network consistently outperforms the previous state of the art by a large margin. At 50%, accuracy is improved by 7.5%, for a total of 76.0%. At 100%, the accuracy is improved by 4.9%. On average, the teacher performs 2.5% better than best previous attempts. For very early predictions (less than or equal to 20%), the teacher underperforms. At 10%, we note a 17.4% difference, and 6.3% at 20% compared to (Pang *et al.*, 2019).

Additionally, we report the offline performances of our online early prediction student. Best results are achieved with $\eta = 1$ for the student with $\alpha = \beta = 0$ for the teacher. The student is not able to match the performance of the teacher, except for an observation ratio of 100% with 86.6% for the student. Nonetheless, the student performs on par with the current state of the art, systematically outperforming for ratios greater than 50%. At 60%, the student performs 1% better than its closer competitor (with 75.5%) and 5.1% better at 100% (with 86.6%).

5.4.3 PKU-MMD action detection dataset

Tableau 5.2 Action detection results on PKU-MMD (in mAP_a)

	Cross-Subject			
θ	**0.1**	**0.3**	**0.5**	**0.7**
RS+DR+DOF (Liu *et al.*, 2017a)	0.647	0.476	0.199	0.026
CS+DR+DOF (Liu *et al.*, 2017a)	0.649	0.471	0.199	0.025
TAP-B (Song *et al.*, 2018)	0.544	0.514	0.461	0.327
TAP-B-M (Song *et al.*, 2018)	0.557	0.53	0.431	0.242
RGB+D+F+S (Luo *et al.*, 2018)	0.903	0.895	0.833	-
OKDAD $\alpha = 1, \beta = 0.5, \eta = 0$	0.899	0.886	0.822	0.644
OKDAD $\alpha = 1, \beta = 0.5, \eta = 1$	0.908	0.893	0.826	0.651
OKDAD $\alpha = 0, \beta = 0, \eta = 0$	0.912	0.897	0.842	0.672
OKDAD $\alpha = 0, \beta = 0, \eta = 1$	**0.915**	**0.905**	**0.850**	**0.679**

PKU-MMD is an RGB+D offline action detection dataset (Liu *et al.*, 2017a). It contains 1,076 long sequences with approximately 20 actions from 51 classes per instance. It totals 21,545 actions performed by 66 subjects. Mean Average Precision of actions (mAP_a) is used as the evaluation metric. We evaluate the offline performances of OKDAD on the cross-subject benchmark.

Table 5.2 shows the performances of our OKDAD network for different temporal intersection over union thresholds θ. Our method outperforms previous attempts for all thresholds. Especially for $\theta = 0.7$, OKDAD does not experience a similar drop in performance as (Liu *et al.*, 2017a) and (Song *et al.*, 2018). This suggests a considerable improvement in accurate temporal boundaries detection, while still performing online.

We evaluate OKDAD with two different teachers, with and without knowledge distillation for each. We find using a teacher without cosine similarity penalties yields better results. However, in both cases, encouraging cosine similarity between student and teacher vectors ameliorates results noticeably. This is in line with our findings section 5.4.4.3.

Videos demonstrating the OKDAD network in action are available in the supplementary materials.

5.4.4 Ablation studies

We provide more experiments to better understand the different contributions of our knowledge distillation framework.

5.4.4.1 Cosine similarity penalties on teacher learning

During teacher training, low intraclass and high interclass distance are encouraged via cosine penalty loss terms. In Table 5.3, we compare the impact of the similarity loss terms on the test set. A "no penalty" ($\alpha = \beta = 0$) teacher is compared to its "cos penalty" ($\alpha = 1, \beta = 0.5$) counterpart. Average intraclass and interclass cosine similarity are reported for different observation ratios. A value closer to 1 means similar orientation, closer to -1 means diametrically opposed.

Tableau 5.3 Impact of cosine penalty on teacher network. Accuracy in % (Acc.), intraclass (Intra) and interclass (Inter) cosine similarity are reported

Observation ratio		**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**	**Avg.**
No penalty	Acc.	8.9	**31.3**	**51.6**	66.7	74.7	79.5	82.6	84.0	84.6	84.8	64.87
	Intra	**0.80**	0.49	0.44	0.46	0.50	0.53	0.55	0.56	0.57	0.57	0.55
	Inter	0.83	0.30	0.11	0.08	0.09	0.10	0.11	0.12	0.13	0.13	0.20
Cos penalty	Acc.	**10.5**	29.5	49.9	66.7	**76.0**	**81.1**	**83.9**	**85.4**	**86.3**	**86.4**	**65.57**
	Intra	0.78	**0.56**	**0.59**	**0.68**	**0.74**	**0.78**	**0.80**	**0.82**	**0.82**	**0.82**	**0.74**
	Inter	**0.73**	**0.25**	**0.10**	**0.07**	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**	**0.15**

The loss terms have the desired effects. Intraclass similarity is considerably increased by 0.19 points on average. The interclass similarity is adequately decreased by 0.05 points on average. But it is interesting to note that the network pushes interclass vectors toward orthogonality without an explicit penalty.

The cosine similarity loss terms also improve the accuracy scores, notably on observation ratios greater than 40%. On average, accuracy is improved by 0.7% and by about 1.5% for ratios greater than 40%.

5.4.4.2 Teacher layer reuse on online early prediction student

Our distillation scheme consists of layer reuse and guided learning. Table 5.4 shows the impact of reusing teacher layers only ($\eta = 0$) compared to our baseline, i.e., the same network without layer reuse and teacher guidance. Note that for the baseline, all 3D CNN parameters are frozen, as the network would not converge otherwise. We use two different teachers. The first one with cosine similarity penalties ($\alpha = 1, \beta = 0.5$), the second one without.

Compared to the baseline, average accuracy nearly doubles (30.84% to about 60% for both students). This clearly demonstrates the importance of an appropriate feature extractor.

We also compare the cosine similarity and the MSE with the teacher at different observation ratios, even if the student is not guided. It can be observed that a penalized teacher naturally

Tableau 5.4 Contribution of layer reuse and knowledge distillation with different teachers on student. Accuracy in % (Acc.), average cosine similarity (Sim.) and MSE between teacher and student feature vectors are reported

Observation ratio			**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**	**Avg.**
Baseline		Acc.	8.9	10.8	14.3	21.5	30.0	37.2	42.4	45.8	48.3	49.2	30.84
Cos teacher	$\eta = 0$	Acc.	18.5	26.3	37.4	53.9	67.0	73.9	78.5	81.7	83.4	84.8	60.54
		Sim.	0.94	0.91	0.83	0.79	0.77	0.77	0.76	0.75	0.75	0.75	0.80
		MSE	0.14	0.20	0.32	0.41	0.50	0.56	0.62	0.67	0.72	0.76	0.49
	$\eta = 1$	Acc.	18.0	27.4	36.9	52.3	65.8	73.8	79.1	82.5	84.4	85.5	60.57
		Sim.	0.97	0.95	0.92	0.91	0.92	**0.93**	**0.93**	**0.93**	**0.94**	**0.94**	**0.93**
		MSE	**0.09**	**0.11**	**0.17**	**0.20**	**0.21**	**0.24**	**0.27**	**0.30**	**0.33**	**0.36**	**0.23**
Raw teacher	$\eta = 0$	Acc.	18.0	27.3	37.3	53.7	66.3	73.9	79.0	82.5	84.2	85.3	60.75
		Sim.	0.94	0.90	0.80	0.74	0.70	0.69	0.68	0.68	0.67	0.67	0.75
		MSE	0.63	0.71	0.87	0.81	0.83	0.85	0.86	0.89	0.91	0.93	0.83
	$\eta = 1$	Acc.	**19.3**	**28.1**	**38.6**	**55.5**	**67.9**	**75.5**	**80.6**	**83.7**	**85.5**	**86.6**	**62.13**
		Sim.	**0.98**	**0.97**	0.92	0.89	0.88	0.87	0.87	0.87	0.87	0.86	0.90
		MSE	0.48	0.51	0.54	0.43	0.40	0.40	0.41	0.42	0.44	0.46	0.45

increases similarity and decreases MSE between feature vectors. This is in line with our hypothesis. However, this does not seem to impact average performance as both student networks perform similarly. Interestingly, the student reusing the layers of the unpenalized teacher performs marginally better, especially in the later stages, even though the latter is less accurate.

5.4.4.3 Knowledge distillation on online early prediction student

Table 5.4 also shows the contribution of the guidance using the same teachers as above.

As hypothesized, increasing cosine similarity between teacher and student feature vectors also reduces MSE. For the student with a penalized teacher, similarity increases by 0.13 points on average and MSE reduces by half (0.49 to 0.23). Similar proportions can be observed using the unpenalized teacher.

However, the increase in performances is marginal for the student with the penalized teacher (60.54% to 60.57% on average). More convincingly, the distillation scheme increases per-

formances by 1.4% on average using the unpenalized teacher, more so for the later stages. Additionally, the student is able to beat all teachers with that configuration for entire sequences.

Indeed, the penalized teacher feature vectors are more convincingly approximated, but the tighter clusters may also leave less room for error, explaining the lesser performances of the student in that case. Future attempts could focus on also reducing MSE explicitly, as this metric, although improved by our distillation scheme, could be further ameliorated.

5.5 Conclusion

We propose a framework to link action recognition and online action detection together. An action recognition architecture is used for early prediction. Knowledge distillation and layer reuse are employed to enhance the performances of an online early prediction network. We finally add an online temporal action proposal module to this architecture (OKDAD) for action detection. Our method achieves state-of-the-art results on action recognition NTU RBG+D and action detection PKU-MMD datasets. We conclude to the efficacy of knowledge distillation for networks performing different tasks. However, it seems conceivable that the student networks should eventually outperform the teachers, as they study sequences in their entirety. As such, future works are encouraged.

CONCLUSION ET RECOMMANDATIONS

Ce mémoire s'inscrit dans une volonté d'automatiser la détection de comportements dangereux dans les milieux carcéraux. Le suicide est la cause principale de mortalité en prison avec 70 morts pour 100 000 détenus par an au Canada. Les méthodes actuelles de détection de dernier recours sont insuffisantes. La surveillance par télévision en circuit fermé est inefficace et intrusive. Il en est de même pour des capteurs de signes vitaux qu'un détenu devrait porter. Les méthodes prédictives à long terme ne sont pas assez puissantes, quand bien même, elles ne peuvent prédire le moment exact d'une tentative. Deux pistes de détection en temps réel de comportements dangereux sont actuellement à l'étude : la détection de signes vitaux par radars et la reconnaissance automatisée par vidéosurveillance. Nos efforts se concentrent sur cette dernière, en proposant une nouvelle méthode. Nous obtenons des résultats supérieurs à l'état de l'art sur deux bases de données publiques : une de reconnaissance d'activités, une autre de détection d'activités humaines.

Le suicide par sectionnement est complexe. Il peut se manifester sur différentes zones du corps avec une tendance prononcée pour les poignets, les coudes, les bras, et le torse. Ce sont des mouvements fins, qui sont difficiles à discerner à l'aide d'un squelette 3D des coordonnées des articulations humaines. Généralement, plusieurs coupures sont réalisées, ce qui est un indice de plus pour un système temps réel efficace. Moins fréquent et moins meurtrier qu'un suicide par pendaison, il n'en reste pas moins problématique.

La reconnaissance d'activités humaines par apprentissage machine a connu de profondes avancées cette dernière décennie. L'avènement des caméras RGB+D grand public encourage l'utilisation de ce support. Sur les bases de données publiques RGB+D, les résultats avoisinent maintenant le sans-faute. L'architecture profonde que nous proposons chapitre 3 obtient des résultats dignes de l'état de l'art actuel sur la plus grosse et complexe base de données de reconnaissance d'activités humaines RGB+D.

L'infrarouge se démarque particulièrement. Son pouvoir de représentation est réellement intéressant pour détecter des actions impliquant des objets sans être limité par les conditions d'éclairage. Avec en plus le squelette 3D des articulations humaines, les performances sont encore meilleures. Mais les résultats sont tels qu'un système utilisant uniquement l'infrarouge est envisageable.

Il n'existe pas de base de données publique contenant des simulations de suicide. Mais il est raisonnable de supposer que certaines actions de la base de données NTU RBG+D sont aussi, voire plus, difficiles à reconnaître. Nous sommes certains que notre réseau transférera bien ses connaissances pour une application plus spécifique.

De fortes performances en reconnaissance d'activités humaines peuvent être distillées vers un système de détection *online*. Le cadre de travail proposé, joignant reconnaissance anticipée et détection par distillation de connaissances, aboutit à des résultats dignes de l'état de l'art sur deux bases de données de référence (NTU RGB+D et PKU-MMD).

Néanmoins, une architecture profonde reste une boite noire peu interprétable. Les réseaux proposés ne donnent pas de degré d'incertitude ni de précision sur les éléments conduisant à telle ou telle décision. Pour des applications de sécurité, il apparaît essentiel de travailler vers des architectures compréhensibles. Des architectures bayésiennes semblent prometteuses.

ANNEXE I

DOCUMENTATION ET REPRODUCTIBILITÉ DES CODES

L'annexe ci-présente est un extrait de la documentation du dépôt *GitHub* des codes de l'architecture présentée chapitre 3. Les codes sont disponibles ici : https://github.com/adeboissiere/FUSION-human-action-recognition.

1. Project Organization

2. Getting started

The first step to replicate our results is to clone the project and create a virtual environment using the Makefile. After that, the raw data should be downloaded and placed according to the default Project Organization provided above. Then various h5 datasets will have to be created using the Makefile. This will take a while but is a more practical way of handling data. Once the h5 files are created, you are ready to replicate our results or use them to implement your own models !

The data used comes from the NTU RGB+D dataset [1].

1. Clone project

   ```
   git clone \
   https://github.com/adeboissiere/FUSION-human-action-recognition
   ```

2. Create virtual environment

   ```
   make create_environment
   ```

3. Activate environment (do so every time you work on this repository)

   `workon fusion` (for virtual env wrapper)

   `source activate fusion` (for conda)

[1] <http://rose1.ntu.edu.sg/datasets/actionrecognition.asp>

```
├── LICENSE
├── Makefile           <- Makefile with commands like `make data` or `make train`
├── README.md          <- The top-level README for developers using this project.
├── data
│   ├── processed      <- The final, canonical data sets for modeling.
│   │   ├── missing_skeleton.txt
│   │   ├── samples_names.txt
│   │   ├── ir.h5 (1TB)
│   │   ├── ir_cropped.h5 (157.8GB)
│   │   ├── ir_cropped_moving.h5 (60.8GB | optional)
│   │   ├── ir_skeleton.h5 (2.0GB)
│   │   ├── skeleton.h5 (3.0GB)
│   │
│   └── raw            <- The original, immutable data dump.
│       ├── nturgb+d_ir          <- Raw IR videos (*.avi)
│       ├── nturgb+d_skeletons   <- Raw skeleton files (*.skeleton)
│
├── docs               <- Sphinx generated documentation
│   ├── source         <- The final, canonical data sets for modeling.
│
├── models             <- Trained models and paper results.
│
├── notebooks          <- Jupyter notebooks. Naming convention is a number (for ordering),
│                         the creator's initials, and a short `-` delimited description, e.g.
│                         `1.0-jqp-initial-data-exploration`.
│
├── paper_cmds.txt     <- All commands used to replicate paper results
│
├── references         <- Data dictionaries, manuals, and all other explanatory materials.
│
├── reports            <- Generated analysis as HTML, PDF, LaTeX, etc.
│   └── figures        <- Generated graphics and figures to be used in reporting
│
├── requirements.txt   <- The requirements file for reproducing the analysis environment, e.g.
│                         generated with `pip freeze > requirements.txt`
│
├── run_cc.sh          <- Shell script to train on Compute Canada cluster
│
│
├── setup.py           <- makes project pip installable (pip install -e .) so src can be imported
├── src                <- Source code for use in this project.
│   ├── __init__.py    <- Makes src a Python module
│   │
│   ├── data           <- Create h5 datasets (see make data documentation)
│   │   └── make_dataset.py
│   │
│   ├── models         <- Scripts to train models and then use trained models to make
│   │   │                 predictions
│   │   ├── predict_model.py
│   │   └── train_model.py
│
└── tox.ini            <- tox file with settings for running tox; see tox.testrun.org
```

Figure-A I-1 Project organization

4. Install requirements

```
make requirements
```

5. Download raw data from the NTU RGB+D website [2], decompress archives and place files as described in the Project Description above.
6. Run the `make data` commands to create the h5 files.

3. Make h5 datasets

Our FUSION model uses both pose and IR data. It is modular, so it is possible to start with one or the other (see the Train model section), but best results are achieved using a combination of the two. To replicate all results from the paper, all h5 datasets (except *ir_cropped_moving.h5*) must be created. Assuming the Project Organization is kept, here are the commands to create the different datasets. The files are **quite heavy**, check the Project Organization for exact sizes.

- **IR 2D skeleton dataset**

```
make data DATASET_TYPE=IR_SKELETON
```

- **3D skeleton dataset**

```
make data DATASET_TYPE=SKELETON
```

- **Raw IR sequences dataset**

```
make data DATASET_TYPE=IR
```

- **Cropped IR sequences dataset** (requires ir.h5 and ir$_skeleton.h5$)

```
make data DATASET_TYPE=IR_CROPPED \
DATA_PATH="./data/processed/"
```

It is not mandatory to keep raw and processed data in the *./data/* folder, though highly encouraged. They could be in a different location (ie. an external drive). However, **it is crucial to keep the h5 file names the same**. Should the data be in a different folder, you will have to specify

[2] <http://rose1.ntu.edu.sg/datasets/actionrecognition.asp>

the input (DATA_PATH) and output path (PROCESSED_DATA_PATH). Check the project documentation [3] (**src.data** module) for details.

4. Train model

After the necessary h5 have been generated, it is time to test our FUSION model. To do so, use the `make train` command with the different hyperparameters. Below is an example of how to use the command, assuming the Project Organization is kept. For the commands used to obtain the results from the paper, check *paper_cmds.txt* in the root folder.

Note that a folder with a unique name will be created upon calling the command.

```
make train \
    EVALUATION_TYPE=cross_subject \
    MODEL_TYPE=FUSION \
    USE_POSE=True \
    USE_IR=True \
    PRETRAINED=True \
    USE_CROPPED_IR=True \
    LEARNING_RATE=1e-4 \
    WEIGHT_DECAY=0.0 \
    GRADIENT_THRESHOLD=10 \
    EPOCHS=15 \
    BATCH_SIZE=16 \
    ACCUMULATION_STEPS=1 \
    SUB_SEQUENCE_LENGTH=20 \
    AUGMENT_DATA=True \
    EVALUATE_TEST=True \
    SEED=0
```

[3] <https://adeboissiere.github.io/FUSION-human-action-recognition/>

If the h5 files are not in the default location, you need to specify the `PROCESSED_DATA_PATH` variable. If you would like to save the model elsewhere (default is *./data/models/*), you need to specify the `MODEL_FOLDER` variable.

Check the documentation [4] for more information on the **src.model** module and the make train command.

5. Plot confusion matrix

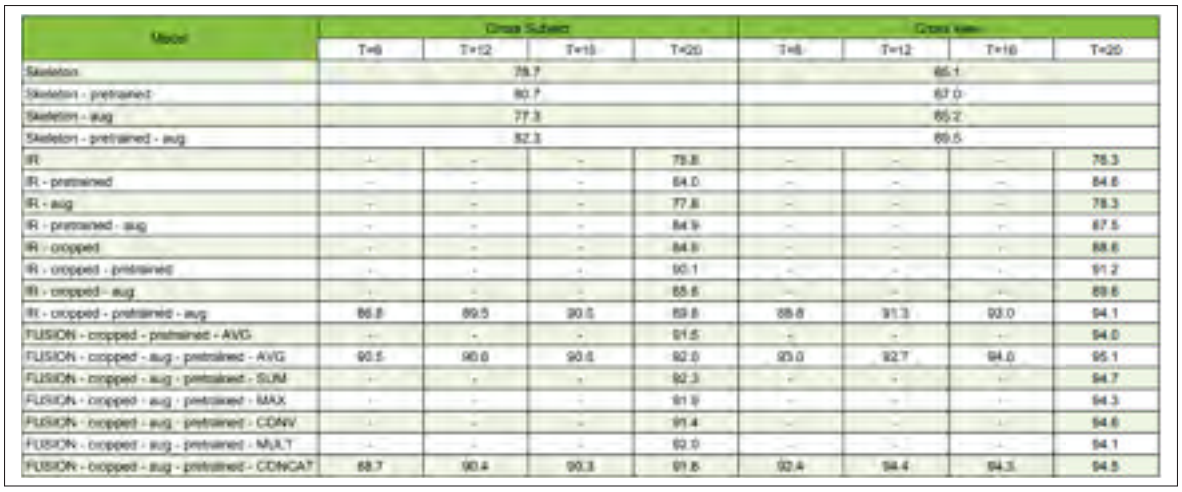
Once the model is trained, the confusion matrix is a great tool to understand where the model struggles. We provide a command to generate a .png image from a trained model. Below is an example of how to do so.

```
make confusion_matrix \
    MODEL_FOLDER="./models/trained_model_folder/" \
    MODEL_FILE="model12.pt" \
    EVALUATION_TYPE=cross_subject \
    MODEL_TYPE=FUSION \
    USE_POSE=True \
    USE_IR=True \
    USE_CROPPED_IR=True \
    BATCH_SIZE=1 \
    SUB_SEQUENCE_LENGTH=20
```

6. Results

Below is a summary of the results from the paper. We achieve state-of-the-art results. The log files of the training can be found in the *./models/fusion_test_tube_seed=0/* folder.

[4] <https://adeboissiere.github.io/FUSION-human-action-recognition/>



Model	Cross Subject				Cross view			
	T=8	T=12	T=16	T=20	T=8	T=12	T=16	T=20
Skeleton	78.7				85.1			
Skeleton - pretrained	80.7				87.0			
Skeleton - aug	77.3				85.2			
Skeleton - pretrained - aug	82.3				89.5			
IR	-	-	-	78.8	-	-	-	78.3
IR - pretrained	-	-	-	84.0	-	-	-	84.6
IR - aug	-	-	-	77.8	-	-	-	78.3
IR - pretrained - aug	-	-	-	84.9	-	-	-	87.5
IR - cropped	-	-	-	84.9	-	-	-	88.6
IR - cropped - pretrained	-	-	-	90.1	-	-	-	91.2
IR - cropped - aug	-	-	-	85.8	-	-	-	89.8
IR - cropped - pretrained - aug	86.8	89.5	90.5	89.8	88.8	91.3	93.0	94.1
FUSION - cropped - pretrained - AVG	-	-	-	91.5	-	-	-	94.0
FUSION - cropped - aug - pretrained - AVG	90.5	90.0	90.6	92.0	93.0	92.7	94.0	95.1
FUSION - cropped - aug - pretrained - SUM	-	-	-	92.3	-	-	-	94.7
FUSION - cropped - aug - pretrained - MAX	-	-	-	91.9	-	-	-	94.3
FUSION - cropped - aug - pretrained - CONV	-	-	-	91.4	-	-	-	94.6
FUSION - cropped - aug - pretrained - MULT	-	-	-	92.0	-	-	-	94.1
FUSION - cropped - aug - pretrained - CONCAT	88.7	90.4	90.3	91.6	92.4	94.4	94.3	94.5

Figure-A I-2 Results summary

7. Documentation

The project's code documentation is available here [5]. Alternatively, should you need the documentation locally or update it, follow these steps (from root directory) :

```
cd docs
# Remove previous build
make clean
# Remove previous generated .rst files
rm -f source/modules.rst source/src.*
# Generate .rst files from .py source files
sphinx-apidoc -o ./source ../src
# Build html files
sphinx-build -b html ./source build
```

[5] <https://adeboissiere.github.io/FUSION-human-action-recognition/>

ANNEXE II

TUTORIEL

Cette annexe inclut un *Jupyter Notebook* avec quelques lignes de codes pour prendre en main l'architecture présentée chapitre 3. Cela inclut :

- création du modèle,
- chargement d'un modèle entraîné,
- prétraitement d'un tenseur d'entrée,
- inférence du modèle avec le tenseur d'entrée,
- évaluation de la prédiction.

1. Création du modèle

Les codes sources se trouvent dans le dossier *src* du projet.

```
import os

# Import FUSION model and prime_X_fusion function
from src.models.pose_ir_fusion import *
# Import "device" and "classes" variables
from src.models.utils import *
```

1.1 Variables globales

Sont renseignés, la localisation du dossier contenant le réseau entraîné, le nom du fichier, les modules compris par le réseau.

```
# Global variables
model_folder = os.getcwd() +'/../'\
                           +'models/'\
                           +'fusion_test_tube_seed=0/'\
                           +'fusion_20/'\
                           +'cross_subject/'\
                           +'aug=True/'
model_file ='model12.pt'
use_pose = True
use_ir = True
```

1.2 Création du modèle

Appel de la classe *FUSION* avec des poids aléatoires.

```
model = FUSION(use_pose, use_ir, pretrained = False)
```

1.3 Chargement des poids

Chargement des poids entraînés.

```
model.load_state_dict(torch.load(model_folder + model_file))
```

1.4 Chargement sur la carte d'entraînement et mise en mode évaluation

Si une carte graphique est détectée, elle sera automatiquement utilisée.

```
model.to(device)
model.to(device)
```

2. Inférence (propagation avant)

La variable *batch_size* indique le nombre de séquences à traiter en parallèle. La variable *seq_len* correspond à la longueur de la séquence normalisée.

```
# Global variables
batch_size = 1
seq_len = 20
```

2.1 Création d'un tenseur aléatoire

Création de tenseurs aléatoires simulant les données du squelette 3D et d'une séquence infrarouge normalisée en temps.

```
X_skeleton = torch.rand(batch_size, 3, 224, 224)
X_ir = torch.rand(batch_size, seq_len, 3, 112, 112)
X = [X_skeleton, X_ir]
```

2.2 Préparation du tenseur en entrée

Les tenseurs en entrée sont ensuite normalisés.

```
X_primed = prime_X_fusion(X, use_pose, use_ir)
```

2.3 Propagation avant (inférence)

Calcule d'une prédiction à l'aide du réseau entraîné pour les tenseurs en entrée.

```
predictions = model(X_primed)
_, class_predicted = predictions.max(1)
print("Class predicted : " + classes[class_predicted.item()])
```

RÉFÉRENCES

Aggarwal, J. K. & Xia, L. (2014). Human activity recognition from 3d data : A review. *Pattern Recognition Letters*, 48, 70–80.

Ainsworth, T. (2002). *Buyer beware, Security Oz*.

Aleman, A. & Denys, D. (2014). A road map for suicide research and prevention. *Nature*, 509(7501), 421–423.

Baradel, F., Wolf, C. & Mille, J. (2017). Pose-conditioned spatio-temporal attention for human action recognition. *arXiv preprint arXiv :1703.10106*.

Baradel, F., Wolf, C., Mille, J. & Taylor, G. W. (2018). Glimpse clouds : Human activity recognition from unstructured feature points. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 469–478.

Barros, J., Morales, S., Echávarri, O., García, A., Ortega, J., Asahi, T., Moya, C., Fischman, R., Maino, M. P. & Núñez, C. (2017). Suicide detection in Chile : proposing a predictive model for suicide risk in a clinical sample of patients with mood disorders. *Revista Brasileira de Psiquiatria*, 39(1), 1–11.

Brunel, C., Fermanian, C., Durigon, M. & de la Grandmaison, G. L. (2010). Homicidal and suicidal sharp force fatalities : autopsy parameters in relation to the manner of death. *Forensic science international*, 198(1-3), 150–154.

Buch, S., Escorcia, V., Ghanem, B., Fei-Fei, L. & Niebles, J. C. (2017a). End-to-End, Single-Stream Temporal Action Detection in Untrimmed Videos. *Proceedings of the British Machine Vision Conference (BMVC)*.

Buch, S., Escorcia, V., Shen, C., Ghanem, B. & Carlos Niebles, J. (2017b). Sst : Single-stream temporal action proposals. *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2911–2920.

Buciluǎ, C., Caruana, R. & Niculescu-Mizil, A. (2006). Model compression. *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 535–541.

Caba Heilbron, F., Carlos Niebles, J. & Ghanem, B. (2016). Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1914–1923.

Carreira, J. & Zisserman, A. (2017). Quo vadis, action recognition ? a new model and the kinetics dataset. *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299–6308.

Chao, Y.-W., Vijayanarasimhan, S., Seybold, B., Ross, D. A., Deng, J. & Sukthankar, R. (2018). Rethinking the faster r-cnn architecture for temporal action localization. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1130–1139.

Cho, K., Courville, A. & Bengio, Y. (2015). Describing multimedia content using attention-based encoder-decoder networks. *IEEE Transactions on Multimedia*, 17(11), 1875–1886.

Cramer, R. J., Wechsler, H. J., Miller, S. L. & Yenne, E. (2017). Suicide prevention in correctional settings : current standards and recommendations for research, prevention, and training. *Journal of correctional health care*, 23(3), 313–328.

Crasto, N., Weinzaepfel, P., Alahari, K. & Schmid, C. (2019). MARS : Motion-augmented RGB stream for action recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7882–7891.

CSC. (2014). *A Three Year Review of Federal Inmate Suicides (2011-2014)*.

Dai, X., Singh, B., Zhang, G., Davis, L. S. & Qiu Chen, Y. (2017). Temporal context network for activity localization in videos. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5793–5802.

Dave, A., Russakovsky, O. & Ramanan, D. (2017). Predictive-corrective networks for action detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 981–990.

De Geest, R. & Tuytelaars, T. (2018). Modeling temporal structure with LSTM for online action detection. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1549–1557.

De Geest, R., Gavves, E., Ghodrati, A., Li, Z., Snoek, C. & Tuytelaars, T. (2016). Online action detection. *European Conference on Computer Vision*, pp. 269–284.

Delgado-Gomez, D., Blasco-Fontecilla, H., Sukno, F., Ramos-Plasencia, M. S. & Baca-Garcia, E. (2012). Suicide attempters classification : Toward predictive models of suicidal behavior. *Neurocomputing*, 92, 3–8.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. & Fei-Fei, L. (2009). Imagenet : A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255.

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

Du, Y., Fu, Y. & Wang, L. (2015a). Skeleton based action recognition with convolutional neural network. *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pp. 579–583.

Du, Y., Wang, W. & Wang, L. (2015b). Hierarchical recurrent neural network for skeleton based action recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1110–1118.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2), 179–211.

Ersen, B., Kahveci, R., Saki, M., Tunali, O. & Aksu, I. (2017). Analysis of 41 suicide attempts by wrist cutting : a retrospective analysis. *European journal of trauma and emergency surgery*, 43(1), 129–135.

Escorcia, V., Heilbron, F. C., Niebles, J. C. & Ghanem, B. (2016). Daps : Deep action proposals for action understanding. *European Conference on Computer Vision*, pp. 768–784.

Fazel, S., Grann, M., Kling, B. & Hawton, K. (2011). Prison suicide in 12 countries : an ecological study of 861 suicides during 2003–2007. *Social psychiatry and psychiatric epidemiology*, 46(3), 191–195.

Fazel, S., Ramesh, T. & Hawton, K. (2017). Suicide in prisons : an international study of prevalence and contributory factors. *The Lancet Psychiatry*, 4(12), 946–952.

Feichtenhofer, C., Pinz, A. & Zisserman, A. (2016). Convolutional two-stream network fusion for video action recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1933–1941.

Folk, M., Heber, G., Koziol, Q., Pourmal, E. & Robinson, D. (2011). An overview of the HDF5 technology suite and its applications. *Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases*, pp. 36–47.

Fujioka, M., Murakami, C., Masuda, K. & Doi, H. (2012). Evaluation of superficial and deep self-inflicted wrist and forearm lacerations. *The Journal of hand surgery*, 37(5), 1054–1058.

Gagnon, A. (2016). Field trial results using a novel integration of unique millimeterwave Doppler radar for high performance non-obtrusive life sign (breathing and heart beating) monitoring of high suicide risk prisonner in observation cell. *Security Technology (ICCST), 2016 IEEE International Carnahan Conference on*, pp. 1–9.

Gao, J., Yang, Z., Chen, K., Sun, C. & Nevatia, R. (2017a). Turn tap : Temporal unit regression network for temporal action proposals. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3628–3636.

Gao, J., Yang, Z. & Nevatia, R. (2017b). Cascaded boundary regression for temporal action detection. *arXiv preprint arXiv :1705.01180*.

Gao, J., Yang, Z. & Nevatia, R. (2017c). Red : Reinforced encoder-decoder networks for action anticipation. *arXiv preprint arXiv :1707.04818*.

Girshick, R. (2015). Fast r-cnn. *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448.

Girshick, R., Donahue, J., Darrell, T. & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587.

Guenat,Claire ; Bérard, J. (2018). Prévenir le suicide en prison au Canada : Les premiers pas d'une politique publique (1970-1987). *Criminologie*, 51(2), 61–85. doi : https://doi.org/10.7202/1054235ar.

Han, F., Reily, B., Hoff, W. & Zhang, H. (2017). Space-time representation of people based on 3D skeletal data : A review. *Computer Vision and Image Understanding*, 158, 85–105.

Hayes, L. M. (2013). Suicide prevention in correctional facilities : Reflections and next steps. *International journal of law and psychiatry*, 36(3-4), 188–194.

He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Heilbron, F. C., Barrios, W., Escorcia, V. & Ghanem, B. (2017). Scc : Semantic context cascade for efficient action detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3175–3184.

Hinton, G., Vinyals, O. & Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv :1503.02531*.

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

Hou, Y., Li, Z., Wang, P. & Li, W. (2016). Skeleton optical spectra-based action recognition using convolutional neural networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(3), 807–811.

Hou, Y., Wang, S., Wang, P., Gao, Z. & Li, W. (2017). Spatially and temporally structured global to local aggregation of dynamic depth information for action recognition. *IEEE Access*, 6, 2206–2219.

Hu, J.-F., Zheng, W.-S., Ma, L., Wang, G., Lai, J. & Zhang, J. (2018a). Early action prediction by soft regression. *IEEE transactions on pattern analysis and machine intelligence*, 41(11), 2568–2583.

Hu, J.-F., Zheng, W.-S., Pan, J., Lai, J. & Zhang, J. (2018b). Deep bilinear learning for rgb-d action recognition. *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 335–351.

Hussein, M. E., Torki, M., Gowayyed, M. A. & El-Saban, M. (2013). Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. *Twenty-Third International Joint Conference on Artificial Intelligence*.

Ioffe, S. & Szegedy, C. (2015). Batch normalization : Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv :1502.03167*.

Johansson, G. (1973). Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14(2), 201–211.

Karger, B., Niemeyer, J. & Brinkmann, B. (2000). Suicides by sharp force : typical and atypical features. *International journal of legal medicine*, 113(5), 259–262.

Karlsson, T. (1998). Homicidal and suicidal sharp force fatalities in Stockholm, Sweden. : Orientation of entrance wounds in stabs gives information in the classification. *Forensic science international*, 93(1), 21–32.

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

Kaster, T. S., Martin, M. S. & Simpson, A. (2017). Preventing Prison Suicide with Life Trajectory-Based Screening. *The journal of the American Academy of Psychiatry and the Law*, 45, 92–98.

Ke, Q., An, S., Bennamoun, M., Sohel, F. & Boussaid, F. (2017a). Skeletonnet : Mining deep part features for 3-d action recognition. *IEEE signal processing letters*, 24(6), 731–735.

Ke, Q., Bennamoun, M., An, S., Sohel, F. & Boussaid, F. (2017b). A new representation of skeleton sequences for 3d action recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3288–3297.

Ke, Q., Liu, J., Bennamoun, M., Rahmani, H., An, S., Sohel, F. & Boussaid, F. (2018). Global regularizer and temporal-aware cross-entropy for skeleton-based early action recognition. *Asian Conference on Computer Vision*, pp. 729–745.

Ke, Q., Bennamoun, M., Rahmani, H., An, S., Sohel, F. & Boussaid, F. (2019). Learning latent global network for skeleton-based action prediction. *IEEE Transactions on Image Processing*, 29, 959–970.

Keselman, L., Iselin Woodfill, J., Grunnet-Jepsen, A. & Bhowmik, A. (2017). Intel realsense stereoscopic depth cameras. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–10.

Kim, T. S. & Reiter, A. (2017). Interpretable 3d human action analysis with temporal convolutional networks. *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, pp. 1623–1631.

Kingma, D. P. & Ba, J. (2014). Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*.

Kong, Y., Kit, D. & Fu, Y. (2014). A discriminative model with multiple temporal scales for action prediction. *European conference on computer vision*, pp. 596–611.

Kong, Y., Tao, Z. & Fu, Y. (2017). Deep sequential context networks for action prediction. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1473–1481.

Kong, Y., Gao, S., Sun, B. & Fu, Y. (2018). Action prediction from videos via memorizing hard-to-predict samples. *Thirty-Second AAAI Conference on Artificial Intelligence*.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, pp. 1097–1105.

Krywanczyk, A. & Shapiro, S. (2015). A retrospective study of blade wound characteristics in suicide and homicide. *The American journal of forensic medicine and pathology*, 36(4), 305–310.

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

Lan, T., Chen, T.-C. & Savarese, S. (2014). A hierarchical representation for future action prediction. *European Conference on Computer Vision*, pp. 689–704.

LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

Lee, I., Kim, D., Kang, S. & Lee, S. (2017). Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1012–1020.

Li, B., Dai, Y., Cheng, X., Chen, H., Lin, Y. & He, M. (2017a). Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep CNN. *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 601–604.

Li, C., Hou, Y., Wang, P. & Li, W. (2017b). Joint distance maps based action recognition with convolutional neural networks. *IEEE Signal Processing Letters*, 24(5), 624–628.

Li, C., Hou, Y., Wang, P. & Li, W. (2018). Multiview-based 3-D action recognition using deep networks. *IEEE Transactions on Human-Machine Systems*, 49(1), 95–104.

Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y. & Tian, Q. (2019). Actional-Structural Graph Convolutional Networks for Skeleton-based Action Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3595–3603.

Li, Y., Wang, N., Liu, J. & Hou, X. (2017c). Demystifying neural style transfer. *arXiv preprint arXiv :1701.01036*.

Lin, T., Zhao, X. & Shou, Z. (2017). Single shot temporal action detection. *Proceedings of the 25th ACM international conference on Multimedia*, pp. 988–996.

Liu, C., Hu, Y., Li, Y., Song, S. & Liu, J. (2017a). PKU-MMD : A large scale benchmark for continuous multi-modal human action understanding. *arXiv preprint arXiv :1703.07475*.

Liu, J., Shahroudy, A., Xu, D. & Wang, G. (2016). Spatio-temporal lstm with trust gates for 3d human action recognition. *European Conference on Computer Vision*, pp. 816–833.

Liu, M., Liu, H. & Chen, C. (2017b). Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68, 346–362.

Luo, Z., Hsieh, J.-T., Jiang, L., Carlos Niebles, J. & Fei-Fei, L. (2018). Graph distillation for action detection with privileged modalities. *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 166–183.

Ma, S., Sigal, L. & Sclaroff, S. (2016). Learning activity progression in lstms for activity detection and early detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1942–1950.

Main de Boissiere, A. & Noumeir, R. (2020). Infrared and 3D skeleton feature fusion for RGB-D action recognition.

Marzano, L., Hawton, K., Rivlin, A., Smith, E. N., Piper, M. & Fazel, S. (2016). Prevention of suicidal behavior in prisons. *Crisis*.

Matsumoto, T., Yamaguchi, A., Chiba, Y., Asami, T., Iseki, E. & Hirayasu, Y. (2004). Patterns of self-cutting : A preliminary study on differences in clinical implications between wrist-and arm-cutting using a Japanese juvenile detention center sample. *Psychiatry and Clinical Neurosciences*, 58(4), 377–382.

McKee, G. R. (1998). Lethal vs nonlethal suicide attempts in jail. *Psychological Reports*, 82(2), 611–614.

Mnih, V., Heess, N., Graves, A. et al. (2014). Recurrent models of visual attention. *Advances in neural information processing systems*, pp. 2204–2212.

Ni, B., Yang, X. & Gao, S. (2016). Progressively parsing interactional objects for fine grained action detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1020–1028.

Nuitrack™. (2020). Nuitrack Full Body Skeletal Tracking Software. Repéré à https://nuitrack.com/.

Oh, J., Yun, K., Hwang, J.-H. & Chae, J.-H. (2017). Classification of suicide attempts through a machine learning algorithm based on multiple systemic psychiatric scales. *Frontiers in psychiatry*, 8, 192.

Olah, C. (2015, August, 27). Understanding LSTM Networks [blog]. Repéré à https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

Pang, G., Wang, X., Hu, J.-F., Zhang, Q. & Zheng, W.-S. (2019). DBDNet : learning bi-directional dynamics for early action prediction. *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 897–903.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L. & Lerer, A. (2017). Automatic differentiation in pytorch.

Player, M. J., Proudfoot, J., Fogarty, A., Whittle, E., Spurrier, M., Shand, F., Christensen, H., Hadzi-Pavlovic, D. & Wilhelm, K. (2015). What interrupts suicide attempts in men : a qualitative study. *PLoS One*, 10(6), e0128180.

Presti, L. L. & La Cascia, M. (2016). 3D skeleton-based human action classification : A survey. *Pattern Recognition*, 53, 130–147.

Pridmore, S., Money, T. T. & Pridmore, W. (2018). Suicide : What the General Public and the Individual Should Know. *Malaysian Journal of Medical Science*, 25(2), 15–19.

Racette, S., Kremer, C., Desjarlais, A. & Sauvageau, A. (2008). Suicidal and homicidal sharp force injury : a 5-year retrospective comparative study of hesitation marks and defense wounds. *Forensic science, medicine, and pathology*, 4(4), 221–227.

Rahmani, H. & Bennamoun, M. (2017). Learning action recognition model from depth and skeleton videos. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5832–5841.

Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. (2016). You only look once : Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788.

Ren, S., He, K., Girshick, R. & Sun, J. (2015). Faster r-cnn : Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, pp. 91–99.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C. & Bengio, Y. (2014). Fitnets : Hints for thin deep nets. *arXiv preprint arXiv :1412.6550*.

Ryoo, M. S. (2011). Human activity prediction : Early recognition of ongoing activities from streaming videos. *2011 International Conference on Computer Vision*, pp. 1036–1043.

Sadegh Aliakbarian, M., Sadat Saleh, F., Salzmann, M., Fernando, B., Petersson, L. & Andersson, L. (2017). Encouraging lstms to anticipate actions very early. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 280–289.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. & Batra, D. (2017). Grad-cam : Visual explanations from deep networks via gradient-based localization. *Proceedings of the IEEE international conference on computer vision*, pp. 618–626.

Shahroudy, A., Liu, J., Ng, T.-T. & Wang, G. (2016). Ntu rgb+ d : A large scale dataset for 3d human activity analysis. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1010–1019.

Shahroudy, A., Ng, T.-T., Gong, Y. & Wang, G. (2017). Deep multimodal feature analysis for action recognition in rgb+ d videos. *IEEE transactions on pattern analysis and machine intelligence*, 40(5), 1045–1058.

Sharma, S., Kiros, R. & Salakhutdinov, R. (2015). Action recognition using visual attention. *arXiv preprint arXiv :1511.04119*.

Shi, L., Zhang, Y., Cheng, J. & Lu, H. (2019a). Skeleton-Based Action Recognition with Directed Graph Neural Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7912–7921.

Shi, L., Zhang, Y., Cheng, J. & Lu, H. (2019b). Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12026–12035.

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A. & Blake, A. (2011). Real-time human pose recognition in parts from single depth images. *CVPR 2011*, pp. 1297–1304.

Shou, Z., Wang, D. & Chang, S.-F. (2016). Temporal action localization in untrimmed videos via multi-stage cnns. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1049–1058.

Shou, Z., Pan, J., Chan, J., Miyazawa, K., Mansour, H., Vetro, A., Nieto, X. G. & Chang, S.-F. (2018). Online action detection in untrimmed, streaming videos-modeling and evaluation. *ECCV*, 1, 5.

Si, C., Chen, W., Wang, W., Wang, L. & Tan, T. (2019). An Attention Enhanced Graph Convolutional LSTM Network for Skeleton-Based Action Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1227–1236.

Simonovsky, M. & Komodakis, N. (2017). Dynamic edge-conditioned filters in convolutional neural networks on graphs. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3693–3702.

Simonyan, K. & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, pp. 568–576.

Simonyan, K., Vedaldi, A. & Zisserman, A. (2013). Deep inside convolutional networks : Visualising image classification models and saliency maps. *arXiv preprint arXiv :1312.6034*.

Song, S., Lan, C., Xing, J., Zeng, W. & Liu, J. (2017). An end-to-end spatio-temporal attention model for human action recognition from skeleton data. *Thirty-first AAAI conference on artificial intelligence*.

Song, S., Lan, C., Xing, J., Zeng, W. & Liu, J. (2018). Spatio-temporal attention-based lstm networks for 3d action recognition and detection. *IEEE Transactions on Image Processing*, 27(7), 3459–3471.

Soomro, K., Zamir, A. R. & Shah, M. (2012). UCF101 : A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv :1212.0402*.

Suto, I. & Arnaut, G. L. (2010). Suicide in prison : A qualitative study. *The Prison Journal*, 90(3), 288–312.

Tran, D., Bourdev, L., Fergus, R., Torresani, L. & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. *Proceedings of the IEEE international conference on computer vision*, pp. 4489–4497.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y. & Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 6450–6459.

Tu, Z., Xie, W., Qin, Q., Poppe, R., Veltkamp, R. C., Li, B. & Yuan, J. (2018). Multi-stream CNN : Learning representations based on human-related regions for action recognition. *Pattern Recognition*, 79, 32–43.

Tu, Z., Li, H., Zhang, D., Dauwels, J., Li, B. & Yuan, J. (2019). Action-stage emphasized spatiotemporal vlad for video action recognition. *IEEE Transactions on Image Processing*, 28(6), 2799–2812.

Vassalini, M., Verzeletti, A. & De Ferrari, F. (2014). Sharp force injury fatalities : a retrospective study (1982–2012) in Brescia (Italy). *Journal of forensic sciences*, 59(6), 1568–1574.

Vemulapalli, R., Arrate, F. & Chellappa, R. (2014). Human action recognition by representing 3d skeletons as points in a lie group. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 588–595.

Walsh, C. G., Ribeiro, J. D. & Franklin, J. C. (2017). Predicting risk of suicide attempts over time through machine learning. *Clinical Psychological Science*, 5(3), 457–469.

Wang, H. & Schmid, C. (2013). Action recognition with improved trajectories. *Proceedings of the IEEE international conference on computer vision*, pp. 3551–3558.

Wang, H. & Wang, L. (2017). Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 499–508.

Wang, J., Liu, Z., Wu, Y. & Yuan, J. (2012). Mining actionlet ensemble for action recognition with depth cameras. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1290–1297.

Wang, L., Qiao, Y. & Tang, X. (2014). Action recognition and detection by combining motion and appearance features. *THUMOS14 Action Recognition Challenge*, 1(2), 2.

Wang, P., Li, Z., Hou, Y. & Li, W. (2016). Action recognition based on joint trajectory maps using convolutional neural networks. *Proceedings of the 24th ACM international conference on Multimedia*, pp. 102–106.

Wang, P., Li, W., Ogunbona, P., Wan, J. & Escalera, S. (2018a). RGB-D-based human motion recognition with deep learning : A survey. *Computer Vision and Image Understanding*, 171, 118–139.

Wang, P., Li, W., Wan, J., Ogunbona, P. & Liu, X. (2018b). Cooperative training of deep aggregation networks for RGB-D action recognition. *Thirty-Second AAAI Conference on Artificial Intelligence*.

Wang, X., Hu, J.-F., Lai, J.-H., Zhang, J. & Zheng, W.-S. (2019). Progressive teacher-student learning for early action prediction. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3556–3565.

WHO. (2019). *Suicide in the world : global health estimates*.

Xie, S., Sun, C., Huang, J., Tu, Z. & Murphy, K. (2018). Rethinking spatiotemporal feature learning : Speed-accuracy trade-offs in video classification. *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 305–321.

Xu, H., Das, A. & Saenko, K. (2017). R-c3d : Region convolutional 3d network for temporal activity detection. *Proceedings of the IEEE international conference on computer vision*, pp. 5783–5792.

Xu, K., Hu, W., Leskovec, J. & Jegelka, S. (2018). How powerful are graph neural networks ? *arXiv preprint arXiv :1810.00826*.

Xu, M., Gao, M., Chen, Y.-T., Davis, L. S. & Crandall, D. J. (2019). Temporal recurrent networks for online action detection. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5532–5541.

Yan, S., Xiong, Y. & Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. *Thirty-Second AAAI Conference on Artificial Intelligence*.

Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H. & Courville, A. (2015). Describing videos by exploiting temporal structure. *Proceedings of the IEEE international conference on computer vision*, pp. 4507–4515.

Yim, J., Joo, D., Bae, J. & Kim, J. (2017). A gift from knowledge distillation : Fast optimization, network minimization and transfer learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4133–4141.

Yuan, J., Ni, B., Yang, X. & Kassim, A. A. (2016). Temporal action localization with pyramid of score distribution features. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3093–3102.

Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R. & Toderici, G. (2015). Beyond short snippets : Deep networks for video classification. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4694–4702.

Zhang, D., Dai, X., Wang, X. & Wang, Y.-F. (2018). S3d : Single shot multi-span detector via fully 3d convolutional networks. *arXiv preprint arXiv :1807.08069*.

Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J. & Zheng, N. (2017). View adaptive recurrent neural networks for high performance human action recognition from skeleton data. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2117–2126.

Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J. & Zheng, N. (2019). View adaptive neural networks for high performance skeleton-based human action recognition. *IEEE transactions on pattern analysis and machine intelligence*.

Zhang, Z. (2012). Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2), 4–10.

Zolfaghari, M., Oliveira, G. L., Sedaghat, N. & Brox, T. (2017). Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2904–2913.