Fusion de modèles bayésiens et de convolution pour la reconnaissance d'actions

Camille Maurice[1] Francisco Madrigal[1] Frédéric Lerasle[1,2]

[1] LAAS-CNRS, Toulouse, France
[2] Université Paul Sabatier, Toulouse, France

{cmaurice, jfmadrig, lerasle}@laas.fr

Résumé

Les différentes actions qui ont lieu au cours d'une séquence vidéo suivent généralement un ordre logique. Dans cet article nous proposons une approche hybride qui résulte de la fusion d'un réseau de convolution avec une approche bayésienne qui repose sur des modèles d'interactions homme-objets et des transitions entre les différentes actions. L'idée est de combiner dans la prédiction finale ces deux approches. Nous validons notre stratégie de fusion sur deux jeux de données publics : CAD-120 [7] et Watch-n-Patch [27]. Par rapport aux deux méthodes individuelles la fusion permet un gain en justesse de +4% et +6% respectivement sur les deux jeux de données. Les performances de reconnaissance d'actions sont clairement améliorées grâce à la fusion, notamment lors de déséquilibre entre les classes.

Mots Clef

Analyse vidéo, modèle bayésien, réseau de convolution, reconnaissance d'actions.

Abstract

During a video sequence, actions usually follow a logical order. In this paper, we propose an hybrid approach by the mean of the fusion of a deep learning network with a Bayesian model based on the interactions between human and objects and transitions between actions. The key idea is to combine the two approaches in the final prediction. We validate our strategy on two public datasets : CAD-120 [7] and Watch-n-Patch [27]. We show a performance gain of respectively +4% ,+6% in accuracy compared to each baseline approach. Action recognition performances are clearly improved by the fusion, especially when classes are imbalanced.

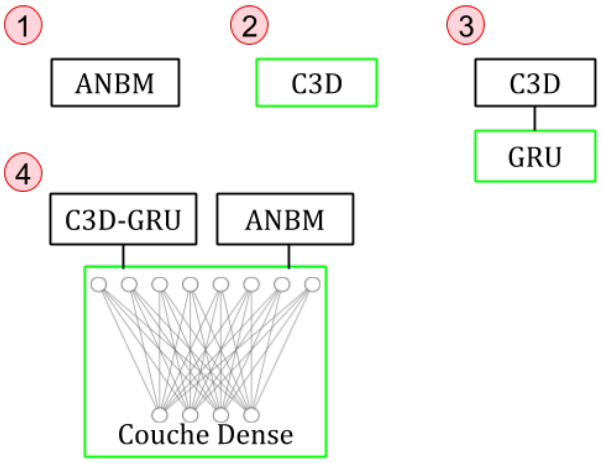
Keywords

Video analysis, Bayesian model, convolutional network, action recognition.

1 Introduction

La reconnaissance des activités humaines est à la source du développement de nombreuses applications pratiques telle que le suivi d'activité domestique ou pour la collaboration homme-robot. Une activité est définie par une suite temporelle d'actions [22, 14] e.g. : *préparer le café* implique les actions *verser l'eau*, *ajouter le café moulu* et *allumer la machine*. Les activités exécutées par les hommes dans un environnement domestique ou industriel peuvent être très différentes, par exemple dans la nature des objets utilisés. En revanche les actions atomiques mises en oeuvre peuvent être similaires quelque soit le contexte. En effet ces actions atomiques concernent le déplacement des objets, leur saisie, ou les interactions qu'ils peuvent avoir avec leur environnement. Nous nous intéressons donc à la reconnaissance d'actions atomiques et à leur séquencement. Car les activités peuvent être représentées par des actions agencées en séquences suivant un certain ordre logique.

FIGURE 1 – Les différentes approches individuelles comparées (1) (2) (3) ainsi que leur fusion (4). Les couches apprises lors de l'entraînement sont encadrées en vert.



Les approches de type data-driven via les réseaux de neurones à convolution (CNN) et leur adaptation au domaine vidéo avec les convolutions 3D tel C3D [23] ont permis la reconnaissance d'actions sur des flux. Les réseaux de neurones à convolution 3D cherchent à apprendre des caractéristiques spatio-temporelles simultanément. Un tel

apprentissage permet aux approches C3D [23] d'obtenir une justesse de 90.4% sur UCF101 [21]. En contrepartie les convolutions 3D augmentent la taille du réseau et donc le nombre de paramètres à apprendre (17M). Ils nécessitent donc de nombreuses données annotées, d'où l'émergence de plus grands jeux de données annotés tel que NTU RGB+D [20], UCF101 [21], Kinetic [2]. Par exemple UCF101 contient 27 heures de vidéos et NTU RGB+D [20] 56880 clips. Malgré cela, la reconnaissance d'actions reste toujours un défi car les réseaux de convolution 3D n'agrègent des caractéristiques temporelles que sur des clips vidéos i.e des actions pré-segmentées. De plus ils sont pris séparément et ne prennent pas en compte la séquence logique d'actions qui se déroule. Souvent ces grands jeux de donnés comme Kinetic [2] et UCF101 [21] sont crées à partir de vidéos collectées sur YouTube. Les différentes classes sont exécutées dans des environnements radicalement différents, par exemple *nager* vs. *jouer de la guitare*. Or, par exemple, dans le contexte du suivi d'activités domestiques, les actions à détecter ont lieu dans un environnement semblable, et ont une cohérence temporelle dans leur enchaînement représentant une certaine activité. La reconnaissance d'actions en séquence avec une faible variance inter-classes, des classes déséquilibrées, et/ou sous-représentées constituent toujours un défi pour les réseaux à convolution usuels.

Historiquement des approches probabilistes [8] [7] proposent de caractériser les actions de manière plus explicite à travers une modélisation des observations des éléments de la scène. Si ces approches basées sur des modèles probabilistes offrent des performances généralement moindre que les réseaux de convolution elles requièrent généralement une quantité de données moins importante car elles ont aussi moins de paramètres sous-jacents à estimer. Leur interprétabilité est donc moins dépendante des données d'apprentissage (e.g. moins sujet au sur-apprentissage). Ces approches sont pertinentes dans le cas d'un petit nombre d'échantillons disponibles pour un entraînement. Par exemple notre approche bayésienne ANBM (pour A New Bayesian Model [13]), modélise à la fois les interactions entre objets, homme-objets à travers environ 50 paramètres. Notons que notre approche ANBM permet en plus de prendre en compte les transitions entre différentes actions afin d'assurer une cohérence temporelle tout au long de la séquence d'actions.

Forts de ces constats nous proposons une approche qualifiée de IA hybride avec une fusion, au niveau décisionnel, d'un réseau à convolution C3D [23] et de notre approche probabiliste ANBM [13] basée sur des percepts homme-objets explicites. Ces deux approches prennent en compte les caractéristiques spatio-temporelles des différentes classes d'actions. En raison du grand nombre de paramètres le réseau C3D a besoin de beaucoup données annotées pour être pertinent, l'apprentissage est difficile sur les classes sous-représentées. L'approche ANBM dépend des modèles qui ont été élaborés et même avec peu de données la prédiction des classes sous représentées est possible.

Ainsi, nos contributions sont : (1) l'ajout d'une première contribution mineure qui est l'ajout d'une couche récurrente GRU (Gated Recurrent Unit) à C3D pour la reconnaissance d'actions afin de modéliser les corrélations temporelles entre actions, (2) la comparaison de deux approches ANBM et C3D-GRU sur deux datasets publics CAD120 et Watch-n-Patch, (3) mise en œuvre et évaluation d'une fusion tardive de ces deux approches (donc fusionnant leurs prédictions) et comparaison avec la littérature sur ces deux datasets afin d'observer les gains grâce à cette approche hybride.

L'article s'organise comme suit. En section 2 nous présentons l'état de l'art et le contexte dans lequel s'insère nos travaux. Puis dans la section 3 nous présentons notre approche hybride pour la détection d'actions. Une étude comparative de nos résultats est présentée dans la section 4. Pour finir, la section 5 présente notre conclusion ainsi que les perspectives à venir.

2 État de l'art

La reconnaissance d'actions statiques peut s'effectuer via la localisation sur une image de certains objets à l'instar de Zhou *et al* [30] ou Oquab *et al* [15]. Cette approche a été popularisée par le défi Pascal VOC 2012 reconnaissance d'actions sur des images [3]. Si cela est pertinent lorsque les classes d'actions à reconnaître interviennent dans des environnements différents, ces approches sont inappropriées pour reconnaître les actions atomiques qui se succèdent dans la même scène. Ce sont les mouvements mis en jeux lors de l'exécution d'une action qui permettent de les discriminer, par exemple lors de l'ouverture ou de la fermeture d'une porte. C'est pourquoi nous nous focalisons sur les approches utilisant des informations spatio-temporelles à partir de vidéos afin de considérer la dynamique des gestes et des objets lors des actions.

Les approches historiques abordent la détection d'actions via une modélisation probabiliste des percepts mis en jeux. En outre ces approches basées sur des modèles peuvent inclure des modèles de trajectoires pour la posture humaine, des informations sur la configuration spatiale des objets dans la scène ou leur affordance. Li *et al* [8] propose l'emploi de mixture de Gaussiennes afin de reconnaître différentes actions sur le jeu de données MSRAction [8]. Koppula et Saxena [7] proposent l'emploi de champs aléatoires conditionnels (CRF) pour modéliser la scène et les relations spatio-temporelles qui apparaissent dans CAD-120 [7]. Plus récemment nous avons proposé une approche bayésienne ANBM [13] qui repose sur la modélisation explicite en 3D des caractéristiques contextuelles, tant spatiales que temporelles. Ces approches reposent sur un nombre de paramètres plus restreint que celles des réseaux C3D. De fait elles requièrent moins de données et sont évaluées sur des jeux de données, généralement plus petits. Par exemple MSRAction [8] contient

420 séquences et CAD-120 [7] contient 120 vidéos pour environ 1000 clips après segmentation des actions. Elles présentent également l'avantage d'être plus interprétables que les approches CNN.

La dépendance au volume de données disponibles pour l'apprentissage est l'un des challenge que l'on retrouve avec les réseaux de convolutions dont l'apprentissage de leurs nombreux paramètres reposent sur la quantité de données disponibles pour l'entraînement. L'introduction des filtres de convolutions 3D [6] permet d'extraire des descripteurs spatio-temporels à partir d'un ensemble de trames, appelé clip. Ces descripteurs sont appropriés pour capturer implicitement le contexte lié au contenu des vidéos. Cette idée a été reprise par C3D [23] et d'autres variantes [25, 24, 11] pour la détection d'actions. L'ajout de clips vidéos en entrée du réseau requiert d'augmenter sa taille par rapport à un réseau de convolution 2D. Les réseaux C3D extraient un descripteur global à partir du clip indépendemment de l'action qui a eu lieu précédemment. Si cela est particulièrement adapté aux grands jeux de données avec de nombreux petits clips tel que UCF101 [21] avec ses 13000 clips dont la durée moyenne est de 7 secondes. Ces réseaux n'agrègent des caractéristiques temporelles que sur une fenêtre de taille fixe, typiquement de 16 trames. Ce n'est pas adapté à la reconnaissance d'actions qui ont une cohérence temporelle dans leur enchaînements.

D'où l'intérêt d'ajouter une couche récurrente à un réseau convolutionel 3D. Wang *et al.* [26] propose d'ajouter une couche Long Short Term Memory layer (LSTM) à un tel réseau. Aussi dans [12, 29] les auteurs proposent d'ajouter soit une couche LSTM ou une couche GRU pour renforcer la cohérence temporelle au sein du clip et ils s'évaluent sur UCF101 par exemple. D'une manière différente nous proposons d'ajouter une cohérence logique dans le séquencement des différentes actions.

La fusion de réseaux C3D avec d'autres modalités a déjà permis d'améliorer les performances dans divers challenges de la communauté Vision. Par exemple la fusion spatio-temporelle [5] consiste à fusionner une image avec une séquence de flux optique qui décrit le mouvement. Cela améliore les performances par rapport à un réseau C3D seul qui chercher à extraire simultanément les caractéristiques temporelles et spatiales au niveau des couches de convolution 3D. Il existe également des méthodes qui proposent une fusion de différentes caractéristiques de nature différentes comme par exemple l'audio et la vidéo [4]. Ces différentes approches montrent l'intérêt d'utiliser un mécanisme de fusion afin d'augmenter les performances générales. Cependant ce gain s'effectue au détriment de la quantité de données nécessaires à l'entraînement. L'ajout de modalités augmentent encore le nombre de paramètres à apprendre pour le réseau à convolution. Ce qui a deux effets le premier la nécessité d'un tel jeu de données, et le deuxième l'allongement des temps d'apprentissage.

Nous proposons de fusionner deux approches spatio-temporelles l'une reposant sur une modélisation du contexte via l'apprentissage telle que C3D [23] et de notre approche ANBM [13] basée sur des modèles bayésien et des percepts homme/objet en 3D de la scène observée. Cette fusion ne se fait pas au niveau des caractéristiques mais tardivement au niveau de leurs prédictions vers une même couche. On propose de les fusionner en utilisant une couche entièrement connectée, dite couche dense. Peu de travaux étudient la fusion tardive de deux classes d'approches à priori complémentaires et des gains que cela peut apporter.

Des jeux de données publics tels que Watch-n-Patch [27] et CAD-120 [7] permettent d'évaluer la reconnaissance d'actions atomiques. Ces jeux de données proposent des vidéos longues d'environ 20 secondes, dans lesquelles différentes actions atomiques sont annotées. Les actions se succèdent dans un ordre logique, par exemple on ne peut pas déplacer un objet qui n'a pas été saisi précédemment. Dans ces jeux de données les séquences d'actions sont plus ou moins corrélées. De plus certaines classes y sont sous-représentées ce qui est généralement un verrou pour l'apprentissage de C3D.

3 Notre approche hybride

Ici nous décrivons l'architecture proposée pour la fusion des probabilités de sortie d'une approche bayésienne ANBM [13] avec celles d'un réseau à convolution C3D [23] modifié.

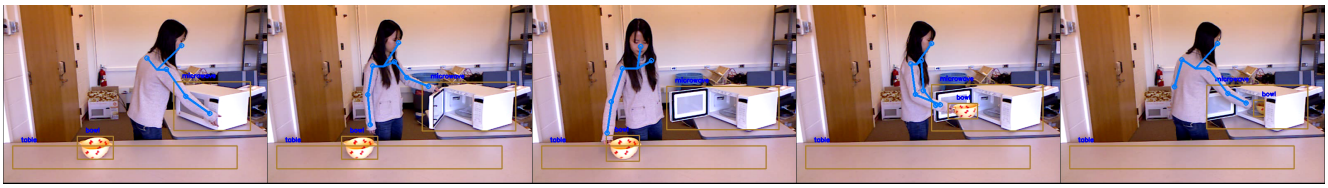
Nous rappelons notre première approche dans la Section 3.1, puis nous décrivons brièvement le réseau C3D dans la Section 3.2 et sa modification dans la section 3.3. La Section 3.4 détaille la stratégie de fusion tardive proposée.

3.1 Approche bayésienne mixant des percepts homme objets [13]

Cette approche [13] repose sur l'observation suivante : les postures, les interactions homme-objets et entre objets lors de la réalisation d'une action fournissent des informations spatiales caractéristiques pour permettre de reconnaître l'action en cours d'exécution. De même que les informations temporelles telles que les transitions entre actions au cours de la séquence. Nous avons modélisé ces observations afin de pouvoir donner, à chaque instant de la vidéo, les probabilités de chacune des actions considérées. L'ensemble des éléments de la scène sont d'abord localisés dans l'image par des détecteurs 2D de l'état de l'art puis modélisés dans l'espace 3D en utilisant des données de calibration de la Kinect. La détection des postures dans l'image s'appuie sur OpenPose [1] entraîné sur MSCOCO keypoint challenge [9]. Nous utilisons Single Shot MultiBox Detector (SSD) [10] pour reconnaître les objets, il est entraîné à partir du jeu de données MSCOCO [9].

Notre modélisation s'écrit alors : On associe à chaque action a un modèle. Soit $\mathrm{A} = \{a^1, a^2, ..., a^n\}$ l'ensemble des n actions. L'observation conjointe de la posture s_t et de l'ensemble d'objets $Omega_t$ est décrite à l'instant t par

FIGURE 2 – Une séquence d'actions extraite de CAD-120 [7] : Acteur 1, référence 2305260828, action réchauffer au micro-ondes (*microwaving-food*). De gauche à droite : atteindre, ouvrir, atteindre, déplacer, placer. En bleu : le squelette détecté par OpenPose. En jaune : les objets détectés par SSD.



$O_t = \{s_t, \Omega_t\}$ avec $\Omega_t = \{\omega^1, \omega^2, ..., \omega^{Card(\Omega)}\}$ avec $Card(\Omega)$ le nombre d'objets dans la scène. L'inférence se fait sur une fenêtre glissante de taille fixe de T trames. Nous modélisons la probabilité *a posteriori* des actions sachant les observations comme suit :

$$p(a_{0:T}|O_{0:T}) \propto \prod_{t=0}^{T} p(O_t|a_t) \prod_{t=1}^{T} p(a_t|a_{t-1}). \quad (1)$$

Où $p(O_t|a_t)$ est la vraisemblance de l'observation sachant l'action a_t. Le terme $p(a_t|a_{t-1})$ caractérise les probabilités de transitions entre deux actions successives. Elle modélise la scène en 3D ce qui lui permet également d'être plus robuste aux changements de point de vue qu'une approche qui se base uniquement sur des caractéristiques spatiales 2D. Le lecteur peut consulter [13] pour plus de détails.

3.2 Réseau Convolutif en 3-Dimensions (C3D)

C3D [23] est un réseau d'apprentissage profond qui prend en compte en plus des images, une troisième dimensions correspondant au temps. L'architecture comprend des filtres convolutifs de taille 3 x 3 x 3, suivis par des couches de pooling de taille 2 x 2 x 2. L'introduction de filtres 3D convolutifs permettent d'apprendre des descripteurs spatio-temporaux à partir d'un flux vidéo.

Ces réseaux apprennent de manière implicite des descripteurs liés au mouvement. Ils proposent une description compacte (4096) d'un flux vidéo de taille H x W x C x L. Avec L la longueur de l'action, généralement 16 trames. Comme les autres réseaux ils permettent un apprentissage de bout en bout sans informations d'experts (contrairement à tout approche probabiliste e.g. ANBM).

Par contre au vu des millions de paramètres à apprendre, les classes faiblement représentées sont plus difficiles à prédire correctement. L'action doit être échantillonnée sur 16 trames dans l'implémentation originale, bien sûr il est possible d'agrandir cette fenêtre temporelle mais cela requiert d'autant plus de mémoire. Tran *et al* [23] propose également un système de fenêtre glissante de moyenne des descripteurs. Cependant dans tous les cas C3D présuppose un pré-découpage des actions dans les séquences, ce qui n'en fait pas une méthode adaptée pour de la reconnaissance d'action en ligne. De plus il n'y a pas de mécanismes pour prendre en compte lors de l'apprentissage le contexte temporel dans lequel le clip vidéo s'insère. Il n'y a donc pas de prise en compte de l'action précédente.

3.3 C3D-GRU

Afin de pallier l'absence de mécanisme de prise en compte de l'action précédente, on propose qu'une fois l'entraînement de C3D est fait, de récupérer ses poids, de les geler et d'y ajouter une couche récurrente de type GRU. Ensuite l'entraînement du réseau est adapté pour prendre en compte deux clips successifs correspondants à deux actions. Nous ne ré-entraînons pas l'ensemble du réseau C3D, nous faisons seulement un paramétrage fin (*fine-tuning*) au niveau de GRU et des dernières couches. Pour illustrer l'importance du séquencement des actions, parmi l'ensemble des transitions envisageables entre deux paires d'actions dans Watch-n-Patch [7], seules environ 20 % sont possibles. Cette stratégie est illustrée par la Fig. 1, assignée du numéro 3. Nous appelons cette approche C3D-GRU par la suite.

3.4 Fusion tardive avec couche dense

Nous avons donc deux approches pour prédire les actions à partir de clips vidéos à partir des données spatio-temporelles, de manière explicite avec ANBM et de manière implicite avec C3D. De plus ANBM considère également les transitions qui existent entre deux actions successives. D'un coté dans ANBM nous avons modélisé chacune des actions, de l'autre C3D-GRU va apprendre à partir des données des jeux de données, dont les classes ne sont pas réparties équitablement. En effet dans la détection d'actions atomiques, certaines actions se retrouvent plus fréquemment par exemple le déplacement d'un objet (*déplacer*) représente 34% des actions de CAD-120. Nous proposons ici la fusion de leurs prédictions respectives. En effet les deux approches prédisent un vecteur de probabilités de chacune des classes pour ANBM, et pour C3D-GRU nous avons un vecteur qui correspond à la sortie de la couche soft-max.

Nous proposons une stratégie qui prend en entrée les clips vidéos qui passent à travers l'approche ANBM et également à travers le réseau C3D-GRU décrit précédemment. L'ensemble des poids du réseau C3D-GRU sont maintenant gelés. Nous obtenons donc deux vecteurs de prédictions pour chacune des approches que nous concaténons. Cette couche de concaténation est connectée à une couche dense de même taille que le nombre de classes N. Il n'y a donc

que $N^2 + N$ paramètres à apprendre (N^2 poids liés à la couche dense et N biais liés à l'activation). L'approche numéro 4 de la Fig. 1 en est une illustration, la couche dense est illustrée avec $N = 4$. Cette interconnexion permet de tirer parti des deux approches dans la détection finale. Nous appelons cette approche C3D-GRU-FD par la suite.

4 Expérimentations et résultats

4.1 Jeux de données

Nous avons proposé une première approche de détection d'actions dans [13]. Cette approche en-ligne est capable de détecter des séquences d'actions à partir d'un flux et de gérer les transitions entre actions. On souhaite tirer bénéfice de cet atout, nous avons donc choisi de nous évaluer sur deux jeux de données publics qui contiennent de telles séquences d'actions : CAD-120 [7] et Watch-n-Patch [27].

CAD-120 Le jeu de données CAD-120 [7] est constitué de 120 vidéos avec canal couleur et profondeur, jouées par quatre acteurs. Il contient 10 activités de la vie quotidienne (préparer un bol de céréales, prendre des médicaments...). Ces activités font intervenir 10 actions : atteindre, déplacer, verser, manger, boire, placer, ouvrir, fermer, nettoyer et aucune action. Ici, chaque vidéo représente une activité comme définie dans la section 1. L'ensemble des actions du dataset et leur distribution inéquitable, exprimée par le pourcentage de trames correspondant, sont décrites dans Tab. 3. Un extrait d'une séquence de ce jeu de données est présentée dans la Fig. 2.

Watch-n-Patch L'environnement "bureau" (office) est constitué de 196 vidéos enregistrées dans 8 bureaux différents. Il y a 10 actions annotées : lire, marcher, quitter le bureau, attraper un livre, remettre un livre, poser un objet, prendre un objet, jouer à l'ordinateur, allumer l'écran éteindre l'écran. Ici encore certaines actions sont dépendantes de celles qui ont lieu précédemment, e.g : pour jouer à l'ordinateur l'écran doit être allumé. Des classes d'actions ne sont pas équitablement réparties comme décrites dans le Tab. 3.

TABLE 3 – Détail de la répartition des classes et nombres de clips (Nb clips) dans les jeux de données.

Jeux de données	Nb clips	Répartition (%)
CAD-120	1149	[23,30,3,3,3,15,4,3,1,14]
Watch-n-Patch	1148	[12,16,21,6,4,14,9,9,5,3]

4.2 Évaluations du système

Gestion des prédictions de ANBM Nous avons enregistré les probabilités en sortie de ANBM, puis nous prenons leurs moyennes sur la durée de chacune des actions afin d'assigner une classe à un clip vidéo représentant une action.

Pré-traitements pour C3D Nous avons conservé les paramètres originaux [23] pour la taille des images en entrée en la fixant à 112 x 112. Les clips vidéos sont recadrés autour de la boîte englobante élargie contenant l'acteur et les objets du contexte de l'action. Cette boîte englobante est détectée au moyen de OpenPose [1]. Ce qui permet de concentrer l'attention sur la zone où à lieu l'activité. Le réseau C3D prend une séquence d'action de taille fixe : 16 trames. En pratique, puisque l'on considère des actions atomiques, relativement courtes, on n'utilise pas une fenêtre glissante sur les séquences, mais plutôt simplement un ré-échantillonage des séquences.

Entraînement Les poids du réseau sont entraînés en utilisant une descente de gradient stochastique sur des mini-lots de taille 16 avec un momentum de 0.9. Nous initialisons le taux d'apprentissage à 0.01 et il diminue par palier, il est divisé par 10 après 20 époques. Nous utilisons la fonction de coût categorical cross-entropy. L'entraînement est fait sur une carte graphique GeForce GTX 1080 Ti.

Test. Les performances de notre approche hybride sont évaluées suivant le principe de la validation croisée de type *k-fold* avec $k = 4$. Chacun des échantillons forment une partition de l'ensemble des données du dataset. Chaque sous-échantillon est utilisé exactement une fois comme ensemble de validation lors de l'entraînement. Dans le jeu de données CAD-120 il y a quatre acteurs et chaque échantillon est associé à un acteur. Dans Watch-n-Patch la publication originale fournit un seul ensemble de test et d'entraînement, nous en avons généré 3 autres en prenant soin de garder les actions d'une même séquence au sein du même *fold*. Nous obtenons la prédiction finale au niveau de la dernière couche d'activation (softmax) présente dans les approches 2,3,4 décrites dans la section 3.4 et illustrées sur la Fig. 1.

Métriques pour l'évaluation. Nous évaluons les différentes variantes proposées dans la section 3.4 avec deux métriques. La première la justesse, dénommée micro-justesse par la suite, est définie comme suit :

$$\text{Justesse} = \frac{\text{Nombre de prédictions correctes}}{\text{Nombre total de prédictions}} \quad (2)$$

qui est le ratio des actions correctement reconnues et du nombre total d'actions à reconnaître. Nous l'appelons ainsi par opposition avec la macro-justesse. La macro-justesse est la moyenne des justesses pour chacune des classes. La macro-justesse donne le même poids à chacune des classes, indépendemment du nombre d'échantillons de cette classe dans le jeu de données. Cela permet de voir si ce sont seulement les classes les plus représentées qui sont correctement reconnues où si globalement l'ensemble des classes, y compris celles sous-représentées sont reconnues correctement. Ces deux métriques sont complémentaires dans l'évaluation des performances pour des jeux de données avec des classes non équilibrées.

TABLE 1 – Résultats de nos différentes approches sur Watch-n-Patch. Les performances considérées sont les macro-justesse (M) et la micro-justesse (μ).

Architecture	Échantillon 0		Échantillon 1		Échantillon 2		Échantillon 3		Moyenne		Écart-type	
	μ	M	μ	M	μ	M	μ	M	μ	M	μ	M
1 - ANBM	0.78	0.79	0.73	0.74	0.76	0.77	0.75	0.75	0.76	0.76	0.02	0.02
2 - C3D	0.72	0.65	0.73	0.64	0.75	0.69	0.74	0.64	0.74	0.66	0.01	0.02
3 - C3D-GRU	0.89	0.87	0.86	0.77	0.85	0.77	0.89	0.84	0.87	0.81	0.02	0.05
4 - C3D-GRU-ANBM-FD	**0.94**	**0.91**	**0.93**	**0.90**	**0.93**	**0.91**	**0.93**	**0.89**	**0.93**	**0.90**	**0.001**	**0.01**

TABLE 2 – Résultats de nos différentes approches sur CAD-120 [7]. Les performances considérées sont les macro-justesse (M) et la micro-justesse (μ) .

Architecture	Acteur 1		Acteur 2		Acteur 3		Acteur 4		Moyenne		Écart-type	
	μ	M	μ	M	μ	M	μ	M	μ	M	μ	M
1 - ANBM	0.84	0.77	0.78	0.81	0.82	0.76	0.82	0.77	0.82	0.78	0.03	**0.02**
2 - C3D	0.58	0.45	0.70	0.61	0.64	0.57	0.56	0.35	0.62	0.50	0.06	0.12
3 - C3D-GRU	0.61	0.49	0.76	0.73	0.66	0.60	0.60	0.45	0.66	0.57	0.07	0.13
4 - C3D-GRU-ANBM-FD	**0.86**	**0.80**	**0.89**	**0.91**	**0.84**	**0.82**	**0.83**	**0.79**	**0.86**	**0.83**	**0.03**	0.05

4.3 Résultats et discussion

Nous allons d'abord comparer les résultats individuels des approches C3D, C3D-GRU et ANBM décrites en section 3 avant d'évaluer leur fusion.

Nous évaluons ici l'apport de la couche récurrente GRU à la sortie de C3D pour prendre en compte la logique entre les actions. D'après le Tab. 1 sur Watch-n-Patch on observe en moyenne un gain en micro-justesse de +13 points de pourcentage (pp) et d'après le Tab. 2 un gain +4 pp sur CAD-120 grâce à l'ajout d'une couche GRU au réseau C3D par rapport à C3D seul. En regardant en détail les matrices de confusion des Fig. 5 et 6 on voit que les classes qui en ont le plus bénéficié sont les classes suivantes : *remettre un livre*, *poser un objet* et *prendre un objet*. En effet l'action *remettre un livre* est souvent précédée de l'action *lire*. Lorsque l'action précédente détectée est *lire* cela réduit et conditionne le choix des possibilités suivantes. L'action de *poser un objet* est elle fréquemment précédée par l'action *marcher*. En effet dans Watch-n-Patch il est fréquent qu'une personne entre dans le bureau en marchant et dépose son téléphone sur la table. Les gains sur CAD-120 sont plus modestes car pour que la couche récurrente apporte de l'information il faut que le réseau C3D gelé ait suffisamment bien appris à reconnaître les classes.

Le réseau C3D-GRU surpasse donc C3D, et maintenant nous le comparons avec notre approche ANBM avant d'évaluer leur fusion. Sur le jeu de données Watch-n-Patch, C3D-GRU a une meilleure micro-justesse que ANBM avec un écart de +11 pp mais l'amélioration de la macro-justesse est moins importante avec +5 pp, cf. lignes 1 et 3 du Tab.1. Comme on peut le voir sur la matrice de confusion de la Fig. 6, les actions les mieux détectées sont *lire*, *marcher* et *sortir* avec des scores respectivement de 1, 0.98, et 0.97. Ces actions représentent 49% des données ($12+16+21 = 49$ cf. Tab. 3) et contribuent plus à la micro-justesse que *éteindre l'écran* qui ne représente que 3%. La matrice de confusion de la Fig. 4 nous montre que l'approche ANBM surpasse C3D-GRU sur 3 classes : *jouer à l'ordinateur*, *allumer l'écran* et *éteindre l'écran*. Les deux approches offrent des approches complémentaires sur des classes différentes et leurs sources de confusions varient également. Comme on peut le voir sur les matrices de confusions des Figs. 4 et 6, les deux approches ont des performances similaires pour l'action *attraper un livre* (0.71 pour ANBM et 0.81 pour C3D-GRU) mais les erreurs diffèrent. En effet ANBM détecte parfois *prendre un objet* alors que C3D-GRU détecte à la place *poser un objet*. Sur CAD-120 le réseau C3D-GRU distingue mieux *atteindre* et *placer* que ANBM, ces deux actions représentent 38% du jeu de données ($23 + 15 = 38$ cf. Tab. 3).

Sur les deux jeux de données C3D-GRU et ANBM apportent donc des performances complémentaires, nous évaluations ici les bénéfices que l'on peut tirer de leur fusion. Sur CAD-120, la capacité pour le réseau C3D-GRU à distinguer *atteindre* et *placer* des autres classes permet lors de la fusion des deux approches d'avoir des gains de +4 pp en micro-justesse, voir Tab. 2. La fusion de C3D-GRU et ANBM a permis d'améliorer la reconnaissance de toutes les actions de Watch-n-Patch sauf de l'action *marcher* qui passe de 0.98 avec C3D-GRU à 0.97 dans la fusion C3D-GRU-ANBM comme le montre les matrices de confusion

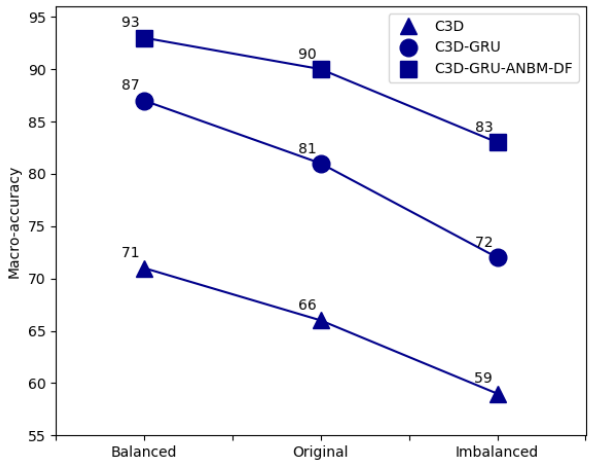


FIGURE 3 – Macro-justesse en fonction du déséquilibre des classes sur Watch-n-Patch. Les classes sont synthétiquement augmentées ou dégradées.

sur les Figs. 4 6 7. Dans l'ensemble la fusion permet d'augmenter la micro-justesse de +6 pp par rapport à C3D-GRU et de +17 pp par rapport à ANBM. Pour les actions mettant en jeu un ordinateur, dans la fusion ce sont les performances de ANBM qui sont privilégiées par rapport à celles de C3D-GRU. La complémentarité des deux approches est bien exploitée également lorsqu'ils présentent individuellement performances similaires pour une classe d'action. Par exemple dans la fusion l'action *attraper un livre* atteint 0.94.

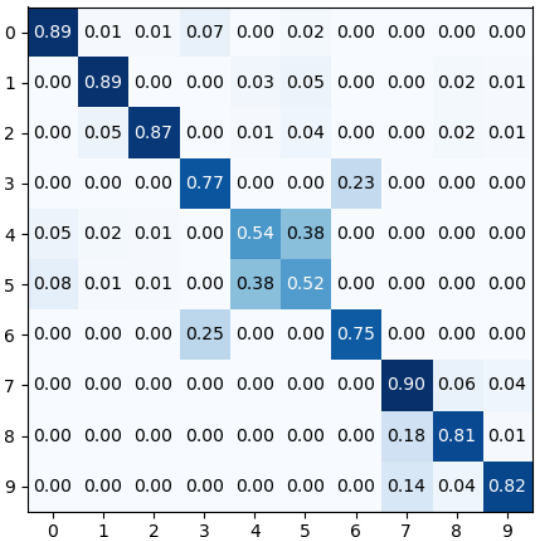
Nous proposons ici d'évaluer la robustesse de notre approche de fusion sur le jeu de données Watch-n-Patch en égalisant ou en dégradant le déséquilibre des classes. De manière synthétique on augmente ou dégrade le nombre d'échantillons des classes, puis on re-entraîne les réseaux C2D, C3D-GRU et C3D-GRU-ANBM-FD pour obtenir les résultats présentés sur la Fig. 3. On observe que l'entraînement de C3D est sensible au nombre d'échantillons. On observe également la dépendance de C3D-GRU au résultats obtenus par C3D seul. En effet la performance de C3D-GRU décroît plus rapidement que C3D, car pour capturer la cohérence temporelle, l'action précédente doit être bien détectée. Quand les classes sont fortement déséquilibrées, C3D détecte mal certaines actions et certaines transitions entre actions ne sont pas modélisées. Dans l'ensemble on note que la fusion C3D-GRU-ANBM-FD résiste mieux à la dégradation du nombre d'échantillons.

Comme le montre le Tab. 4, notre stratégie de fusion permet d'améliorer nos performances précédentes, tout en ayant des performances proches ou supérieures à celles de l'état de l'art. Nous avons choisi des approches récentes de référence de l'état de l'art, si possible s'évaluant sur les deux mêmes jeux de données telle que GEPHAPP [17]. Les deux approches considérées dans notre stratégie de fusion prennent en compte les transitions d'une action précédente vers une suivante. Or dans CAD-120, l'action *déplacer* précède presque toutes les autres, ce qui est peu déterminant. On y voit également que notre stratégie de fusion permet de surpasser l'état de l'art en reconnaissance d'actions sur le jeu de données Watch-n-Patch en améliorant la reconnaissance d'actions de +8.4 pp par rapport à l'approche proposée par Qi *et al* [17]. Une vidéo qui illustre nos résultats sur les séquences vidéos est disponible à l'adresse indiquée en pied de page [1].

TABLE 4 – Comparaison à la littérature. Justesse de la détection des actions sur CAD-120 et Watch-n-Patch.

Jeu de donnée	Approche	Justesse
CAD-120	GEPHAPP [17]	79.4
	ANBM [13]	82.2
	GPNN [18]	87.3
	Notre approche	**86.1**
Watch-n-Patch	WBTM [28]	35.2
	PoT [19]	49.93
	ANBM [13]	76.4
	GEPHAPP [17]	84.8
	Notre approche	**93.0**

FIGURE 4 – Matrices de confusion de ANBM sur l'échantillon 0 de Watch-n-Patch. [0 - lire ; 1 - marcher ; 2 - sortir ; 3 - attraper un livre ; 4 - remettre le livre ; 5 - poser un objet ; 6 - prendre un objet ; 7 - jouer à l'ordinateur ; 8 - allumer l'écran ; 9 - éteindre l'écran]. Prédictions sur les colonnes et vérité sur les lignes.



1. https ://youtu.be/Z-cgTcZvRiY

FIGURE 5 – Matrices de confusion de C3D sur l'échantillon 0 de Watch-n-Patch. [0 - lire ; 1 - marcher ; 2 - sortir ; 3 - attraper un livre ; 4 - remettre le livre ; 5 - poser un objet ; 6 - prendre un objet ; 7 - jouer à l'ordinateur ; 8 - allumer l'écran ; 9 - éteindre l'écran]. Prédictions sur les colonnes et vérité sur les lignes.

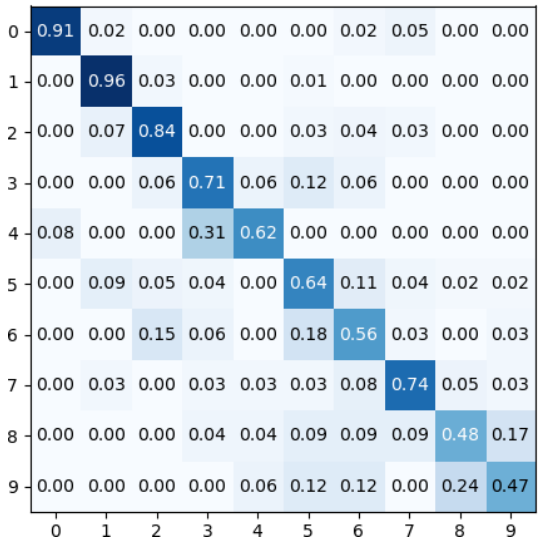


FIGURE 6 – Matrices de confusion de C3D-GRU l'échantillon 0 de Watch-n-Patch. [0 - lire ; 1 - marcher ; 2 - sortir ; 3 - attraper un livre ; 4 - remettre le livre ; 5 - poser un objet ; 6 - prendre un objet ; 7 - jouer à l'ordinateur ; 8 - allumer l'écran ; 9 - éteindre l'écran]. Prédictions sur les colonnes et vérité sur les lignes.

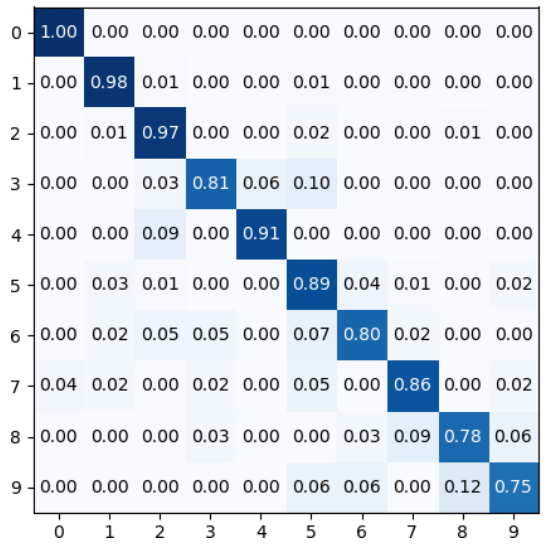
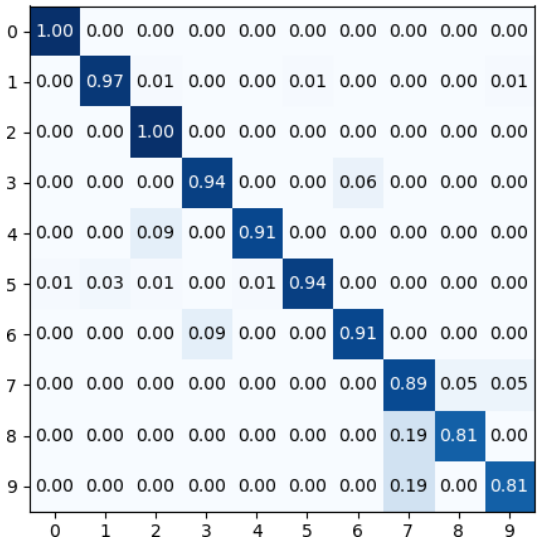


FIGURE 7 – Matrices de C3D-GRU-FD confusion sur l'échantillon 0 de Watch-n-Patch. [0 - lire ; 1 - marcher ; 2 - sortir ; 3 - attraper un livre ; 4 - remettre le livre ; 5 - poser un objet ; 6 - prendre un objet ; 7 - jouer à l'ordinateur ; 8 - allumer l'écran ; 9 - éteindre l'écran]. Prédictions sur les colonnes et vérité sur les lignes.



5 Conclusion et perspectives

Dans cet article nous avons comparé différentes approches de détection d'actions et proposé l'ajout de couche récurrente à C3D pour bénéficier des relations temporelles qui existent entre les actions. Nous avons exploré une manière de fusionner au niveau décisionnel deux approches de reconnaissance d'actions au moyen d'une couche dense. Nous les avons expérimentées sur deux jeux de données de la littérature présentant un déséquilibre entre les classes et nous avons montré des gains d'autant plus significatifs que les approches sont complémentaires. Dans les perspectives nous envisageons d'évaluer notre approche de fusion sur la détection d'activités de haut niveau.

Remerciements

Ces travaux ont été partiellement financés par Bpifrance dans le cadre du projet français LinTO issu de l'appel à projet Programme d'Investissements d'Avenir 3.

Références

[1] Z. Cao and T. Simon and S. E. Wei and Y. Sheikh, Realtime Multi-Person 2D Pose Estimation using Part Affinity Field, In CVPR, 2017.

[2] J. Carreira and A. Zisserman, Quo vadis, Action Recognition ? A New Model and the Kinetics Dataset, In CVPR, 2017.

[3] M. Everingham, S. A. Eslami, L. Van Gool, C. K Williams, J. Winn and A. Zisserman, The Pascal Visual Object Classes (VOC) challenge., International Journal of Computer Vision, 2010

[4] Y. Fan, X. Lu, D. Li and Y. Liu, Video-based emotion recognition using CNN-RNN and C3D hybrid networks, In ACM, 2016.

[5] C. Feichtenhofer, A. Pinz, and A. Zisserman, Convolutional two-stream network fusion for video action recognition, In CVPR, 2016.

[6] S. Ji, W. Xu, M. Yang,and K. Yu, 3D convolutional neural networks for human action recognition, PAMI, 2012.

[7] H. S. Koppula, R. Gupta and A.Saxena, Learning human activities and object affordances from rgb-d videos, The International Journal of Robotics Research, pp. 951-970, 2013.

[8] W. Li, Z. Zhang, Z. Liu, Action recognition based on a bag of 3D points, In CVPRW, 2010.

[9] T-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco : Common objects in context, In ECCV, 2014.

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, SSD : Single Shot MultiBox Detector, In ECCV, 2016.

[11] K. Liu, W. Liu, G. Wu, M. Tan and H. Ma, T-C3D : temporal convolutional 3d network for real-time action recognition, In AAAI 2018

[12] N. Lu, Y. Wu, L. Feng and J. Song, Deep learning for fall detection : Three-dimensional CNN combined with LSTM on video kinematic data, Journal of Biomedical and Health Informatics, pp. 314-323, 2018.

[13] C. Maurice, F. Madrigal, A. Monin and F. Lerasle, A New Bayesian Modeling for 3D Human-Object Action Recognition, In AVSS, 2019.

[14] T. B. Moeslund, A. Hilton and V. Krüger, A survey of advances in vision-based human motion capture and analysis, CVIU, pp. 90-126, 2006.

[15] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks, In CVPR, 2014.

[16] S. Qi, S. Huang, P. Wei, and S. C. Zhu, Predicting human activities using stochastic grammar, In ICCV, 2017

[17] S. Qi, B. Jia, S. Huang, P. Wei and S. C. Zhu, A Generalized Earley Parser for Human Activity Parsing and Prediction, In PAMI, 2019.

[18] S. Qi, W. Wang, B. Jia, J. Shen, and S. C. Zhu, Learning human-object interactions by Graph Parsing Neural Networks, In ECCV, 2018.

[19] M.S. Ryoo, B. Rothrock and L. Matthies,, Pooled motion features for first-person videos In CVPR, 2015

[20] A. Shahroudy, J. Liu, T. T. Ng and G. Wang, Ntu rgb+ d : A large scale dataset for 3d human activity analysis, In CVPR, 2016.

[21] K. Soomro, A. Roshan Zamir and M. Shah, UCF101 : A Dataset of 101 Human Action Classes From Videos in The Wild, 2012.

[22] P. Turaga, R. Chellappa, V. S. Subrahmanian and O. Udrea, Machine Recognition of Human Activities : A Survey, Transactions on Circuits and Systems for Video Technology, 2008.

[23] D. Tran, L. Bourdev, R. Fergus, L. Torresani and M. Paluri, Learning spatiotemporal features with 3d convolutional networks, In ICCV, 2015.

[24] D. Tran, J. Ray, Z. Shou, S-F. Chang, M. Paluri, Convnet architecture search for spatiotemporal feature learning, arXiv preprint arXiv :1708.05038, 2017.

[25] G. Varol, I. Laptev, C. Schmid, Long-term temporal convolutions for action recognition, PAMI, pp. 1510-1517, 2017.

[26] X. Wang, L. Gao, J. Song and H. Shen, Beyond frame-level CNN : saliency-aware 3-D CNN with LSTM for video action recognition, In IEEE Signal Processing Letters, 2016

[27] C. Wu, J. Zhang, S. Savarese and A. Saxena, Watch-n-patch : Unsupervised understanding of actions and relations, In CVPR, 2015.

[28] C. Wu, J. Zhang, B. Selman, S. Savarese, and A. Saxena, Watch-bot : Unsupervised learning for reminding humans offorgotten actions, In ICRA, 2016.

[29] G. Yao, X. Liu and T. Lei, Action recognition with 3d convnet-gru architecture, In Proceedings of the 3rd International Conference on Robotics, Control and Automation, 2018.

[30] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, Learning deep features for discriminative localization, In CVPR, 2016.