Étude de la classification des vidéos de foule par apprentissage profond

Mounir Bendali-Braham

Année 2022 N° d'ordre : (attribué par le SCD)

Université De Haute-Alsace
Université De Strasbourg

Thèse

Pour l'obtention du grade de
Docteur de l'Université de Haute-Alsace
École Doctorale : Mathématiques, Sciences de l'Information et de l'Ingénieur
(ED 269) Discipline : Informatique

Présentée et soutenue publiquement

par

Mounir Bendali-Braham

Le 27 janvier 2022

Étude de la Classification des Vidéos de Foule par Apprentissage Profond

---

Sous la direction du Prof. Pierre-Alain MULLER

Jury

Prof. Hazem WANNOUS, IMT Lille-Douai (Rapporteur)
Dre. HDR. Sylvie CHAMBON, Toulouse INP (Rapportrice)
Dr. Fabien PIERRE, Université de Lorraine (Examinateur)
Prof. Antoine VACAVANT, Université Clermont Auvergne (Examinateur)
Prof. Pierre-Alain MULLER, Université de Haute-Alsace (Directeur de thèse)
Prof. Lhassane IDOUMGHAR, Université de Haute-Alsace (co-Directeur de thèse)
Prof. Germain FORESTIER, Université de Haute-Alsace (co-Directeur de thèse)
Dr. HDR. Jonathan WEBER, Université de Haute-Alsace (Encadrant de thèse)

UNIVERSITÉ DE HAUTE-ALSACE

Résumé

Faculté des Sciences et Techniques
Institut de Recherche en Informatique, Mathématiques, Automatique et Signal

Doctor of Philosophy

Étude de la Classification des Vidéos de Foule par Apprentissage Profond

Du fait de la multiplication des rassemblements dans les grandes villes, leur encadrement constitue un enjeu majeur pour les forces de l'ordre. Fort heureusement, les forces de l'ordre peuvent compter sur la présence des caméras de vidéoprotection pour optimiser leur intervention. Toutefois, le traitement automatique des vidéos qu'elles récoltent n'est pas réalisé de manière systématique, ce qui retarde le temps de réaction des forces de l'ordre. Cette thèse apporte une pierre à l'édifice de cette automatisation en mettant en place des modèles réalisant une classification des vidéos de comportements de foule. Après avoir étudié les travaux existants en analyse de foule et proposé une taxonomie permettant de classer les travaux de l'état-de-l'art, nous avons proposé de classer les clips vidéo issus du jeu de données Crowd-11. Ce jeu de données comporte onze catégories de mouvements assez représentatives des comportements de foule pouvant se dérouler dans les espaces publics et privés.

Dans le cadre d'un travail préliminaire sur l'apprentissage par transfert, nous avons étudié et analysé la classification des clips vidéo de foule réalisée par des réseaux de neurones convolutifs. Nos meilleurs réseaux classent correctement la majeure partie des clips de l'ensemble de test. Toutefois, une étude plus approfondie de leurs résultats nous a permis de constater qu'ils éprouvent des difficultés avec trois classes de mouvements assez similaires. Afin de pallier les limites rencontrées par nos premiers réseaux, nous avons proposé deux nouvelles approches : la première approche exploite la détection de têtes, en tant qu'étape de pré-traitement, avant de recourir à la classification. La deuxième approche se base sur la classification ensembliste qui associe les connaissances de plusieurs méthodes de classification permettant de produire de meilleures décisions.

Dans le cadre de l'exploitation de la détection de têtes, nous nous sommes basés sur les déplacements des têtes des personnes détectées dans les clips vidéo de foule. Les positions des têtes, extraites de toutes les personnes apparaissant dans les scènes du jeu de données, ont été rassemblées dans des cartes de détection de têtes. Ces cartes ont été envoyées en entrée des réseaux de neurones convolutifs pour caractériser les comportements de foule et les classer.

Dans le cadre de la classification ensembliste, plusieurs modèles entraînés différemment ont été réunis dans un ensemble pour classer des clips vidéo de foule. Huit ensembles de modèles homogènes ont été étudiés et analysés. Par la suite, des ensembles de modèles hétérogènes ont été mis en place pour étudier toutes les combinaisons possibles des ensembles de modèles homogènes. L'objectif de cette étude a été de trouver la combinaison qui associe le mieux les compétences de chaque ensemble homogène pour obtenir l'ensemble de modèles hétérogènes le plus performant.

UNIVERSITÉ DE HAUTE-ALSACE

Abstract

Faculté des Sciences et Techniques
Institut de Recherche en Informatique, Mathématiques, Automatique et Signal

Doctor of Philosophy

Étude de la Classification des Vidéos de Foule par Apprentissage Profond

Due to the proliferation of gatherings in large cities, their supervision represents a major issue for the security forces. Fortunately, security forces can count on the presence of video-surveillance cameras to optimize their intervention. However, the automatic processing of the videos they collect is not done systematically, which delays the reaction time of security forces. This thesis sets up a building block for this automation by proposing classification models for video-recorded crowded scenes. After studying existing work in crowd analysis and proposing a taxonomy for classifying state-of-the-art work, we proposed to classify video clips from the Crowd-11 dataset. This dataset contains eleven categories of crowd movements that are representative of crowd behaviors that can occur in public and private spaces.

As part of preliminary work on transfer learning, we investigated and analyzed the classification of crowd video clips performed by convolutional neural networks. Our top networks correctly categorize most of the clips in the test set. However, a closer study of their results showed us that they encounter difficulties with three similar classes of crowd movements. In order to overcome the limitations encountered by our first networks, we have proposed two new approaches : the first one leverages heads detection, as a pre-processing step, before applying video classification. The second one relies on ensemble classification which combines the knowledge of several classification methods to produce better decisions.

In the context of heads detection, we relied on the heads displacements to describe crowd movements. Heads positions, taken from all of the individuals appearing in the scenes were included into heads detection maps. These maps were sent as input to convolutional neural networks to characterize crowd behaviors and classify them.

As part of the ensemble classification, several models trained differently were brought together into an ensemble to classify crowd video clips. Eight ensembles of homogeneous models were compared. Afterthat, ensembles of heterogeneous models were set up to study all possible combinations of ensembles of homogeneous models. The purpose of this study was to find the combination that best combines the knowledge of all homogeneous ensembles to obtain the best performing ensemble of heterogeneous models.

Remerciements

Je remercie chaleureusement toutes les personnes qui m'ont assisté pendant l'élaboration de ce travail, dans la rédaction de la thèse et des articles ainsi que tous ceux qui m'ont apporté leur précieuse aide et leur soutien, et qui sans eux, ce travail de thèse n'aurait pas pu toucher à sa fin.

J'exprime ici mes vifs remerciements et toute ma reconnaissance envers :

Mon directeur de thèse, Professeur Pierre-Alain Muller, Président de l'Université de Haute-Alsace, pour m'avoir proposé ce sujet, pour m'avoir accueilli au sein de son institution, pour m'avoir encadré et dirigé. Qu'il trouve ici ma profonde gratitude, ma reconnaissance et mon profond respect.

Mes co-directeurs de thèse :

Professeur Lhassane Idoumghar, Directeur de l'IRIMAS, Université de Haute-Alsace, pour m'avoir accueilli au sein de son Institut, pour ses qualités professionnelles, pour son intérêt et sa rigueur scientifique. Qu'il trouve ici l'expression de ma profonde gratitude et de ma reconnaissance.

Professeur Germain Forestier, responsable de l'équipe Modélisation et Science des Données (MSD) et Directeur du département Informatique au sein de l'IRIMAS, pour sa disponibilité en toutes circonstances, ses nombreux conseils durant la rédaction de ma thèse, et pour sa rigueur scientifique. Qu'il trouve ici l'expression de ma profonde gratitude et de ma reconnaissance.

Mon encadrant de thèse, Docteur HDR. Jonathan Weber, maître de conférence (IRIMAS-Université de Haute-Alsace), pour sa disponibilité en toutes circonstances, sa patience, sa gentillesse, ses multiples conseils, son assistance et pour tout le temps qu'il m'a consacré. Qu'il trouve ici ma profonde gratitude et ma reconnaissance.

Je tiens à remercier les membres du jury, Professeur Hazem Wannous et Docteure HDR. Sylvie Chambon, pour m'avoir fait l'honneur d'accepter d'examiner mon travail de thèse.

Je remercie également les autres membres du jury, Professeur Antoine Vacavant et Docteur Fabien Pierre, pour m'avoir fait l'honneur d'évaluer ce travail.

Je remercie mes collègues, membres du projet ANR OpMOPS, pour les échanges enrichissants que nous avons pu avoir sur la gestion des mouvements de foule. Je remercie particulièrement Dre. Cassandra Rotily et Dr. Julien Kritter pour leur amitié et leur soutien.

Je remercie Dr. Bertrand Luvison, un des concepteurs du jeu de données Crowd-11, pour m'avoir assisté dans l'acquisition rapide de son contenu.

Je remercie l'Agence Nationale de la Recherche pour le financement de ce projet de thèse (subvention ANR-16-SEBM-0004) et le Mésocentre de Strasbourg pour m'avoir permis de mener des calculs sur leur cluster de GPUs et pour l'aide technique qu'ils m'ont apporté pour mener à bien mes expériences. Je remercie la société Nvidia pour la fourniture de cartes GPU à notre équipe au sein de l'IRIMAS. Je remercie mes collègues au sein de l'IRIMAS et de l'Université de Haute-Alsace et tout particulièrement celles et ceux avec qui j'ai pu avoir des échanges constructifs au sein du bureau LSI au cours de ces quatre années de thèse : Hassan, Robin, Bastien, Baptiste, Hojjat, et tous ceux que j'ai omis de citer involontairement. Je remercie particulièrement Dr. Hassan Ismail Fawaz pour m'avoir accompagné dans mon apprentissage du Deep Learning. Je remercie d'autres collègues en dehors du bureau LSI avec qui j'ai eu l'occasion de travailler à l'instar de Dr. Hojjat Rakhshani. Je remercie mes collègues au sein de l'IUT de Saint-Dié-des-Vosges qui m'ont offert et continuent de m'offrir, pendant la durée de mon contrat ATER, un cadre de travail chaleureux. Je

tiens à remercier mes encadrantes de stage de Master 2, Dre. Ndèye Niang Keita et Pre. Sylvie Thiria, qui m'ont encouragé à m'engager dans cette aventure académique.

Je remercie Dr. Djallel Dilmi de m'avoir soutenu tout au long de la réalisation de ce projet de thèse. Il n'a pas cessé de croire en moi, de m'encourager et de me pousser vers l'avant, surtout dans les moments difficiles. Je lui dois en partie ma réussite. Qu'il trouve ici le témoignage de ma profonde gratitude et ma reconnaissance.

Je remercie également mon ami de longue date ElKhalil Betrouni, qui m'a accompagné sur le plan scientifique du lycée jusqu'à mon doctorat, et mes amis Robin Barkas, Khalil Yala, Frédéric Haykal, Abdelhak Djouahra, Nafis Fateri, Honaïn Mohib Derrar, Hatem Bouabana, Samy Melaïne, Nassim Guerroumi.

Enfin, je souhaiterais remercier mes parents, ma tante Fifi, et mon défunt oncle Schahriar, qui ont été et me sont, jusqu'à ce jour, d'une aide incommensurable dans tous les projets que j'entreprends.

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

OF	Optical Flow
RVB	Rouge Vert Bleu
C3D	3D Convolutional Network
I3D	Inflated 3D
2S-I3D	TwoStream Inflated 3D
3S-I3D	ThreeStream Inflated 3D
R3D	3D Residual Network
CNN	Convolutional Neural Networks
RCN	Réseaux de Neurones Convolutifs
SVM	Support Vector Machine
MLP	Multi Layer Perceptron
PMC	Perceptron Multicouche
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
LSTM	Long-short Term Memory

Introduction Générale

Contexte

Dans les villes densément peuplées, les rassemblements liés aux activités de la vie quotidienne ou à des manifestations à caractère culturel ou revendicatif sont fréquents. L'encadrement de ces rassemblements constitue un enjeu majeur pour les forces de l'ordre. L'Histoire nous a appris qu'un rassemblement mal contrôlé peut mettre en danger la sécurité des individus qui le forment : des événements nous le rappellent tels que la bousculade meurtrière survenue lors de la Loveparade à Berlin en 2010 (Krausz et Bauckhage, 2012), et l'engorgement dramatique survenu au pèlerinage de la Mecque en 2015 (Bolia, 2015). Malgré l'omniprésence des caméras et leur positionnement optimal (Kritter et al., 2019), le traitement automatique du flux qu'elles produisent n'est actuellement pas mis en place de manière systématique, ce qui retarde le temps de réaction des forces de l'ordre.

Les forces de l'ordre ont besoin d'algorithmes qui puissent reconnaître des comportements de foule et soient capables de les prédire (Güler, 2012). Pour réussir l'étape de la prédiction des mouvements et des comportements de foule, il faut tout d'abord réussir à la perfection l'étape de leur classification.

Au vu de l'intérêt suscité par l'usage de l'apprentissage profond (Deep Learning) pour la vision par ordinateur et ses prouesses remarquables dans la classification d'images (Krizhevsky, Sutskever et Hinton, 2012), nous avons décidé d'employer ces méthodes pour classer des vidéos de foule.

Objectif

L'objectif de cette thèse est d'analyser et d'étudier des méthodes de classification des vidéos de comportements de foule et d'en proposer des nouvelles. Cet objectif se décline en deux parties successives :

1. Étude et analyse de méthodes existantes de classification des vidéos de foule ;
2. Proposition et mise en œuvre de nouvelles méthodes de classification.

Tout d'abord, une sélection a été faite de méthodes récentes et efficaces dans la classification des actions humaines dans des scènes individuelles. Celles-ci ont été étudiées et analysées dans le cadre de la classification des vidéos de foule.

Ensuite, de nouvelles méthodes de classification ont été proposées : D'une part, des approches qui focalisent l'intérêt de la classification sur des zones ciblées d'une scène en utilisant des méthodes de détection comme étape de pré-traitement. D'autre part, des approches qui associent les connaissances de différentes méthodes de classification, ce qui permet, *in fine*, de prendre de meilleures décisions.

L'indisponibilité en grande quantité de données représentatives de la totalité des mouvements de foule qui puissent survenir dans l'espace public présente un verrou technique. Cette indisponibilité est dûe, entre autres, à la durée limitée de conservation des vidéos issues des caméras de vidéo-protection et les règles protégeant l'identité

des personnes. Ce verrou a été partiellement levé grâce à l'existence du jeu de données Crowd-11.

Contributions et plan

Ce manuscrit est divisé en trois chapitres. Le premier chapitre aborde les tendances récentes en analyse de foule. Tout d'abord, les articles d'état-de-l'art de chaque thème de l'analyse de foule ont été analysés. Cette analyse a permis d'établir une taxonomie qui divise l'analyse de foule en différentes catégories. Par la suite, une étude a été réalisée sur les travaux récents qui ont été publiés en analyse de foule, et dans une certaine mesure, en reconnaissance des actions humaines. Les méthodes proposées pour la reconnaissance des actions humaines peuvent être appliquées en analyse des comportements de foule. Un répertoire des jeux de données les plus utilisés dans le domaine a été dressé. Les données les plus étudiées portent essentiellement sur quelques tâches communes en analyse de foule qui sont l'analyse des trajectoires, la reconnaissance d'actions dans des scènes individuelles, et le calcul des statistiques d'une scène de foule. Notre intérêt se porte sur la reconnaissance des mouvements de foule, un axe de recherche peu exploré en analyse de foule du fait de la rareté des données. Nous avons noté l'existence de certains jeux de données essentiels à cet axe de recherche tels que le jeu de données Crowd-11. Pour faire face à cette rareté de données et pour créer des jeux de données semblables ou plus volumineux que Crowd-11, il est essentiel de se pencher sur le développement des annotateurs. Le premier chapitre a dressé une analyse de quelques annotateurs récents utilisés pour des tâches liées à l'analyse de foule, et a proposé des pistes permettant de mettre en place des annotateurs qui répondent aux besoins actuels de l'analyse des comportements de foule.

Dans le deuxième chapitre, nous avons entraîné des Réseaux de Neurones Convolutifs à classer des clips vidéo. Les modèles statistiques que nous avons entraînés se basent sur un apprentissage de vidéos de mouvements de foule provenant du jeu de données Crowd-11. Ce jeu de données développé par (Dupont, Tobias et Luvison, 2017), est constitué de 6272 vidéos réparties en 11 classes de mouvements.

Nos meilleurs modèles réussissent à bien classer 68% des clips de l'ensemble de test. Une étude plus approfondie a permis d'observer qu'ils éprouvent le plus de difficultés avec 3 classes de mouvements assez similaires : 3. Turbulent Flow, 5. Merging Flows, et 6. Diverging Flow. Afin de pallier les limites que ces réseaux rencontrent, nous avons mis en place, dans un premier temps, l'exploitation de la détection de têtes, comme une étape de pré-traitement, pour la classification, et dans un second temps, l'usage de la classification ensembliste.

La détection de têtes a été exploitée pour suivre les déplacements des têtes des personnes détectées dans les scènes de foule. À partir d'un détecteur de têtes, nous avons produit des vidéos de cartes de détection de têtes. Dans une image d'une foule, une carte de détection de têtes contient les centres de toutes les têtes détectées, le reste de l'image est ignoré. Notre but est d'entraîner un réseau à trois branches *ThreeStream* issu de l'architecture Inflated 3D (Carreira et Zisserman, 2017) sur les vidéos de foule, leurs flux optiques et les vidéos de cartes de détection des têtes associées afin d'étudier l'apport de chaque information sur les performances des modèles lors de la classification.

Dans le troisième chapitre, nous avons réuni plusieurs modèles entraînés différemment dans un ensemble pour classer des clips de Crowd-11. Dans un premier temps, nous avons entraîné huit ensembles de modèles homogènes provenant de différentes

architectures ayant ou non bénéficié d'une augmentation de données afin de trouver la stratégie d'entraînement qui produit les meilleurs résultats en termes de précision globale. Par la suite, nous avons créé des ensembles de modèles globaux pour tester toutes les combinaisons possibles des modèles déjà comparés. À la suite de cette comparaison, nous avons choisi d'analyser la combinaison qui associe le mieux les compétences de chaque modèle pour obtenir l'ensemble de modèles hétérogènes le plus performant.

Dans la conclusion, nous avons passé en revue nos contributions ainsi que les résultats que nous avons obtenus au cours de nos expériences, et nous avons proposé des perspectives qui incitent d'une part à se pencher sur l'aspect temps réel de nos méthodes de classification, et d'autre part d'expliquer leurs décisions de classification.

Cette thèse a été financée par l'Agence Nationale de la Recherche (ANR) et entre dans le cadre du projet franco-allemand OPMoPS [1] (ANR-16-SEBM-0004).

Liste des publications

Article de journal

1. Mounir Bendali-Braham, Jonathan Weber, Germain Forestier, Lhassane Idoumghar, Pierre-Alain Muller. Recent trends in crowd analysis : A review. Machine Learning with Applications. 2021.

Communications internationales

1. Mounir Bendali-Braham, Jonathan Weber, Germain Forestier, Lhassane Idoumghar, Pierre-Alain Muller. Ensemble classification of video-recorded crowd movements. 12th International Symposium on Image and Signal Processing and Analysis (ISPA). 2021.
2. Mounir Bendali-Braham, Jonathan Weber, Germain Forestier, Lhassane Idoumghar, Pierre-Alain Muller. Transfer learning for the classification of video-recorded crowd movements. 11th International Symposium on Image and Signal Processing and Analysis (ISPA). 2019.

Communication nationale

1. Mounir Bendali-Braham, Jonathan Weber, Germain Forestier, Lhassane Idoumghar, Pierre-Alain Muller. Classification ensembliste de vidéos de mouvements de foule. ORASIS 2021.

Atelier

1. Mounir Bendali-Braham, Jonathan Weber, Germain Forestier, Lhassane Idoumghar, Pierre-Alain Muller. Apprentissage par transfert pour la classification de séquences vidéo de mouvements de foule. Apprentissage Profond : Théorie et Applications (APTA) EGC 2020.

Divers

1. Rotily Cassandra, Soheila Ghambari, Julien Kritter, Mounir Bendali-Braham. Préparation et gestion des manifestations à fort potentiel de conflit : le projet OPMoP. Revue de la Gendarmerie Nationale. 2020.

1. https://anr.fr/Projet-ANR-16-SEBM-0004

Chapitre 1

Tendances récentes en analyse de foule

1.1 Introduction

La croissance de la population mondiale et sa citadinisation conduisent à l'augmentation du nombre de villes et de grandes villes aux quatre coins du monde. Certaines de ces villes sont sujettes à des mouvements de foule massifs dus à des événements ponctuels à l'instar des parades urbaines, à des rassemblements à caractère culturel, politique, ou social (Latour et Pauvert, 2018), ou bien à des scènes banales de la vie quotidienne tels que les rassemblements aux entrées et sorties des gares ferroviaires, aux intérieurs de supermarchés, de théâtres, de cinémas, ou de musées, etc. Ces événements peuvent provoquer de multiples problèmes de sécurité qui nécessitent l'intervention des forces de l'ordre pour les gérer (Krausz et Bauckhage, 2012). Malheureusement, malgré tous les efforts organisationnels déployés, certaines situations, parfois dramatiques, demeurent inévitables. Certaines de ces situations sont dûes à l'inexistence d'outils prédictifs qui permettent d'accroître l'anticipation des forces de l'ordre. Pour y faire face, les systèmes de surveillance basés sur des caméras de vidéo-protection déjà mis en place peuvent être mis à contribution (Porikli et al., 2013). Ces systèmes multi-caméras sont, le plus souvent, supervisés par des agents humains. Ces agents se trouvent devant des postes de travail multi-écrans à visualiser, des heures durant, la transmission de ces caméras. Maintenir une vigilance accrue pendant un tel laps de temps n'est pas une chose aisée, et peut souvent donner lieu à des erreurs humaines.

Avec l'émergence des villes intelligentes (Russell, 2015), il devient utile de recourir à des systèmes de surveillance intelligents qui font appel à des algorithmes de reconnaissance automatique des comportements humains dans des scènes individuelles et dans des scènes de foule (Isern et al., 2020). De nombreuses études ont été menées pour comprendre le comportement des humains dans des scènes de foule (Walia et Kapoor, 2016). Ces études appartiennent au domaine de l'analyse de foule. Selon de nombreux travaux (Lamba et Nain, 2017 ; Grant et Flynn, 2017 ; Zhan et al., 2008), l'analyse de foule est subdivisée en deux domaines principaux : le calcul des statistiques de foule et l'analyse des comportements de foule. Le but du calcul des statistiques de foule est d'estimer la densité d'une foule au moyen de méthodes de comptage du nombre de personnes dans une foule ou au moyen de méthodes de régression. La métrique la plus utilisée pour évaluer la densité de la foule est le niveau de service d'une foule (Level of Service (LoS)). Cette métrique est tirée du domaine de l'estimation des flux de trafic véhiculaire (Grant et Flynn, 2017). Le but de l'analyse des comportements de foule (Crowd Behavior Analysis) est d'étudier le comportement d'une foule. Ce domaine est généralement subdivisé en deux domaines principaux : suivi des trajectoires dans une foule (crowd motion tracking),

et l'analyse des actions dans une scène de foule (crowd action recognition) (Grant et Flynn, 2017 ; Lamba et Nain, 2017).

Des travaux récents ont saisi l'importance de déplacer l'intérêt de l'analyse des comportements de foule vers la détection et la prédiction des mouvements et des comportements de foule (Li et al., 2015 ; Thida et al., 2013). Ceci a pour effet de partager l'analyse des comportements de foule en deux sujets : l'analyse des trajectoires et la reconnaissance des actions de foule. Au vu de l'intérêt suscité par l'usage de l'apprentissage profond (Deep Learning) pour la vision par ordinateur, nous avons été témoins des capacités de prédiction offertes par cette catégorie de méthodes (Krizhevsky, Sutskever et Hinton, 2012). Avant de nous plonger dans la présentation des travaux publiés ces dernières années en analyse de foule, nous avons exploré l'état-de-l'art de cet axe de recherche. Cette exploration est l'objet de la section 1.2.

Dans la section 1.4, nous avons mentionné principalement des travaux récents en analyse de foule. En outre, nous nous sommes focalisés sur les thèmes suivants :

— L'usage des réseaux de neurones profonds (Deep Neural Networks) dans les travaux récents en analyse de foule, sans exclure des travaux tout autant récents qui reposent toujours sur l'extraction manuelle des indices visuels et utilisent des méthodes classiques issues de l'apprentissage automatique ;
— L'apprentissage profond (Deep Learning) étant omniprésent dans la vision par ordinateur, nous ne consacrons pas de section à l'extraction des caractériques, mais nous parlons de la détection de certains objets importants pour l'analyse de foule qui sont les piétons et les groupes ;
— Nous avons mis en lumière les thèmes de l'analyse de foule qui restent peu explorés dans la littérature ;
— Nous avons dressé une liste des sources de données : qu'elles proviennent de la vidéosurveillance en direct ou de jeux de données privés ou publics ;
— Et enfin, nous avons abordé l'usage des annotateurs. Leur contribution est nécessaire pour enrichir l'analyse de foule de nouveaux jeux de données. De nombreux domaines de l'analyse de foule sont encore à leur genèse en raison de la rareté des données.

Ce chapitre est organisé comme suit : **La section 1.2** rassemble un résumé de l'état-de-l'art de l'analyse de foule et ses domaines. Beaucoup de ces articles adoptent une taxonomie particulière pour décrire notre axe de recherche. Certains articles d'état-de-l'art ne présentent pas l'analyse de foule dans sa totalité, mais analysent l'un de ses domaines. Par conséquent, nous avons divisé cette section en sous-sections en fonction du thème présenté. **La section 1.3** est dédiée à la détection des piétons et des groupes. Nous n'avons pas inclus ces travaux dans la section de l'étude comparative des travaux récents en analyse de foule (Section 1.4), car nous considérons la détection des piétons et des groupes comme des outils essentiels pour l'analyse de foule mais pas comme un thème en soi. **La section 1.4** présente la partie de la littérature récente sur l'analyse de foule. La section est divisée en deux thèmes principaux : le calcul des statistiques de foule et l'analyse des comportements de foule. La plupart des travaux mentionnés sont basés sur l'apprentissage profond (Deep Learning). **La section 1.5** présente toutes les sources de données que nous avons rencontrées. Ces sources de données se subdivisent : en sites de diffusion en temps réel, et en jeux de données publics et privés. Les jeux de données que nous avons mentionnés sont pertinents à l'analyse de foule et sont très souvent utilisés. Certains d'entre eux sont utilisés pour la détection de piétons et de groupes. **La section 1.6** est dédiée à la description de certains annotateurs en vogue : outils dont nous estimons le développement vital à l'enrichissement et la pérennisation de l'analyse de foule.

1.2 État-de-l'art sur l'analyse de foule

Depuis une vingtaine d'années, de nombreux articles d'état-de-l'art ont été rédigés sur l'analyse de foule. Certains d'entre eux étudient l'axe de recherche dans son ensemble, d'autres se concentrent sur l'un de ses domaines. Dans cette section, nous présentons les articles d'état-de-l'art qui nous semblent les plus emblématiques. Pour établir une base solide pour nos travaux futurs, nous avons analysé une douzaine d'articles et les avons séparés en sections. Chacun des articles suivants adopte dès le départ une taxonomie pour classer les travaux analysés en analyse de foule. Certaines taxonomies sont redondantes et d'autres sont uniques.

1.2.1 L'analyse de foule

L'état-de-l'art que nous avons mentionné, dans cette section, présente une vue panoramique du domaine de l'analyse de foule.

L'état-de-l'art de (Zhan et al., 2008) fournit un aperçu de ce qui a été fait en analyse de foule jusqu'en 2008. L'article synthétise tous les travaux partant du calcul des statistiques de foule à l'analyse des comportements de foule. (Zhan et al., 2008) donnent un aperçu sur l'extraction des caractéristiques qui permet par la suite l'estimation de la densité d'une scène, le comptage du nombre de personnes dans une foule, la reconnaissance d'actions dans des scènes individuelles, et le suivi de la trajectoire des piétons. Ils présentent également trois taxonomies différentes adoptées en analyse de foule :

1. La dichotomie entre le calcul des statistiques de foule et l'analyse des comportements de foule.
2. La division des travaux en études macroscopiques, microscopiques, et mésoscopiques (proposée par la Federal Highway Administration, en premier lieu (FHWA, 2004)).
3. La division en travaux issus de la vision par ordinateur, approches basées sur la mécanique des fluides et la physique, et les approches basées sur la sociologie.

Une visualisation de ces taxonomies est proposée dans la figure 1.1.

(Grant et Flynn, 2017) étudient l'analyse de foule et la divisent en deux grandes catégories : le comptage du nombre de personnes dans une foule et l'analyse des comportements de foule. Une visualisation de la taxonomie, qu'ils proposent, peut être observée dans la figure 1.2. Les auteurs montrent que les travaux antérieurs sur la reconnaissance de l'activité humaine portent sur des scènes individuelles où il y a rarement plus d'une personne qui apparaît dans une scène. L'intérêt pour les actions de groupe ou les actions au sein d'une foule est apparu plus tardivement dans les thématiques de l'analyse de foule. Dans leur article, pour parler de ce qui a trait à l'analyse des comportements de foule, (Grant et Flynn, 2017) mentionnent des travaux sur l'analyse des comportements de groupe, la détection d'événements anormaux dans des scènes de foule, et l'analyse des mouvements de foule. Pour cette dernière, on fait référence ici à l'analyse des trajectoires de piétons. En ce qui concerne le calcul des statistiques de foule, les auteurs évoquent l'utilisation du niveau de service (Level of Service (LoS)) (TRB, 2000), dans les travaux estimant la densité des scènes de foule. Comme le soulignent les auteurs, les travaux analysés ne s'attaquent pas aux scènes complexes, où les défis sont multiples. Ces travaux privilégient plutôt des scènes à faible densité peu sujettes aux occlusions, aux vibrations de la caméra, à sa mobilité, et au changement d'intensité lumineuse. Ce manque de défi facilite souvent la tâche des méthodes employées. Les travaux rapportés par (Grant et Flynn,

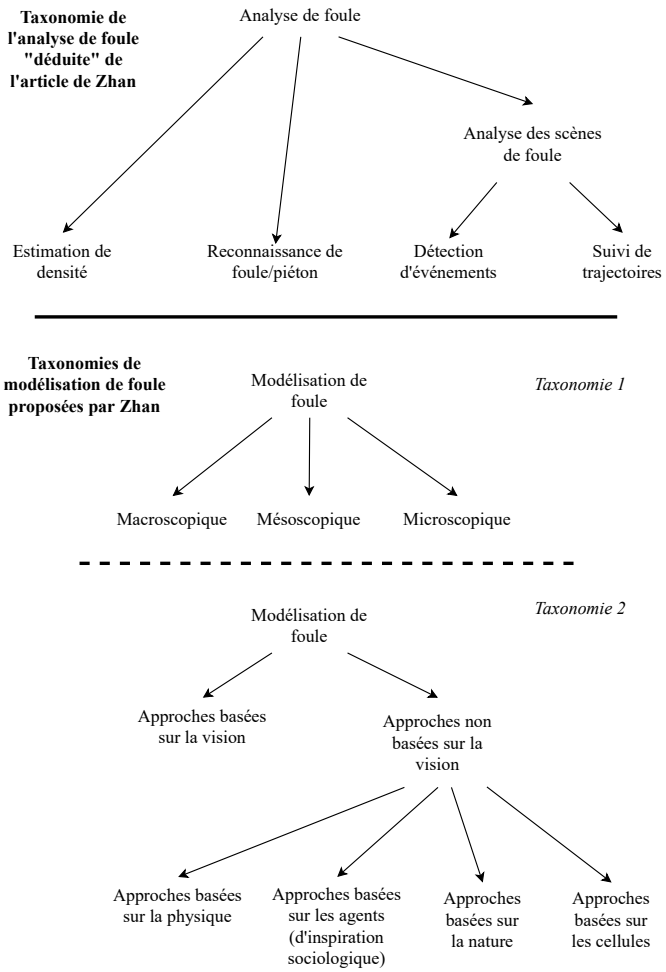


FIGURE 1.1 – Illustration de la taxonomie tirée de l'article d'état-de-l'art de (ZHAN et al., 2008)

2017) ne s'intéressent pas beaucoup à l'analyse des actions et des comportements humains au sein d'une foule. Cet article conclut que peu de recherches sont menées sur l'automatisation de l'identification des contextes environnants pouvant représenter un danger pour une foule en raison de la rareté des données. Une référence est faite à certains jeux de données populaires utilisés pour l'analyse de foule. Cependant, comme le soulignent (TRIPATHI, SINGH et VISHWAKARMA, 2018), malgré la prévalence des méthodes issues de l'apprentissage profond en vision par ordinateur, une infime partie des travaux mentionnés en analyse de foule est basée sur le Deep Learning.

(LAMBA et NAIN, 2017) commencent par souligner l'incapacité des méthodes traditionnelles à modéliser la dynamique des foules en raison de l'occlusion et des scènes congestionnées. Ils évoquent dans leur état-de-l'art les descripteurs (ou, autrement dit, les extracteurs d'indices visuels), souvent employés pour épauler des méthodes d'analyse des scènes de foule et les catégorisent en descripteurs spatio-temporels locaux basés sur le flux optique ou des descripteurs spatio-temporels basés sur les trajectoires ou les pistes (tracklets). Ils ont classé les travaux analysés en deux catégories : le comptage du nombre de piétons, le suivi des trajectoires des piétons, et l'analyse des comportements de foule. Nous pouvons visualiser ces catégories de travaux sur la figure 1.3.

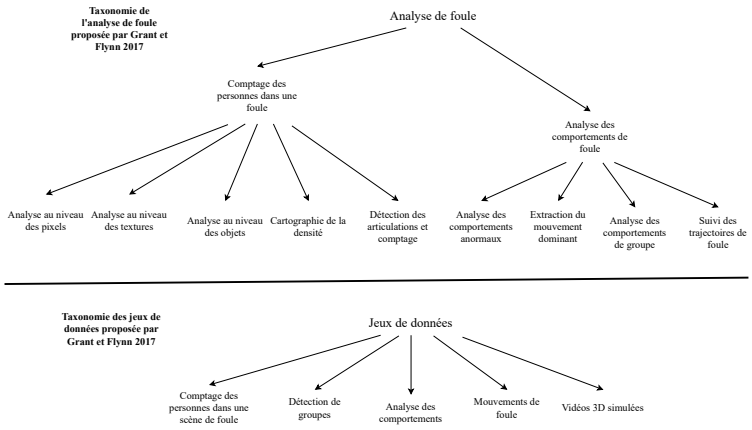


FIGURE 1.2 – Taxonomies proposées par (GRANT et FLYNN, 2017)

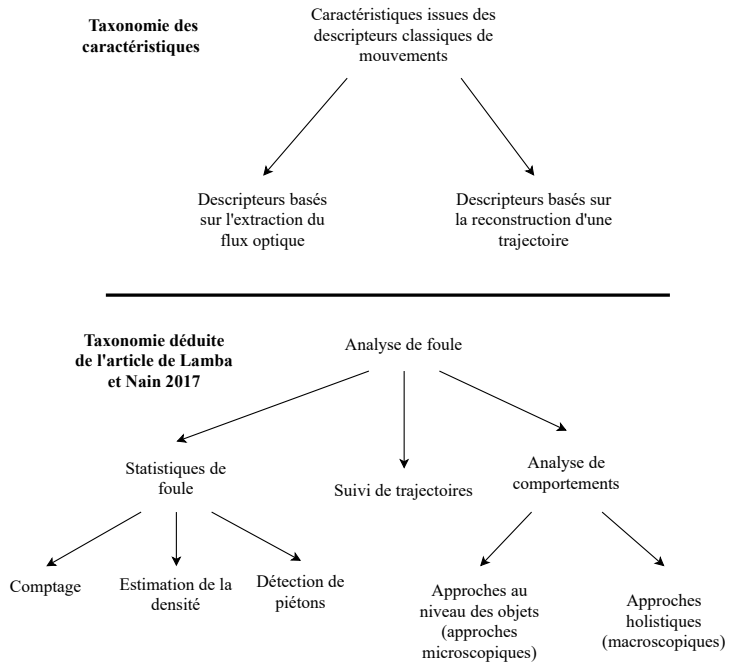


FIGURE 1.3 – Taxonomies proposées et déduites de l'article d'état-de-l'art de (LAMBA et NAIN, 2017)

(TRIPATHI, SINGH et VISHWAKARMA, 2018) regrettent que les articles d'état-de-l'art récents sur l'analyse de foule aient accordé peu d'importance aux travaux basés sur des méthodes issues du Deep Learning malgré l'omniprésence de ces méthodes dans toutes les branches de la vision par ordinateur. L'état-de-l'art comprend une analyse de plus d'une centaine de travaux employant des Réseaux de Neurones Convolutifs (Convolutional Neural Networks (CNN)). Devant la quantité de travaux collectés, (TRIPATHI, SINGH et VISHWAKARMA, 2018) ont trouvé un moyen de les séparer en quatre catégories selon la stratégie de paramétrage ou de mise en place des Réseaux de Neurones :

1. Travaux qui ont pour préoccupation majeure la variation du nombre de couches du Réseau de Neurones et du type de données envoyées en entrée ;
2. Travaux qui s'apppuient sur la mise en cascade d'une variété de Réseaux de Neurones Convolutifs et la fusion de leurs résultats de classification ;

3. Travaux qui utilisent les Réseaux de Neurones Convolutifs uniquement pour l'extraction des caractéristiques et s'appuient sur des méthodes issues de l'apprentissage automatique classique pour les tâches de classification ;
4. Travaux qui s'appuient sur l'incorporation de Réseaux de Neurones Convolutifs à d'autres architectures issues de l'apprentissage profond afin d'augmenter les performances globales.

Dans le sillage des articles précédents sur l'état-de-l'art, (Tripathi, Singh et Vishwakarma, 2018) ont divisé la littérature de l'analyse de foule en quatre domaines, comme nous pouvons l'observer sur la figure 1.4. (Tripathi, Singh et Vishwakarma, 2018) ont mis en exergue les défis auxquels sont confrontées les méthodes issues du Deep Learning. Parmi ces défis figurent le manque de données annotées et le besoin de puissantes infrastructures de calcul publiquement disponibles pour entraîner des modèles. Pour remédier à ces problèmes, ils suggèrent de recourir à l'apprentissage par transfert. Cependant, les auteurs ne donnent aucune piste sur les orientations que devrait prendre la recherche en analyse de foule.

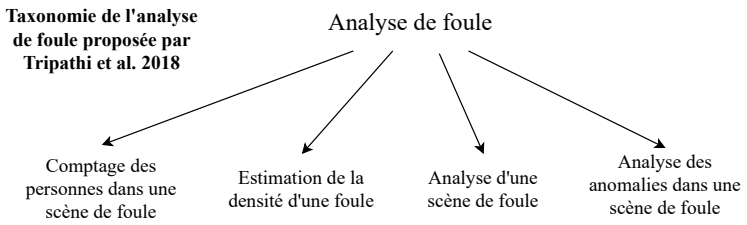


Figure 1.4 – Taxonomie proposée par (Tripathi, Singh et Vishwakarma, 2018)

Dans cette section, nous avons vu quatre articles d'état-de-l'art qui abordent l'analyse de foule tout en évoquant d'autres parties du domaine. Contrairement à (Zhan et al., 2008), les autres articles ont été publiés assez récemment et longtemps après l'émergence du Deep Learning en tant qu'approche incontournable en Vision par Ordinateur, et sa relative omniprésence en analyse de foule. Seuls (Tripathi, Singh et Vishwakarma, 2018) ont concentré leur état-de-l'art sur l'application des méthodes issues du Deep Learning à l'analyse de foule. Contrairement aux articles d'état-de-l'art précédents, (Tripathi, Singh et Vishwakarma, 2018) ne proposent pas une taxonomie totalement différente de l'analyse de foule. Toutefois, ils mettent en lumière toutes les utilisations qu'il est possible de faire de l'apprentissage profond dans l'analyse de foule. (Zhan et al., 2008), quant à eux, proposent trois taxonomies et permettent d'observer sous différents points de vue des facettes différentes de chaque travail qu'ils ont cité sur l'analyse de foule. Alors que (Grant et Flynn, 2017) proposent une taxonomie pour l'analyse de foule ainsi qu'une taxonomie pour catégoriser les jeux de données répertoriés, (Lamba et Nain, 2017) le font pour l'analyse de foule ainsi que pour les descripteurs essentiels aux différentes approches d'analyse.

1.2.2 Analyse des comportements de foule

Dans cette section, les articles d'état-de-l'art sont consacrés à l'analyse des comportements de foule, une branche importante de l'analyse de foule.

Contrairement à l'une des taxonomies de (Zhan et al., 2008), (Thida et al., 2013) ont divisé l'analyse de foule en deux grandes catégories : les approches microscopiques et les approches macroscopiques. Les auteurs montrent comment la foule est observée à travers ces deux prismes.

— D'un point de vue microscopique, nous adoptons des approches dites ascendantes (bottom-up). Ici, nous commençons par la détection des piétons, poursuivons avec le suivi de leurs trajectoires, et terminons par l'analyse d'actions au sein d'une foule. Les difficultés auxquelles ces approches sont confrontées sont : les occlusions, la complexité des événements qui se déroulent dans la scène du fait de la multiplication des interactions, les vibrations ou la mobilité de la caméra, etc.
— D'un point de vue macroscopique, nous adoptons des approches dites descendantes (top-down). Ces dernières considèrent une foule comme une seule entité. Les approches descendantes peuvent faire face à des obstacles lorsque la foule n'est pas structurée. Autrement dit, quand les individus se déplacent dans tous les sens, de manière anarchique, ce qui ne permet pas de dégager de tendances générales permettant de modéliser leurs déplacements. Dans ce contexte, il est difficile de trouver des motifs réguliers et de les caractériser par des modèles statistiques.

(Thida et al., 2013) soulignent qu'une grande partie des travaux menés en analyse des comportements de foule sont destinés à la détection d'événements, et en particulier à la détection d'événements anormaux. Cependant, la définition de l'anormalité n'est pas unanime parmi les chercheurs en analyse de foule. Parfois associée à la rareté, certaines définitions de l'anormalité la relient davantage à des événements jamais observés. Les approches macroscopiques reposent sur les propriétés holistiques d'une scène. La modélisation macroscopique emploie soit des indices visuels extraits du flux optique, soit d'autres indices spatio-temporels. En revanche, les approches microscopiques sont généralement basées sur des agents et se basent sur le suivi des trajectoires des entités atomiques en mouvement dans une scène. (Thida et al., 2013) mentionnent plusieurs approches de suivi de trajectoires ascendantes tirées de l'approche Particle Filter. Cette approche est principalement basée sur des indices visuels colorimétriques, mais les auteurs mentionnent également d'autres travaux où le Particle Filter est associé à d'autres indices. Ils évoquent la possibilité d'améliorer la qualité du suivi des trajectoires dans une foule en exploitant des informations contextuelles exogènes ou des informations endogènes telles que les interactions sociales. Enfin, pour faire face au défi que représente souvent les occlusions, les auteurs proposent de recourir à un système multi-caméras, et nous pouvons envisager ici d'employer des méthodes de fusion d'informations qui permettent de pallier les faiblesses d'un certain nombre de méthodes d'analyse de foule. (Thida et al., 2013) proposent une taxonomie pour catégoriser les méthodes d'analyse des comportements de foule que nous pouvons visualiser dans la figure 1.5. L'état-de-l'art se conclut sur une évocation des méthodes de détection d'événements dans des scènes de foule, où la détection d'anomalies est brièvement mentionnée. Cependant, nous regrettons que cette section n'ait pas été incluse dans la taxonomie proposée par les auteurs.

(Li et al., 2015) montrent comment la foule est perçue par la Dynamique des Foules, un domaine issu de la Mécanique des Fluides, et comment elle l'est par la Vision par Ordinateur. Alors que la Dynamique des Foules peut considérer la foule soit comme un fluide, soit comme un ensemble d'individus, la Vision par Ordinateur perçoit la foule à travers différentes échelles de grandeur. Du point de vue de la Dynamique des Foules :

— Si la foule est perçue comme un fluide, les représentations qui lui en sont faites sont tirées de la mécanique statistique et la thermodynamique,
— Si la foule est perçue comme un ensemble d'individus, les représentations qui en sont faites sont tirées des sciences sociales et sont basées sur des agents. Elles dérivent toutes du modèle de Force Sociale (Social Force Model (SFM))

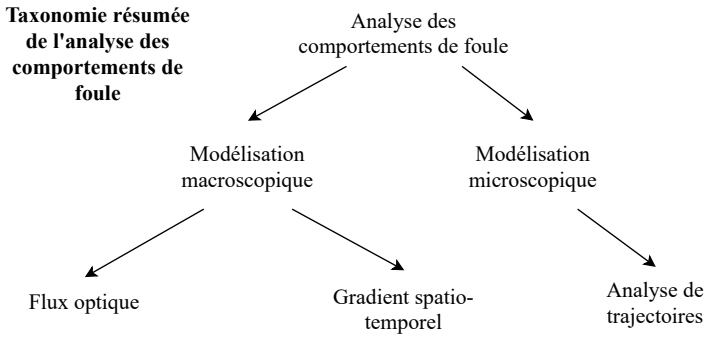


FIGURE 1.5 – Taxonomie trouvée dans l'article de (THIDA et al., 2013) et résumée

proposé par (HELBING et MOLNAR, 1995).
Du point de vue de la Vision par Ordinateur les différentes échelles d'observation de la foule sont principalement les échelles microscopiques et macroscopiques. Parfois, on trouve des approches qui adoptent une échelle intermédiaire, l'échelle mésoscopique, qui se revendique être un compromis entre les deux échelles extrêmes. Les auteurs soulignent l'importance de l'extraction des caractéristiques visuelles pour modéliser les mouvements de foule. Ils classent les caractéristiques en trois niveaux de hiérarchie selon leur degré d'expressivité :

1. Caractéristiques (ou indices) visuels de bas niveau, communément basées sur le flux optique.
2. Caractéristiques spatio-temporelles de niveau intermédiaire.
3. Descripteurs de trajectoires/tracklets qui sont des caractéristiques de haut niveau.

Comme observé à partir de la figure 1.6, les travaux étudiés sont classés en trois catégories :

1. La segmentation de motifs de mouvement dans une scène de foule,
2. La reconnaissance des comportements de foule,
3. La détection des anomalies dans une scène de foule.

Cet état-de-l'art n'évoque pas la détection des piétons et des groupes. Il n'aborde pas, non plus, l'analyse des comportements de groupe qui est pourtant perçue comme une belle illustration d'approches mésoscopiques en analyse de foule du point de vue de la vision par ordinateur.

1.2.3 L'analyse des comportements anormaux

L'analyse des comportements anormaux est un thème très étudié en analyse de foule.

Nous commençons cette section par l'état-de-l'art de (CHONG et TAY, 2015) qui abordent la détection d'anomalies dans les vidéos de manière générale sans se focaliser particulièrement sur l'analyse de foule. (CHONG et TAY, 2015) recommandent l'usage des méthodes issues de l'apprentissage profond pour détecter des événements anormaux dans les vidéos. La détection d'anomalies dans ce type de données est une tâche ardue. Ces difficultés peuvent provenir de la résolution des vidéos où une mauvaise qualité peut favoriser l'apparition d'artefacts ouvrant la porte à de mauvaises interprétations. Ces difficultés proviennent également de la variété des changements

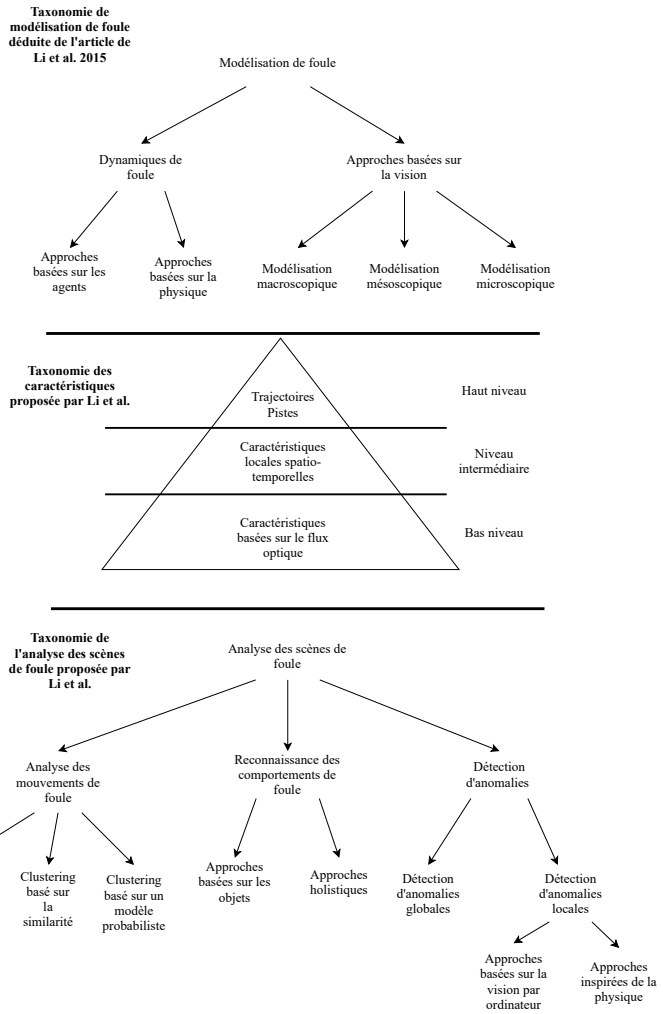


FIGURE 1.6 – Taxonomies trouvées et déduites de l'article de (LI et al., 2015)

soudains et imprévus qui surviennent dans une vidéo tels qu'une vibration de la caméra ou des variations contextuelles extérieures aux mouvements humains. Traditionnellement, la détection d'anomalies se décline en quatre étapes : le pré-traitement de la vidéo, l'extraction des caractéristiques ou des indices visuels via des descripteurs, la modélisation des mouvements de foule, et la classification ou le regroupement de ces mouvements via des méthodes issues de l'apprentissage automatique classique. Les caractéristiques extraites sont souvent basées sur le flux optique ou les trajectoires. Des caractéristiques utilisées pour un type de scènes, avec un environnement particulier, doivent souvent être reparamétrées pour s'adapter à de nouveaux environnements. De nos jours, cette tâche harassante de reparamétrage peut être déléguée à des méthodes issues de l'apprentissage profond. Il est même possible d'alléger encore plus cette tâche en ayant recours à l'apprentissage par transfert. Les auteurs déconseillent d'envoyer directement des vidéos à des méthodes de classification en raison de la variation des motifs d'actions, des changements contextuels, et du désordre apparaîssant dans une scène de foule. Ils recommandent de trouver d'abord une bonne représentation pour les clips vidéo. Pour ce faire, ils donnent un aperçu des différents descripteurs classiques avant d'évoquer des extracteurs de caractéristiques issues de l'apprentissage profond. Ils classent ces derniers en trois catégories :

1. Les Machines de Boltzmann Restreintes conditionnelles et les Réseaux de croyance profonds spatio-temporels,
2. Analyse en Composantes Indépendantes et ses variantes,
3. Analyse profonde des caractéristiques lentes et les modèles à portes.

Ils comparent les méthodes d'extraction de caractéristiques issues de l'apprentissage profond à des méthodes classiques sur les jeux de données KTH et Hollywood2. Ces méthodes classiques sont : HOG3D (Histogramme des dégradés orientés 3D) et HOG/HOF (Histogramme des gradients orientés/Histogramme de flux optique). Nous pouvons visualiser les classes des descripteurs explorés dans cet état-de-l'art dans la figure 1.7. Hormis KTH et Hollywood2, l'état-de-l'art ne donne pas d'aperçu sur les jeux de données qui sont utilisés pour la détection d'anomalies dans des vidéos de scènes de foule.

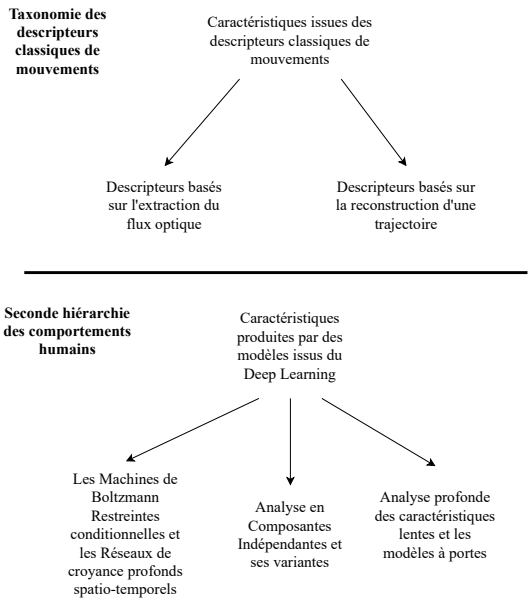


FIGURE 1.7 – Taxonomies proposées par (CHONG et TAY, 2015)

(KIRAN, THOMAS et PARAKKAL, 2018) parlent principalement de la détection et la prédiction d'anomalies dans les vidéos dans le cadre d'un apprentissage non supervisé. Ils mettent en exergue la rareté des données annotées et les problèmes qui en découlent malgré la disponibilité de données vidéo brutes. Ils définissent l'anomalie comme étant l'apparition d'objets non-identifiés et l'occurrence d'événements peu fréquents. L'état-de-l'art mentionne six jeux de données qui sont fréquemment utilisés : UCSD Anomaly Detection, CUHK Crowd, et UMN Social Force que nous décrivons plus loin dans la section 1.5.2. Il nous introduit également à des jeux de données moins utilisés comme les jeux de données Subway Entry/Exit, le Train Station dataset, le Queen Mary University of London U-turn, et le LV dataset créé par (LEYVA, SANCHEZ et LI, 2017). Partant du principe que la normalité est définie par un arrière-plan statique, une foule au comportement ordinaire, et l'absence de changement brusque de trajectoire, les jeux de données mentionnés comprennent des vidéos où l'arrière-plan est toujours fixe. L'état-de-l'art examine les méthodes d'apprentissage profond utilisées pour la détection non supervisée et semi-supervisée d'anomalies dans les vidéos. (KIRAN, THOMAS et PARAKKAL, 2018) classent, dans un premier étage, ces méthodes en fonction de l'approche d'apprentissage suivie : reconstruction, prédiction, ou génération. Dans un second étage, ces méthodes sont classées encore plus précisément

selon le type d'architecture de réseaux utilisée. Cette catégorisation est illustrée dans la figure 1.8 :

— Les modèles de reconstruction comprennent l'analyse en composantes principales (PCA), les auto-encodeurs, les auto-encodeurs convolutifs, les auto-encodeurs à contraction et d'autres modèles profonds comme les SDAE (Stacked De-noising AutoEncoders) et les DBN (Deep Belief Nets).

— Les modèles prédictifs comprennent le modèle LSTM composite qui réalise la reconstruction et la prédiction, le LSTM convolutif qui est également un LSTM composite, 3D-AutoEncoder and Predictor, Slow Feature Analysis (SFA) qui est calculé en utilisant le batch PCA itéré deux fois.

— Les modèles génératifs profonds comprennent des AutoEncoders Variationnels (VAE), des Réseaux Adversaires Génératifs (GAN), des AutoEncoders Adversaires (AAE).

(Kiran, Thomas et Parakkal, 2018) ont réalisé un comparatif entre des méthodes analysées sur les jeux de données CUHK Crowd et UCSD Anomaly Detection. Pour évaluer leurs méthodes, ils ont employé les mesures Précision-Rappel (PR) et la courbe Receiver-Operating-Characteristic (ROC).

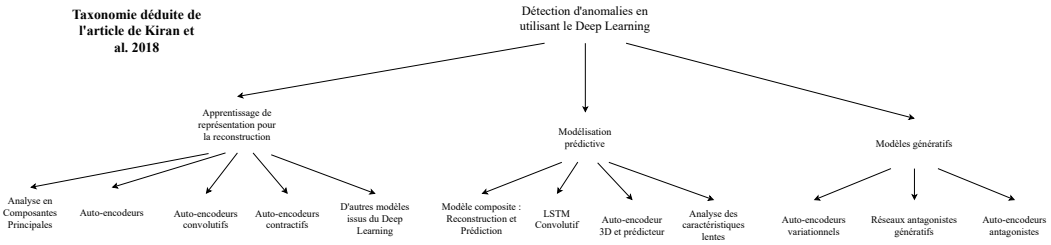


Figure 1.8 – Taxonomie déduite de l'article de (Kiran, Thomas et Parakkal, 2018)

1.2.4 Le suivi de trajectoires des piétons et des foules

Le suivi des trajectoires est un thème important en analyse des comportements de foule. Dans cette section, nous mentionnons seulement l'état-de-l'art de (Walia et Kapoor, 2016).

(Walia et Kapoor, 2016) comparent dans leur état-de-l'art différentes approches de suivi de trajectoires des objets. Les méthodes qu'ils étudient se regroupent dans deux grandes catégories : des méthodes uni-indiciaires (Single-cue) et des méthodes multi-incidaires.

— Les méthodes uni-indiciaires (Single-cue) uni-modales n'extraient qu'un seul type d'indices ou de caractéristiques à partir d'un seul type de capteur.

— Les méthodes multi-indiciaires (Multi-cue) uni-modales (resp. multi-modales) extraient plusieurs types d'indices à partir d'un seul capteur (resp. à partir de plusieurs capteurs différents).

(Walia et Kapoor, 2016) commencent par mettre en lumière les inconvénients des méthodes uni-indiciaires de suivi de trajectoires. Ces méthodes éprouvent des difficultés manifestes dans les conditions réelles. Leurs alternatives sont des méthodes de suivi de trajectoires de plusieurs types d'indices (Multi-cue Tracking). (Walia et Kapoor, 2016) classent ces derniers en fonction de la provenance de leurs caractéristiques qui sont des capteurs mono-modaux ou multi-modaux. Cette catégorisation est illustrée dans la figure 1.9.

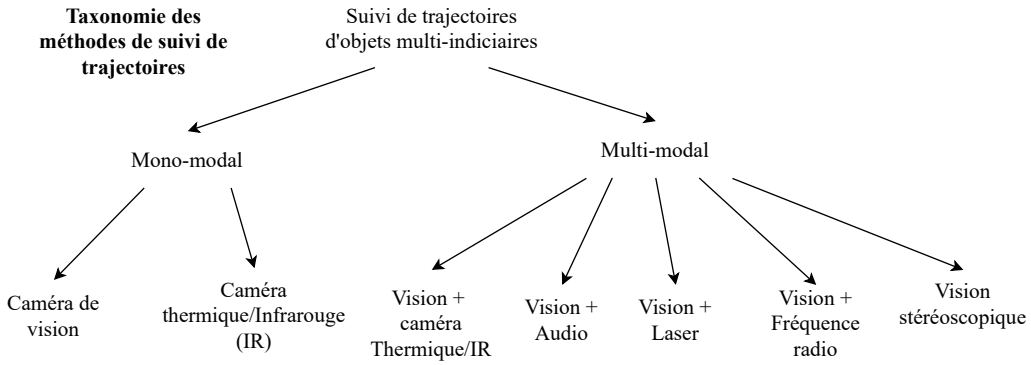


FIGURE 1.9 – Traduction de la taxonomie du suivi de trajectoires d'objets multi-indiciaires proposée par (WALIA et KAPOOR, 2016)

— Les méthodes uni-modales multi-indiciaires extraient plusieurs types d'indices, à partir d'un seul capteur, pour réaliser le suivi de trajectoires.
— Les méthodes multi-modales multi-indiciaires extraient plusieurs types d'indices à partir de capteurs différents. Les caractéristiques extraites sont par la suite combinées à l'aide de méthodes de fusion d'informations pour réaliser le suivi de trajectoires des objets.

Dans une grande partie des travaux analysés par (WALIA et KAPOOR, 2016), l'extraction des caractéristiques se fait au moins à partir d'un capteur visuel (une caméra). Ces indices visuels sont parfois combinés avec d'autres indices provenant de capteurs non-visuels tels que des caméra IR, des capteurs thermiques, des capteurs audio, etc. Les méthodes employées pour le suivi de trajectoires sont soit déterministes, soit stochastiques. (WALIA et KAPOOR, 2016) dressent une liste de combinaisons possibles de capteurs :

— Capteur de vision associé à un capteur thermique/InfraRouge. Cette combinaison est requise lorsque le suivi des trajectoires d'objets est effectué pendant la nuit. Cependant, la fusion de données et la calibration du système multi-capteurs sont des tâches ardues.
— Combinaison de capteurs visuels et audio. Cette combinaison est privilégiée pour récolter le maximum d'informations lors des conférences et des réunions.
— Combinaison de capteurs visuels et laser. Le recours à ce type de combinaisons se trouve dans des applications de vidéosurveillance. Le scanner laser présente les avantages des faibles besoins en termes de ressources de calcul, l'insensibilité aux changements environnementaux, et la projection facile des données laser aux coordonnées rectangulaires. Cependant, selon les travaux de (CUI et al., 2008) et (SONG et al., 2013), le scanner laser peut faillir devant des situations de congestion et d'occlusion.
— La combinaison de capteurs radio et vision. Les informations radiales fournies par les radars permettent une identification plus précise des objets. Par ailleurs, les systèmes radars sont de moins en moins coûteux.
— La vision stéréo a également été étudiée pour sa capacité à capturer les différents mouvements d'un objet et sa résilience face aux variations lumineuses.

Dans l'article, les auteurs présentent une série de jeux de données servant à évaluer les travaux précédemment introduits. Ils mentionnent également les mesures de performance permettant de mesurer la robustesse et l'efficacité de plusieurs méthodes de suivi de trajectoires. Malgré l'utilisation répandue du Deep Learning (DL) pour le suivi des trajectoires des objets, (WALIA et KAPOOR, 2016) ne mentionnent aucun

travail basé sur l'apprentissage profond.

1.2.5 Analyse des comportements de groupe

L'analyse des comportements de groupe est un domaine de l'analyse des comportements de foule. Nous mettons en évidence les tendances récentes de l'analyse des comportements de groupe à travers l'état-de-l'art de (Borja-Borja, Saval-Calvo et Azorin-Lopez, 2017).

(Borja-Borja, Saval-Calvo et Azorin-Lopez, 2017) associent la nature d'une action humaine au nombre d'individus qui l'exécutent. Ils mentionnent la hiérarchisation d'actions proposée dans d'autres articles (Azorin-Lopez et al., 2015 ; Chaaraoui, Climent-Pérez et Flórez-Revuelta, 2012) qui classent les actions humaines en fonction de leur durée d'exécution. Ces deux classifications hiérarchiques sont représentées sous la forme de pyramides dans la figure 1.10.

— La première classification décompose les comportements en quatre niveaux selon leur durée et le nombre de personnes qui y sont impliquées : gestes, actions, interactions, et activités de groupe.
— La deuxième classification divise les comportements en quatre niveaux en fonction de leur durée : mouvements, actions, activités, et comportements.

Dans la deuxième partie de l'état-de-l'art, les auteurs décrivent les jeux de données utilisés pour la reconnaissance d'actions de groupe tels que : BEHAVE, CAVIAR, CVBASE, ETISEO, ETH, UHD, HMDB, SportsVU, PETS, ViF. À l'aide du descripteur Group Activity Descriptor Vector (GADV), proposé par (Azorin-Lopez et al., 2016), les auteurs classent les comportements de groupe en prenant en compte le nombre d'individus impliqués dans chacun d'eux. Ce descripteur est obtenu après l'extraction des caractéristiques de trajectoires à l'aide des Réseaux de Neurones. Par la suite, les auteurs présentent les caractéristiques employées pour plusieurs tâches liées à la reconnaissance des comportements de groupe telles que la détection d'anomalie lorsque cette anomalie est causée par un petit groupe au sein d'une foule, la distinction entre un comportement normal et un comportement anormal basée sur l'hypothèse qu'un comportement normal dure souvent plus longtemps, et l'estimation des caractéristiques internes d'un groupe telles que la distance entre ses membres et la vitesse de chaque membre. Bien que (Borja-Borja, Saval-Calvo et Azorin-Lopez, 2017) offrent un aperçu de ce qui est fait en analyse des comportements de groupe, aucune mention n'est faite sur les tâches de détection de groupes.

1.2.6 Conclusion et discussion

Pour résumer les taxonomies proposées dans les articles d'état-de-l'art précédents, nous avons proposé une taxonomie synthétique illustrée dans la figure 1.11. À travers ce schéma, qui s'inspire principalement des conclusions des articles d'état-de-l'art précédents, nous résumons notre perception de l'état des connaissances en analyse de foule. Comme indiqué précédemment, le domaine du calcul des statistiques de foule est divisé en comptage du nombre de personnes dans une foule et l'estimation de la densité d'une foule. Même si ces deux sujets sont liés l'un à l'autre, l'estimation de la densité d'une foule est fréquemment liée à la gestion des foules et peut être utilisée par les forces de l'ordre pour prévoir quand un lieu est congestionné et peut représenter un danger pour la foule qui s'y trouve, tandis que le comptage du nombre de personnes dans une foule peut être utile pour les décisionnaires qui doivent mesurer leur audience à des fins statistiques. L'analyse des comportements de foule peut être divisée en trois grands domaines :

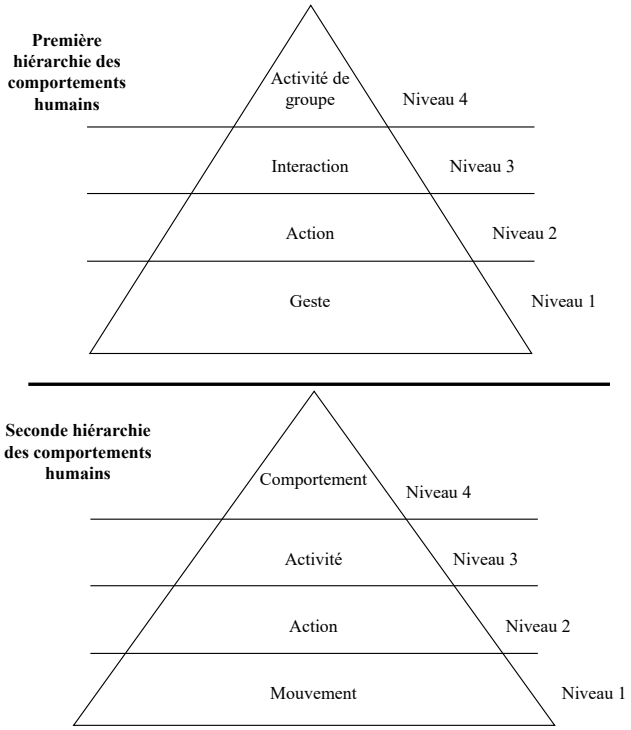


FIGURE 1.10 – Hiérarchies des comportements humains trouvées dans l'article de (BORJA-BORJA, SAVAL-CALVO et AZORIN-LOPEZ, 2017). La pyramide du haut provient de l'article de (VISHWAKARMA et AGRAWAL, 2013), et la pyramide du bas provient de l'article de (CHAARAOUI, CLIMENT-PÉREZ et FLÓREZ-REVUELTA, 2012)

— La reconnaissance/classification des comportements de foule, un sujet légèrement lié à la reconnaissance d'actions dans des scènes individuelles, comme nous le verrons par la suite dans la section 1.4.2.1,
— L'analyse des trajectoires, qui comprend le suivi et la prédiction de trajectoires,
— L'analyse des comportements de groupe, qui nécessite souvent une détection préalable de groupes suivie d'une reconnaissance des actions des groupes dans une scène de foule.

Dans notre taxonomie, nous ne mentionnons pas directement la division de l'analyse de foule en approches microscopiques (lagrangiennes), mésoscopiques, et macroscopiques (eulériennes) évoquées par les articles d'état-de-l'art de (THIDA et al., 2013 ; ALLAIN, COURTY et CORPETTI, 2012 ; ZHAN et al., 2008 ; WANG, CHENG et WANG, 2018). D'une part, cette division fait écho aux branches de la taxonomie que nous proposons pour l'analyse des comportements de foule :

— L'analyse des trajectoires est fréquemment associée à des approches microscopiques basées sur la détection préalable des piétons,
— La reconnaissance des comportements de foule découle d'une perception holistique d'une foule,
— L'analyse des comportements de groupe couvre les approches mésoscopiques qui sont considérées comme approches intermédiaires.

D'autre part, nous observons que le calcul des statistiques de foule peut également obéir à la dichotomie des approches microscopiques/macroscopiques. Si nous considérons le comptage du nombre de personnes dans une foule comme une approche microscopique pour les statistiques de foule, parce que souvent basée sur la détection de piétons, nous pouvons considérer l'estimation de la densité comme une approche

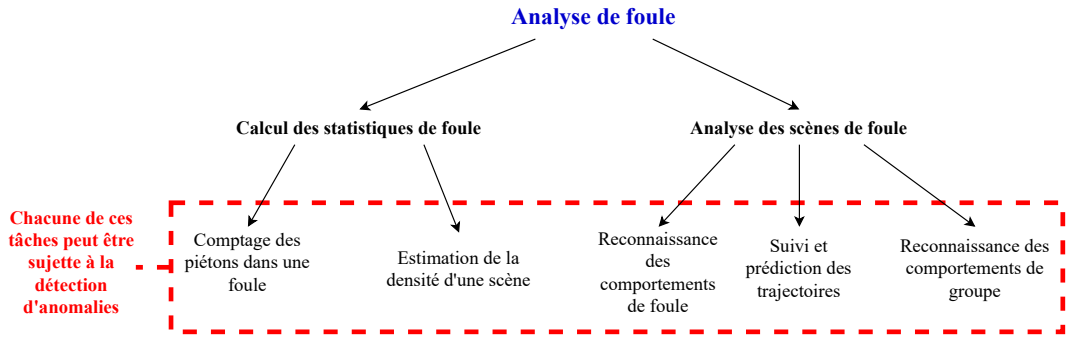


FIGURE 1.11 – Taxonomie proposée pour l'analyse de foule

macroscopique, car elle perçoit la foule comme une seule entité.

Dans la taxonomie que nous proposons, nous préférons réserver une catégorie à la détection d'anomalies et la considérer comme une branche de l'analyse de foule. Nous considérons que chaque domaine de l'analyse de foule peut inclure des travaux liés à la détection d'anomalies.

Cette taxonomie nous sert à structurer la section 1.4 de ce chapitre. Les différentes catégories illustrées dans la taxonomie de la figure 1.11 constituent les différents domaines de recherche abordés dans la section 1.4 et qui sont le calcul des statistiques de foule, divisée en comptage des personnes et en estimation de la densité, et l'analyse des comportements de foule, divisée en reconnaissance d'actions, suivi et prédiction des trajectoires, et l'analyse des comportements de groupe. Comme chacun de ces domaines peut être sujet à l'apparition d'une anomalie, la section 1.4.3 est spécialement réservée aux travaux liés à la détection d'anomalies.

Depuis 2012, suite aux prouesses des Réseaux de Neurones Convolutifs dans la classification d'images, illustrées par le modèle AlexNet (KRIZHEVSKY, SUTSKEVER et HINTON, 2012), l'usage des méthodes issues de l'apprentissage profond en Vision par Ordinateur a décuplé. De plus en plus d'approches adoptent le Deep Learning dans l'analyse de foule comme l'illustre parfaitement la variété d'approches analysées et classées par (TRIPATHI, SINGH et VISHWAKARMA, 2018). Cependant, en raison du manque de diversité et de la richesse des données annotées, en particulier pour l'analyse des comportements de groupe et l'analyse des comportements de foule, la recherche dans ces deux thèmes de l'analyse de foule est encore à ses débuts.

1.3 La détection de piétons et de groupes

Avant de nous plonger dans l'analyse des mouvements ou des comportements de foule, nous allons parler des travaux portant sur la détection des composantes d'une foule. Ces éléments sont représentés par des unités atomistes, les piétons. Ces derniers peuvent également former des granulats représentés par les groupes de piétons.

Avant l'avènement des méthodes issues de l'apprentissage profond dans l'analyse de foule, de nombreux travaux utilisaient des méthodes traditionnelles pour extraire les caractéristiques de foule et les intégraient à des approches d'apprentissage automatique pour détecter des piétons et des groupes. Dans cette section, nous commençons par mentionner un travail moins récent, mais non moins souvent cité, qui utilise des méthodes traditionnelles pour la détection des piétons et des groupes. Ensuite, nous passons en revue et comparons les travaux qui ont été réalisés plus récemment dans ces deux domaines.

(Ge, Collins et Ruback, 2012) proposent une approche de clustering hiérarchique agglomérative ascendante pour détecter de petits groupes d'individus. Leur travail cible deux types de scènes de foule. Scènes filmées à partir d'une caméra surélevée où les individus sont minuscules et scènes filmées à partir du niveau d'une caméra de vidéo-protection avec une résolution plus élevée, où les individus peuvent être clairement identifiés. Pour détecter les piétons à partir de caméras en hauteur, les auteurs utilisent la méthode RJMCMC (Reversible Jump Markov Chain Monte Carlo). Pour les vidéos de plus haute résolution, ils utilisent une version modifiée du détecteur HoG (Dalal et Triggs, 2005). Après l'opération de détection, ils obtiennent des pistes (tracklets) en utilisant le filtre de particules Sampling Importance Resampling (SIR) (Doucet et Johansen, 2009). Les pistes (tracklets) sont assemblées en ensembles de tracklets. La fenêtre glissante est utilisée pour trouver des tracklets qui se chevauchent et qui constituent des candidates potentielles pour des trajectoires plus longues, ce qui permet aux auteurs de produire des ensembles de trajectoires. Les nouvelles tracklets sont assignées à la bonne trajectoire en utilisant l'algorithme Hongrois (Kuhn, 1955). Si dans une trajectoire les auteurs font face à des emplacements manqués, ils les déduisent par interpolation linéaire. À partir d'un ensemble de trajectoires, ils combinent la stratégie de la fenêtre glissante au regroupement hiérarchique pour trouver de petits groupes de personnes. Ensuite, pour mesurer la proximité entre petits groupes, ils utilisent la distance symétrique de Hausdorff (une distance déjà utilisée pour l'analyse de trajectoire (Wang, Tieu et Grimson, 2006)). Les auteurs ont entrepris leurs expériences sur un jeu de données, qu'ils ont eux-mêmes conçu, constitué de 5 séquences vidéo : SU1, SU2, Artefest, Stadium1, Stadium2. Dans leurs expériences, ils surpassent, en termes de précision de détection des groupes, les méthodes de Corner Clustering (Rabaud et Belongie, 2006) et une autre catégorie de méthodes : (Brostow et Cipolla, 2006) et (Sugimura et al., 2009).

1.3.1 La détection de piétons

Dans les conditions réelles fréquemment observées par les systèmes de vidéo-protection, la détection des piétons n'est pas une tâche aisée. Les scènes sont souvent encombrées et les détecteurs font face à plusieurs types d'occlusions. Alors que les méthodes traditionnelles ne parviennent souvent pas à détecter les piétons dans des situations scénaristiques complexes, les modèles issus du Deep Learning obtiennent souvent de très bons résultats. Lorsque nous développons un détecteur de piétons, le but ultime est de créer un détecteur rapide, tout en étant précis, qui peut fonctionner sur des systèmes avec une puissance de calcul abordable (Angelova et al., 2015).

(Li, Wu et Zhang, 2016) entraînent un Region Proposal Multi-Layered Convolutional Neural Network (RCNN), sur la détection des piétons. Au lieu de la fenêtre glissante classique (Sliding Window), les auteurs utilisent l'algorithme Edge Boxes (Zitnick et Dollár, 2014) pour les propositions de régions. Après cela, ils utilisent une Machine à Vecteurs de Support (Support Vector Machine (SVM)) pour classer les caractéristiques obtenues afin d'identifier les piétons. Les auteurs ont entraîné et testé le RCNN sur le jeu de données de reconnaissance de piétons de l'INRIA (INRIA Person Dataset) (Dalal et Triggs, 2005). Pendant la procédure d'apprentissage, ils ont affiné le modèle AlexNet (Krizhevsky, Sutskever et Hinton, 2012) qui a été pré-entraîné sur le jeu de données ILSVRC2012. Pour éviter un surapprentissage, les auteurs ont augmenté le jeu de données INRIA en lui ajoutant des données similaires sur des piétons issues des travaux de (Wang et al., 2007). Au cours de la

phase de test, ils ont évalué un modèle entraîné sur le jeu de données INRIA Person, non augmenté, et un modèle basé sur la recherche sélective (Selective Search) (Uijlings et al., 2013). Le modèle des auteurs dépasse la performance des autres modèles en termes de faux positifs par image (FPPI) et de taux d'erreur (Miss Rate (MR)). Le modèle obtient de meilleurs résultats que les caractéristiques extraites en utilisant l'Histogramme des Gradients (HOG) et les techniques de Viola Jones. En d'autres termes, le modèle atteint 23% de taux d'erreur, ce qui est 23% plus élevé que la méthode HOG. La méthode développée et ses variantes ont été comparées aux méthodes traditionnelles. Cependant, malgré l'omniprésence des méthodes issues de l'apprentissage profond dans ce domaine, nous regrettons l'absence de comparaison entre la méthode des auteurs avec d'autres modèles issus du Deep Learning.

(Tian et al., 2015) proposent DeepParts, un détecteur de piétons basé sur la reconnaissance des membres d'un piéton et qui se présente comme un outil efficace face aux occlusions. L'algorithme repose sur l'apprentissage d'une vaste palette des composants d'un corps humains à différentes échelles et orientations, ce qui permet par la suite de reconnaître la présence d'un piéton à partir de l'occurrence d'un de ses membres. Les propriétés suivantes caractérisent DeepParts : la capacité à être entraîné sur des données faiblement annotées ; traitement des propositions positives de faible *Intersection over Union (IoU)* qui s'éloignent de la vérité de terrain ; chaque détecteur de membre peut détecter un piéton en ne visualisant qu'une partie d'une proposition. L'apprentissage et l'évaluation de DeepParts ont eu lieu sur le jeu de données Caltech (Dollar et al., 2012). Au cours des expériences qui ont été menées sur ce jeu de données, un modèle global de DeepParts a atteint un score de taux d'échec de 11.89% (Tian et al., 2015). DeepParts a également été évalué sur le jeu de données KITTI (Geiger et al., 2013), et bien qu'il ait été entraîné sur Caltech, il a obtenu un score de précision moyenne (AP) de 58.67%. Ce score est toutefois à nuancer avec le fait que KITTI est un jeu de données moins dense, et DeepParts y est dépassé en termes de précision moyenne (AP) par l'algorithme Regionlets (Wang et al., 2013). En guise de perspectives, les auteurs proposent d'inclure DeepParts dans une pile en cascade contenant d'autres détecteurs afin d'augmenter les performances de détection. Ils suggèrent également une compression du modèle, en incluant tous les détecteurs de membres d'un piéton dans un Réseau de Neurones Convolutifs unique.

Inspirés par les succès apportés par le développement de détecteurs précis basés sur le Deep Learning (Luo et al., 2014) et par la vitesse remarquable des Very Fast Cascades (Benenson et al., 2012) lors du traitement des patchs d'image, (Angelova et al., 2015) ont construit DeepCascade, une combinaison des Very Fast Cascades avec des Réseaux de Neurones Profonds. Plus précisément, ils n'ont pas complètement copié les cascades de (Benenson et al., 2012), mais ont modifié leur composition pour ne conserver que 10% de leurs étages. Pour obtenir un bon modèle, ils ont utilisé un Réseau de Neurones Profond pré-entrainé sur le jeu de données ImageNet. Par ailleurs, ils ont utilisé une méthode d'augmentation des données. De plus, pour sélectionner le meilleur modèle, ils ont entraîné leur algorithme sur trois jeux de données produisant trois modèles différents : un modèle issu du jeu de données Caltech Pedestrian (Dollar et al., 2012), un modèle entraîné sur un jeu de données collecté indépendamment et un jeu de données supplémentaire contenant Caltech, ETH (Pellegrini et al., 2009) et Daimler (Munder et Gavrila, 2006). Le modèle le plus performant est issu de l'apprentissage sur le jeu de données supplémentaire. À l'évaluation, ils ont obtenu de très bons résultats en termes de précision. Ils ont atteint le score moyen du taux d'échecs de 26.2% sur le jeu de données Caltech. Pour améliorer leur modèle, ils proposent d'inclure des informations du mouvement à partir d'images et à augmenter la profondeur de DeepCascade en ajoutant des réseaux profonds supplémentaires et

en évaluant le compromis efficacité-précision.

Nous avons vu récemment que bien que Faster RCNN réalise d'excellentes performances pour la détection d'objets (Ren et al., 2015), ce n'est pas un détecteur de piétons parfait (Zhang et al., 2016a). Lorsque nous choisissons un détecteur de piétons, nous devons faire un compromis difficile entre vitesse et précision. Par exemple, (Liu et al., 2016b) font partie de ceux qui plaident pour ce genre de compromis tout en tentant de réduire au maximum ses effets secondaires.

1.3.2 Détection de groupes

Deux types d'approches peuvent être trouvées dans la littérature sur la détection de groupe : les approches descendantes, qui partent d'une scène de foule, puis la segmentent en petits groupes (Wang, Cheng et Wang, 2018 ; Li et al., 2017 ; Chen, Wang et Li, 2017a ; Chen, Wang et Li, 2017b). Les approches ascendantes, qui commencent par détecter les piétons, puis les regroupent en groupes d'individus (Shao, Dong et Zhao, 2018 ; Yuan, Lu et Wang, 2017 ; Vascon et Bazzani, 2017).

(Shao, Dong et Zhao, 2018) proposent une approche ascendante en temps réel pour la détection de petits groupes d'individus qui fonctionne sur des scènes de foule ayant une densité faible ou moyenne. L'élément clé de leur méthode est la combinaison du modèle de Force Sociale (Social Force Model (SFM)) (Helbing et Molnar, 1995) employée pour la prédiction des objectifs des individus avec l'algorithme de filtrage cohérent (Coherent Filetering (CF)) tourné vers les objectifs des piétons pour la détection de groupes d'individus. Ils commencent par extraire les trajectoires selon deux méthodes (Dollár et al., 2014 ; Milan, Roth et Schindler, 2014). Après cela, ils construisent un modèle d'évitement de collision basé sur SFM (Karamouzas et al., 2009). Ensuite, ils appliquent l'algorithme K Nearest Neighbors (KNN) pour trouver les K voisins d'un piéton. Enfin, ils utilisent le filtrage cohérent (Coherent Filtering (CF)) basé sur les objectifs pour regrouper les piétons partageant le même objectif. Ils comparent leur méthode à celle de (Solera, Calderara et Cucchiara, 2016), et ils obtiennent, la plupart du temps, une meilleure précision avec des scores de rappel fréquemment similaires en un temps de calcul très court. Ils ont testé leur méthode sur les jeux de données student003, GVEII, et MPT-20X100.

(Wang, Cheng et Wang, 2018) proposent une approche descendante pour détecter des groupes de piétons. Elle consiste en une segmentation de la foule. La méthode se décline en deux étapes : la détection des points d'intérêt à l'aide de la méthode de (Tomasi et Kanade, 1991) et le calcul de la valeur de ses attributs. La foule est segmentée en plusieurs groupes en se basant sur les attributs calculés. Les auteurs ont évalué leur méthode sur le jeu de données CUHK Crowd dans diverses situations : des foules structurées, non structurées denses, ou non structurées et moins denses. Les auteurs comparent leur méthode à celle de (Shao, Change Loy et Wang, 2014), où ils la supplantent en termes de temps de calcul. Cependant, ils n'ont pas comparé leur méthode à d'autres détecteurs sur le jeu de données CUHK Crowd.

(Voon et al., 2019) proposent la méthode Collective Interactions Filtering (CIF), une approche de clustering basée sur l'algorithme Espérance-Maximisation (EM) qui regroupe des trajectoires pour détecter des groupes d'individus. Tout d'abord, ils commencent par extraire les tracklets (ou bouts de trajectoires) des piétons à l'aide du tracker Kanade-Lucas-Tomasi (KLT) (Tomasi et Kanade, 1991). Les tracklets sont utilisées comme entrées dans CIF pour trouver les clusters de tracklets. Au sein d'un cluster, ils sélectionnent comme personne clé (centre d'un cluster) le piéton qui a la plus longue durée et dont la trajectoire a un faible écart-type. Deuxièmement, ils calculent le degré de connexion entre chaque personne avec les personnes clés. Pour ce

Référence	Date	Axe de recherche	Datasets	?DL	Code de source
(Ge, Collins et Ruback, 2012)	2012	Détection de piétons & de groupes	Ad hoc	×	Non disponible
(Li, Wu et Zhang, 2016)	2016	Détection de piétons	INRIA Person	✓	Non disponible
(Tian et al., 2015)	2015	Détection de piétons	Caltech, KITTI	✓	Non disponible
(Angelova et al., 2015)	2015	Détection de piétons	Caltech	✓	Disponible
(Shao, Dong et Zhao, 2018)	2018	Détection de groupes	Student003, GVEII, MPT-20X100	×	Unavailable
(Wang, Cheng et Wang, 2018)	2018	Détection de groupes	CUHK Crowd	×	Non disponible
(Voon et al., 2019)	2019	Détection de groupes	CUHK Crowd	×	Non disponible

TABLE 1.1 – Présentation synthétique des travaux vus en détection de piétons et de groupes. **La colonne " ?DL" :** DL représentent les initiales de Deep Learning, cette colonne renseigne sur l'usage (✓) ou non (×) de techniques liées à l'apprentissage profond dans l'article étudié

faire, des distances telles que la Distance Connectivity (DC), Occurrence Connectivity (OC), Speed Correlation (SC), sont calculées, de la première à la dernière image, entre la personne clé et les autres personnes, puis sont stockées dans une matrice de contingence. Après cela, ils utilisent l'algorithme EM pour trouver les personnes proches des personnes clés. Un seuil est utilisé pour évaluer la méthode sur des scènes ayant différents niveaux de densités, et contenant des foules aux diverses structures et ayant des degrés d'occlusion différents. La méthode est évaluée sur le jeu de données CUHK Crowd. Comparée à la méthode Collective Transition (SHAO, LOY et WANG, 2017) et au Coherent Filtering (ZHOU, TANG et WANG, 2012), leur méthode obtient de meilleurs résultats en termes d'Information Mutuelle Normalisée (NMI) et d'Indice de Rand (RI).

Bien qu'il existe une quantité considérable de travaux dédiés à la détection de groupe, le sujet demeure peu étudié dans le domaine de l'analyse de foule. Par ailleurs, nous n'avons pas remarqué de travail récent employant activement le Deep Learning pour réaliser la détection de groupes dans des scènes de foule. Toutes les études explorées de cette section sont brièvement présentées dans le Tableau 1.1.

1.4 L'analyse de foule

Comme le montre l'arborescence de la figure 1.11, la littérature de l'analyse de foule se subdivise en deux branches principales : le calcul des statistiques de foule et l'analyse des comportements de foule (ZHAN et al., 2008 ; GRANT et FLYNN, 2017 ; LAMBA et NAIN, 2017). Dans cette section, nous explorons les études publiées dans chacune de ces deux branches principales. Tous les travaux explorés de cette section sont brièvement présentés dans le Tableau 1.2.

Les espaces publics couverts par des caméras de vidéo-protection peuvent être le théâtre de différents niveaux de rassemblements de personnes : ayant différents degrés de niveaux de service LoS. Ces rassemblements peuvent être structurés ou non. Comme le soulignent (THIDA et al., 2013), il est très facile d'identifier des événements anormaux au sein de foules structurées. La tâche devenant fastidieuse face à des foules se déplaçant anarchiquement. Tous les domaines de l'analyse de foule étant sujets à des situations anormales, la détection ou la prédiction d'anomalies peuvent être effectuées pour le calcul des statistiques de foule ainsi que pour l'analyse des comportements de foule.

1.4.1 Calcul des statistiques de foule

Le calcul des statistiques de foule consiste à déterminer la quantité de personnes présentes dans une scène. Cela peut être fait soit en estimant la densité d'une scène

de foule, soit en calculant le nombre de piétons apparaissant dans une scène à travers l'usage de méthodes de détection de piétons. La revue de littérature de (Sindagi et Patel, 2018) montre que de nombreux travaux, ayant eu recours à des méthodes de Deep Learning, ont été consacrés aux statistiques de foule.

Dans cette section, nous commençons par mentionner un travail qui utilise des méthodes *ad hoc* datant de l'ère pré-Deep Learning, alliant méthodes traditionnelles d'extraction de caractéristiques avec des méthodes issues de l'apprentissage automatique dit classique. Après cela, nous présentons principalement des travaux qui emploient l'apprentissage simultané de modèles de comptage du nombre de personnes dans une foule et de méthodes d'estimation de la densité d'une foule.

(Chan, Liang et Vasconcelos, 2008) développent une approche descendante de comptage du nombre de personnes dans une foule qui ne repose pas sur la détection et le suivi des piétons, mais qui exploite des caractéristiques holistiques. L'objectif des auteurs est d'estimer la taille des foules non homogènes. Les auteurs segmentent la foule en utilisant le Mélange des Textures Dynamiques (DTM) (Doretto et al., 2003) Ils extraient les caractéristiques holistiques suivantes des régions segmentées : segments, bords internes (via le détecteur de bord de Canny (Canny, 1986)), textures via la Matrice de Co-Occurrence de Niveau de Gris (GLCM) (Haralick, Shanmugam et Dinstein, 1973). Après cela, la régression du processus gaussien est utilisée pour trouver le nombre de piétons dans chaque segment de foule. Les auteurs valident leur approche sur le jeu de données UCSD Anomaly, qu'ils ont créé. La courbe ROC (Receiver Operating Characteristic) a été utilisée pour valider le processus de segmentation du DTM. Au cours des expériences, le DTM dépasse la méthode NCuts (Shi et Malik, 1998). L'erreur quadratique moyenne (MSE) et l'erreur absolue ont été utilisées pour évaluer le processus gaussien qui approxime le nombre de piétons. Dans le contexte du jeu de données UCSD Anomaly Detection, les auteurs démontrent la supériorité de leur approche et du choix des caractéristiques extraites sur celles qui ont été choisies par (Davies, Yin et Velastin, 1995) et (Kong, Gray et Tao, 2005).

(Marsden et al., 2017) proposent ResnetCrowd, une architecture Residual Network (ResNet) pour apprendre de nombreuses tâches liées à l'analyse de foule. L'architecture vise à apprendre simultanément le comptage du nombre de personnes dans une foule et l'estimation de la densité de la foule ainsi que la détection des comportements violents. L'architecture est basée sur ResNet18 (He et al., 2016), qui est à l'origine pré-entraînée sur le jeu de données ImageNet. Les auteurs l'ont légèrement modifiée pour l'adapter aux tâches que les auteurs comptent réaliser. L'architecture est entraînée, validée et testée sur un jeu de données de leur propre conception, le jeu de données Multi Task Crowd, que nous avons présenté dans la section 1.5.2. L'architecture a été évaluée en termes d'erreur absolue moyenne (MAE), d'erreur quadratique moyenne (MSE) et d'aire sous la courbe (AUC), et elle a été comparée à des méthodes état-de-l'art sur d'autres jeux de données, tels que UCF Crowd 50, WWW Crowd (Shao et al., 2015) et UMN Anomaly. Leurs expériences montrent que les architectures qui sont entraînées pour effectuer de nombreuses tâches proches, donnent de meilleurs résultats pour la reconnaissance de la violence et l'estimation de la densité que celles entraînées uniquement sur une seule tâche. Cependant, (Marsden et al., 2017) notent que les architectures entraînées uniquement sur le comptage du nombre de personnes dans une foule réalisent de meilleures performances que ResnetCrowd dans cette tâche. Il est à noter que la méthode des auteurs est massivement dépassée par les approches de (Zhang et al., 2016b) et (Marsden et al., 2016a) dans le comptage du nombre de personnes dans une foule sur les jeux de données UCF Crowd 50, et légèrement par la reconnaissance de la violence par la

méthode de (Shao et al., 2015) sur le jeu de données WWW Crowd, et légèrement dans la détection des anomalies de (Li, Mahadevan et Vasconcelos, 2014) et de (Marsden et al., 2016b) sur le jeu de données UMN.

Dans la même veine, (Sindagi et Patel, 2017) proposent une approche de bout en bout basée sur une cascade de Réseaux de Neurones Convolutifs (Cascade of CNN) pour apprendre conjointement le comptage des personnes dans une foule et l'estimation de la densité de la scène. Leur but est de faire face aux variations d'échelle et d'apparence, et réaliser un classement du nombre de foules en fonction de l'estimation de la densité de la scène. Pour ce faire, le réseau apprend les caractéristiques globales discriminantes pertinentes pour estimer les cartes de densité. L'architecture de leur modèle se décompose en deux branches. Elle contient des couches convolutives partagées entre les deux branches. La première branche réalise les opérations de comptage du nombre de personnes qui sont nécessaires aux opérations d'estimation de la densité de la seconde branche. L'apprentissage et l'évaluation ont été effectuées sur deux jeux de données publiquement disponibles : ShanghaiTech (Zhang et al., 2016b) et UCF Crowd 50 (Idrees et al., 2013). Comparée à d'autres méthodes sur ces jeux de données, la méthode dépasse les méthodes état-de-l'art en termes d'erreur quadratique moyenne (MSE) et d'erreur absolue moyenne (MAE) sur ShanghaiTech. Cependant, elle est supplantée par les approches de (Onoro-Rubio et López-Sastre, 2016) et de (Walach et Wolf, 2016) en termes de MSE sur UCF CROWD 50.

(Liu, Salzmann et Fua, 2019) proposent l'architecture End-to-End Multi-Scale Trainable Deep Architecture qui réalise l'estimation de la densité d'une scène afin de compter le nombre de personnes dans une foule d'individus. L'architecture extrait des caractéristiques de différentes zones de l'image en utilisant différentes tailles du champ réceptif des Réseaux de Neurones Convolutifs afin de prendre en compte les variations d'échelle. En fonction du contexte, ils ne prédéfinissent pas de patchs d'images adaptés à chaque échelle, mais pondèrent chaque caractéristique extraite avant de fusionner les informations multi-échelles. Leur méthode fonctionne bien sur les caméras non calibrées, mais exploite la présence d'informations de calibration. Ils utilisent un réseau VGG-16 pré-entraîné pour extraire des caractéristiques. Ils obtiennent des caractéristiques sensibles à l'échelle en utilisant la méthode Spatial Pyramid Pooling (He et al., 2015) en considérant 4 échelles différentes. Les caractéristiques extraites des différentes échelles sont par la suite concaténées. Les performances du réseau sont renforcées lorsque les informations de calibration sont disponibles. Pour tirer parti de l'existence des informations de calibration, (Liu, Salzmann et Fua, 2019) utilisent une branche supplémentaire, qui consiste en un réseau VGG tronqué, dédié à l'extraction des caractéristiques d'une carte en perspective qui code le nombre de pixels par mètre. Ils ont mené leurs expériences sur les jeux de données suivants : ShanghaiTech [1], WorldExpo'10, UCF Crowd Counting 50, UCF QNRF [2], et le jeu de données *ad hoc* de Venise. Pour évaluer leurs méthodes, ils utilisent l'Erreur Absolue Moyenne (Mean Absolute Error (MAE)) et l'Erreur Quadratique Moyenne (Root Mean Squared Error (RMSE)). Par rapport aux méthodes de l'état-de-l'art, ils ont observé que leur méthode fonctionne mieux sur les scènes encombrées mais produit des résultats comparables ou parfois moins bons sur les scènes moins denses.

(Wang et al., 2019) proposent un collecteur de données et un étiqueteur qui génère des scènes encombrées et les annote automatiquement. Cela les aide à créer un jeu de données synthétiques nommé GTA 5 Crowd Counting (GCC). Les images du jeu de données proviennent du jeu vidéo Grand Theft Auto V (GTA 5), créé par la

1. ShanghaiTech : https://www.kaggle.com/tthien/shanghaitech
2. UCF QNRF : https://www.crcv.ucf.edu/research/data-sets/ucf-qnrf/

société de développement de jeux vidéos Rockstar Games [3]. Par ailleurs, ils proposent un Réseau de Neurones Convolutifs pour le comptage des personnes dans une scène de foule, appelé Spatial Fully Convolutional Network (SFCN). Ils pré-entrainent, ce dernier, sur le jeu de données GCC, puis l'affinent sur des données réelles. Ils obtiennent leurs meilleurs résultats en utilisant comme support ResNet101 pour le compteur de personnes sur les jeux de données UCF-QNRF, ShanghaiTech A et B, UCF Crowd Counting 50. De plus, ils proposent un compteur de foule adaptatif au domaine afin de convertir les données synthétiques en données réelles. Après cela, ils entraînent le réseau SFCN sur les données converties et le testent sur les données réelles. Le convertisseur qu'ils utilisent est le réseau SSIM Embedding Cycle Generative Adverserial Network, SE Cycle GAN. Il est équipé de la fonction de coût : l'indice de similarité structurelle (SSIM). Cette fonction de coût mesure la distance entre deux images en comparant leurs motifs locaux. Ils ont évalué leur méthode en utilisant la mesure de l'erreur quadratique moyenne (MSE), l'erreur absolue moyenne (MAE), la mesure SSIM et la mesure Peak Signal to Noise Ratio (PSNR) sur les jeux de données suivants : UCF-QNRF, UCF_CC_50, ShanghaiTech A/B, et WorldExpo'10. Leur méthode dépasse Cycle GAN (Zhu et al., 2017) et SFCN (sans l'option d'adaptation au domaine) dans chaque jeu de données.

(Wan et Chan, 2019) proposent d'entraîner conjointement de bout en bout un compteur de personnes dans une foule, un générateur de carte de densité, et un affineur de carte de densité. Leur architecture est constituée d'un réseau pour l'ajustement des cartes de densité et d'un réseau de comptage des personnes. Le réseau de comptage génère une carte de densité à partir d'une scène de foule. L'ajusteur reçoit en entrée une carte de points de vérité terrain qui représentent les positions des individus dans une scène de foule. Le raffineur produit une version améliorée de la carte de densité et la compare à la vérité terrain. Le réseau de raffinement applique une convolution préliminaire qui fait passer la carte des positions de têtes à travers divers noyaux gaussiens. Cette étape produit des cartes de densité floues qui sont masquées à l'aide d'un module d'attention. Ces cartes de densité masquées sont fusionnées pour produire une carte de densité finale. La carte de densité produite est utilisée comme vérité terrain pour entraîner le compteur. (Wan et Chan, 2019) comparent différents réseaux de comptage existants : MCNN (Zhang et al., 2016b), FCN-7c (Kang et Chan, 2018), SFCN (Wang et al., 2019) et CSRNet (Li, Zhang et Chen, 2018). Une combinaison de fonctions de coûts pour le raffinement et le comptage est calculée pendant le processus d'apprentissage pour entraîner conjointement le compteur et le raffineur. Des expériences ont été menées sur les jeux de données suivants : ShanghaiTech A et B, WorldExpo'10, UCF-QNRF. Les méthodes ont été évaluées en utilisant l'Erreur Moyenne Absolue (MAE) et l'Erreur Quadratique Moyenne (RMSE).

Comme nous pouvons le voir dans les travaux mentionnés ci-dessus (Sindagi et Patel, 2017 ; Marsden et al., 2017), l'apprentissage multi-tâches est une bonne idée pour apprendre/ajuster simultanément plusieurs tâches liées au calcul des statistiques de foule. Cependant, l'apprentissage doit être entrepris sur différents types de jeux de données pour produire un modèle performant. Nous observons que les travaux récents sur le calcul des statistiques de foule ont tendance à associer le comptage de foule à l'estimation de la densité d'une scène. Plus précisément, une estimation de densité est entreprise afin de déduire un décompte du nombre de personnes dans une scène de foule. La recherche dans le domaine est en pleine ébullition, et la concurrence assez rude. Toutefois, il y reste beaucoup à faire. L'un des défis majeurs du calcul des statistiques de foule est d'identifier avec précision les positions des têtes dans une

3. https ://www.rockstargames.com/fr/games/V

scène de foule, comme illustré dans le travail de (Sam et al., 2019). Récemment, (Wan, Kumar et Chan, 2020) ont compté le nombre de personnes réalisant une action spécifique dans une scène de foule ce qui est une contribution relativement rare dans le domaine. Nous n'avons pas trouvé de travail similaire et il serait intéressant de voir des travaux du même acabit à l'avenir. Avec une telle contribution, (Wan, Kumar et Chan, 2020) réduisent considérablement le fossé entre le calcul des statistiques de foule et l'analyse des comportements de foule.

1.4.2 L'analyse des scènes de foule

L'analyse des scènes de foule doit être considérée comme une tâche différente de l'analyse du comportement humain dans des scènes individuelles. Cette dernière se concentre sur un spectre réduit de travaux où la classification et l'apprentissage supervisé sont omniprésents (Marsden et al., 2017), contrairement à l'analyse des scènes de foule où les types de techniques d'apprentissage employées et de données rencontrées peuvent être très divers et variés. Au cours des dernières années, l'analyse des scènes de foule a suscité un certain intérêt au sein de la communauté de la vision par ordinateur. De nombreux travaux ont vu le jour. Selon des revues de la littérature récentes (Tripathi, Singh et Vishwakarma, 2018 ; Chong et Tay, 2015), nous pouvons classer les méthodes d'analyse de scènes de foule en deux catégories principales : les méthodes classiques appartenant à l'ère pré-Deep Learning et les méthodes basées sur l'apprentissage profond (Deep Learning). Cependant, comme nous n'avons pas l'intention de réaliser une étude comparative des méthodes d'apprentissage profond utilisées dans l'analyse des comportements de foule, nous n'adopterons pas cette catégorisation pour structurer cette section. Nous suivrons plutôt la taxonomie illustrée dans la figure 1.11.

1.4.2.1 La reconnaissance d'actions

(Hassner, Itcher et Kliper-Gross, 2012) proposent une taxonomie pour les méthodes de reconnaissance de l'action humaine dans une scène individuelle, et les catégorisent en tant que locale, basée sur les points d'intérêt, ou basée sur une seule image, ou globale. L'étude de la reconnaissance et la détection d'actions humaines dans les scènes individuelles n'entre pas dans le cadre de cette thèse, mais nous mentionnons, dans ce chapitre, quelques travaux qui seront pertinents pour la détection et la reconnaissance d'actions dans des scènes de foule.

(Siva et Xiang, 2010) utilisent la méthode de la fenêtre glissante 3D (3D Sliding-Window) pour caractériser des actions à l'intérieur de cuboïdes spatio-temporels dans une vidéo d'une scène de foule. Les auteurs extraient des points saillants et les traquent à l'aide du descripteur de transformation de caractéristiques visuelles invariante à l'échelle (Scale-Invariant Feature Transform (SIFT)) et décrivent le mouvement à l'aide du descripteur de transition de trajectoire (Trajectory Transition descriptor (TTD)). Ces descripteurs sont ensuite utilisés pour construire la représentation Sac de Mots (Bag of Words (BoW)) de chaque action. (Siva et Xiang, 2010) ont divisé la vidéo en 24 canaux pour chaque descripteur (ce qui représente en tout 48 canaux pour les deux descripteurs). Ils utilisent la routine de sélection des canaux (comparable à celle de (Laptev et al., 2008)) pour élire 5 centres de clips parmi les 48 canaux en se basant sur la validation croisée des données d'apprentissage. Un cuboïde d'action est représenté par le sac de mots multicanal. Le cuboïde 3D est glissé à travers l'espace et le temps, et l'algorithme Machines à Vecteurs de Support (Support Vector Machines (SVM)) est utilisé pour apprendre chaque cuboïde d'action 3D sur

une séquence annotée. Les auteurs modélisent le problème de détection d'actions à l'aide de l'apprentissage d'instances multiples (Multiple Instance Learning (MIL)) en considérant les clips contenant une action spécifique comme des sacs positifs et les clips ne la contenant pas comme des sacs négatifs. Ils annotent un seul clip positif avec un cuboïde d'actions. Ils résolvent le problème à instances multiples en utilisant la méthode des K proches voisins (K Nearest Neighbors (KNN)). Leur méthode est validée sur les jeux de données CMU (Ke, Sukthankar et Hebert, 2007) et i-LIDS (Branch, 2006). Les paramètres d'évaluation avec lesquels ils travaillent sont la courbe précision-rappel et la précision moyenne (MAP).

Dans leur contribution pionnière, (Baccouche et al., 2011) proposent l'une des premières utilisations des Réseaux de Neurones Convolutifs 3D dans la classification des clips vidéo pour la reconnaissance des actions humaines. Ils apprennent les caractéristiques spatio-temporelles des clips d'actions à l'aide des Réseaux de Neurones Convolutifs 3D (CNN), après avoir étendu les convolutions 2D vers la 3D. Ils y adjoignent des Réseaux de Neurones Récurrents (RNN) avec des unités de mémoire à long terme (Long-Short Term Memory (LSTM)) pour étiqueter chaque clip en utilisant l'évolution temporelle de ses caractéristiques. L'architecture d'extraction de caractéristiques (3D CNN) contient 10 couches alternant convolutions, rectification et sous-échantillonnage en 8 couches et se terminant par 2 réseaux entièrement connectés (Fully Connected Network (FCN)). L'architecture contient 17169 paramètres entraînables. L'architecture du classifieur (RNN) contient 50 unités LSTM. Elle contient 25000 paramètres entraînables et est entraînée à l'aide de la rétropropagation du gradient (online Backpropagation through time with momentum) (Gers, Schraudolph et Schmidhuber, 2002). L'approche a été testée sur le jeu de données KTH (Schuldt, Laptev et Caputo, 2004) qui contient 6 classes de clips vidéo et sur lequel ils obtiennent de bons résultats par rapport aux méthodes existantes. (Baccouche et al., 2011) suivent le protocole d'évaluation proposé par (Gao et al., 2010) qui repose sur une validation croisée à 5 segments (folds).

(Tran et al., 2018) utilisent une version factorisée des 3D Residual Nets (ResNets) pour modéliser séparément les composants spatiaux-temporels d'un clip vidéo. Ils utilisent des blocs R(2+1)D, qui sont des blocs spatio-temporels convolutifs, entraînés à partir de zéro sur deux jeux de données destinés à la reconnaissance d'actions dans des scènes individuelles : Sports-1m et Kinetics. Ils ont affiné leurs blocs sur deux autres jeux de données moins volumineux : UCF-101 et HMDB-51. (Tran et al., 2018) ont comparé les variantes de leur modèle à plusieurs variantes de l'architecture Inflated 3D (Carreira et Zisserman, 2017). Leur modèle réalise de bonnes performances en termes de précision mais est dépassé par un modèle issue de l'architecture TwoStream Inflated 3D sur les jeux de données UCF-101 et HMBD-51.

Les conclusions des travaux précédents nous amènent aux travaux de (Carreira et Zisserman, 2017) sur l'architecture Inflated 3D ConvNets. Leur modèle repose sur les Inflated 2D ConvNets. Lors de leurs expériences sur les jeux de données UCF-101 et HMDB-51, ils ont supplanté, en termes d'accuracy, tous les modèles récents qu'ils ont évalués. Les modèles TwoStream I3D, qui ont été pré-entraînés sur Kinetics-400, ont atteint en moyenne $80,9\%$ d'accuracy, et les modèles TwoStream-I3D, pré-entraînés sur ImageNet et Kinetics, ont atteint 98% d'accuracy.

Les deux travaux précédents portent sur la détection/reconnaissance d'actions dans des scènes individuelles. Dans l'analyse de foule, nous sommes davantage intéressés par la détection et la reconnaissance d'actions dans des scènes de foule. (You et Jiang, 2018) proposent l'architecture Action4D-Net. Ils commencent par détecter et réaliser le tracking de chaque personne dans une scène, puis utilisent le Action4D-Net pour reconnaître l'action que cette personne effectue. Ils ont entraîné leur modèle sur

un jeu de données d'actions modélisées en 4 dimensions (3 dimensions spatiales sur les axes X,Y,Z + la dimension temporelle), qu'ils ont eux-mêmes créé. Bien que leur méthode puisse être utilisée en temps réel dans un système multi-caméras, l'utilisation des caméras RGBD n'est pas assez répandue pour rendre leur algorithme applicable à tous les contextes de mouvements de foule. Ils ont testé leur méthode en termes d'accuracy, d'accuracy révisé (RAcc), et de matrice de confusion.

(Wei et al., 2020) soutiennent que la nature d'un rassemblement découle de l'humeur et du comportement des individus qui le constituent. Dans cette optique, ils proposent un modèle de représentation d'une foule, résumée par le triplet suivant : <Comportement, Humeur, Organisation> <Behavior, Mood, Organization (BMO)>. Ils perçoivent trois types de foule, chacun avec un certain nombre de caractéristiques auxquelles ils associent un système d'alerte fondé sur des règles :

1. les foules hétérogènes qui affluent vers des lieux communs communément bondés tels que les gares et les supermarchés, ne nécessitent pas une surveillance accrue,
2. les foules homogènes à l'origine des manifestations, des parades urbaines à caractère culturel ou sportif, doivent être gérés modérément,
3. les foules prenant part à des manifestations violentes qui résultent d'une exacerbation d'une manifestation à caractère socio-politique doivent être maîtrisées.

Pour apprendre ce modèle de représentation, ils proposent d'entraîner le réseau de neurones Crowd Type Recognition Network (CTRN). Le CTRN dispose d'une architecture de réseau à deux branches. Chaque branche est un réseau de type Visual Geometry Group (VGG16) constitué de 5 couches de neurones convolutifs qui ont été pré-entraînés sur le jeu de données ImageNet. La 1ère branche reçoit en entrée une image (RVB) statique d'une scène, et la seconde branche reçoit en entrée une carte de mouvement d'une scène qui contient des caractéristiques de trajectoires. Le réseau CTRN est inclus dans un système qui lance des alertes lorsque le triplet BMO remplit certaines conditions. Les modèles issus de cette architecture sont entraînés et évalués sur les jeux de données CUHK Crowd et Normal-Abnormal.

— CUHK Crowd est constitué de foules homogènes et hétérogènes ;
— Normal-Abnormal est un jeu de données *ad hoc* qui contient des foules homogènes et hétérogènes et très souvent violentes.

La représentation BMO d'une foule conduit à l'usage d'un algorithme de classification qui classe ces foules en produisant des étiquettes sous la forme d'un triplet BMO. Toutefois, les conditions de la remontée d'une alerte (c.f. Tableau 11 de l'article de (Wei et al., 2020)) ne répondent pas aux cas particuliers d'une foule hétérogène qui pourrait être mise en danger par le comportement anormal d'éléments individuels. Les mesures d'évaluation utilisées sont la courbe (Receiver Operating Characteristics (ROC)), l'aire sous la courbe de ROC (Area Under Curve (AUC)), et l'accuracy moyenne (Mean Accuracy (MA)).

(Wang et O'Sullivan, 2016) proposent d'utiliser un Processus de Dirichlet Hiérarchique Spatio-temporel (Spatio-temporal Hierarchical Dirichlet Process (STHDP)) afin d'entraîner des modèles spatiaux et temporels à détecter des activités normales et anormales dans des données vidéo. Le temps est représenté comme un processus continue non-markovien, ce qui permet de gérer la durée fluctuante de chaque activité. Le STHDP est une méthode de clustering bayésienne non paramétrique qui ne nécessite pas beaucoup de connaissances préalables sur la dynamique des foules. Dans ce contexte, le nombre de clusters n'est pas prédéfini. Chaque cluster est un ensemble de trajectoires. Les auteurs ne regroupent pas les trajectoires à l'aide d'une

métrique de distance, mais regroupent les observations individuelles des trajectoires d'une image à l'autre. La méthode capte la date d'apparition d'une activité, sa durée, et sa date de disparition. La méthode a été entraînée et testée sur des données synthétiques et sur des données réelles. Trois jeux de données ont été utilisés pour les données réelles :

1. Edinburgh dataset (Forum),
2. MIT Carpark (Carpark),
3. New York Central Terminal (Train Station).

Sur ces jeux de données, STHDP a été comparé à deux méthodes non récentes : MOTIF (Emonet, Varadarajan et Odobez, 2011) et les Processus de Dirichlet Duaux Hiérarchiques (Dual Hierarchical Dirichlet Processes (DHDP)) (Wang et al., 2011b). Bien que STHDP soit plus lent que MOTIF et aussi rapide que DHDP, STHDP est légèrement plus accurate que DHDP. Il est amplement plus accurate que MOTIF.

(Yan, Zhu et Yu, 2019) proposent une approche de sous-titrage de vidéo de scènes de foule appliquée aux spectateurs d'un événement quelconque. Les auteurs trouvent ce type de foule peu analysé par les travaux de recherche en analyse de foule. Dans leur travail, ils proposent de classer les comportements de cette foule en générant 8 commentaires différents illustrant simultanément la densité et le comportement d'une foule :

— someone walk in : quelqu'un entre
— someone run in : quelqu'un entre en courant,
— someone walk out : quelqu'un sort,
— someone run out : quelqu'un sort en courant,
— many people walk in : beaucoup de personnes entrent,
— many people walk out : beaucoup de personnes sortent,
— many people run in : beaucoup de personnes entrent en courant,
— many people run out : beaucoup de personnes sortent en courant.

Comme le suggèrent les classes-commentaires possibles, ces foules de spectateurs se produisent aux entrées/sorties des stades, théâtres, rassemblements, etc. Ils appliquent leurs méthodes sur le jeu de données WorldExpo'10 et les évaluent à l'aide des métriques d'évaluation suivantes pour le sous-titrage : CIDER, METEOR, BLEU, ROUGE. Ils proposent un pipeline d'un codeur-décodeur constitué d'un extracteur de caractéristiques (le codeur) qui alimente un réseau séquence-à-séquence qui convertit la vidéo en texte (le décodeur). L'extracteur de caractéristiques est soit un réseau 3D ConvNets (C3D) pré-entraîné sur le jeu de données Sports-1m, soit ResNet-152, Inception V3 / V4, chacun d'eux étant formé sur ImageNet. Le réseau séquence-à-séquence se nomme Sequence to Sequence-Video To Text (S2VT) et a été proposé par (Venugopalan et al., 2015). Le S2VT est un réseau neuronal récurrent (RNN) à deux couches dont les cellules sont soit Long Short-Term Memory (LSTM), soit Gated Recurrent Unit (GRU). Ils évaluent toutes les combinaisons possibles, en termes de précision et les métriques mentionnées ci-dessus, pour sélectionner la meilleure combinaison. Cette dernière finit par être le réseau Inception V3 comme extracteur de caractéristiques couplé avec réseau S2VT ayant pour cellules les unités GRU.

(Ullah et al., 2016) travaillent sur la classification des configurations des mouvements de foule. Dans leur contribution, ils identifient 5 types de configurations de foule. Ceux-ci sont : la configuration en forme de voie, la configuration en forme de arc ou anneau, la configuration qui aboutit en un goulot d'étranglement, la situation de blocage, la configuration en forme de tête de fontaine. Ils proposent une approche

basée sur l'extraction de flux optique à partir d'un clip vidéo, en utilisant l'algorithme de Farnebäck (Farnebäck, 2003), sur laquelle ils appliquent le Processus de Diffusion Thermique (TDP) (Wang et al., 2014) pour le rendre plus cohérent. Après cela, les particules en mouvement extraites sont limitées aux individus sur lesquels ils appliquent une version modifiée du modèle de force sociale (Modified Social Force Model (M-SFM)) pour comprendre les interactions entre les individus. À la fin de ce processus, ils se retrouvent avec un système dynamique continu qui décrit le champ de flux de mouvement dont ils extraient le premier ordre et les dérivées du second ordre pour identifier le comportement de la foule. Leur méthode a été validée sur le benchmark proposée par (Solmaz, Moore et Shah, 2012) et ensuite le jeu de données UCD (Ullah et Conci, 2012). Ils ont évalué leur méthode à l'aide du F1-score, et l'ont comparée à la méthode de (Solmaz, Moore et Shah, 2012) où ils rencontrent le plus de difficultés avec la configuration en forme de voie.

Plus tard, (Ullah et al., 2019) ont proposé la classification des scènes de foule. Ils proposent une architecture d'un Réseau de Neurones Convolutifs composé de deux branches. Une branche extrait des caractéristiques spatiales d'une image en RVB (Rouge-Vert-Bleu). La seconde branche extrait des caractéristiques temporelles d'un descripteur basé sur la trajectoire (Wang et al., 2011a) calculée sur un champ de flux de mouvement d'un certain nombre d'images consécutives. Les flux sont initialisés avec des poids pré-entraînés sur le jeu de données ImageNet. Leur méthode est évaluée sur le jeu de données CUHK Crowd et comparée à des méthodes état-de-l'art en termes d'accuracy et de matrices de confusion : Tensor Learning Classification (TLC) (Zhang, Liu et Jiang, 2018), Spatio-Temporal Classification (STC) (Li, Liang et Jin, 2016), et Energy-Based Features (Zhang et al., 2018).

À travers toutes les approches précédemment vues en reconnaissance d'actions dans des scènes individuelles, en reconnaissance d'actions dans des scènes de foule, en reconnaissance des mouvements de foule qui est aussi apparenté à la classification des mouvements de foule, nous observons que même si la reconnaissance d'actions dans des scènes individuelles est un sujet très souvent abordé en traitement de vidéos, la reconnaissance d'actions ou de comportements dans des scènes de foule n'est pas suffisamment traitée dans les travaux récents, malgré l'existence de quelques études récentes plus qu'intéressantes (Dupont, Tobias et Luvison, 2017 ; Yan, Zhu et Yu, 2019 ; Ullah et al., 2016 ; Ullah et al., 2019). L'application de modèles issus de l'apprentissage profond développés pour la détection et la reconnaissance d'actions dans des clips vidéo de scènes de foule est rarement abordée en raison de la rareté des données et de leur faible diversité.

1.4.2.2 Suivi et prédiction des trajectoires

Avant le début de l'application massive des approches issues de l'apprentissage profond (Deep Learning) en vision par ordinateur (Computer Vision) à partir de 2012, la recherche en analyse de trajectoires et de mouvements était partagée entre :

— L'analyse des flux de foule, qui s'inscrit dans le cadre d'approches descendantes ou holistiques. Une foule étant considérée comme une seule entité qui est par la suite segmentée en plusieurs zones de flux distincts.

— L'analyse des trajectoires des piétons qui s'inscrit dans le cadre d'approches ascendantes ou atomistes. Ces approches partent de la détection des piétons ou des indices visuels, de l'extraction des trajectoires pour comprendre le mouvement global dans une scène.

(Ali et Shah, 2007) utilisent la dynamique des particules Lagrangiennes pour segmenter un flux de foule et détecter les instabilités au sein de la foule. Leur approche

est à la croisée des chemins entre l'analyse de mouvement, l'analyse des comportements de foule, et la détection d'anomalies. Les auteurs considèrent la foule en mouvement comme un système dynamique apériodique. Ils commencent par calculer un champ de flux optique qui décrit les interactions des individus au sein d'une foule et l'influence qu'exerce l'environnement physique sur ces individus. Ils projettent une grille de particules sur ce champ d'écoulement et l'advectent pour créer des cartes de flux. Après cela, ils calculent le champ de l'exposant de Lyapunov à temps fini (Finite Time Lyapunov Exponent (FLTE)) à partir du tenseur de gradient spatial de la carte de flux. Cette opération donne les Structures Cohérentes Lagrangiennes (Lagrangian Coherent Structures (LCS)) qui permettent de diviser la foule en plusieurs flux. L'approche est testée sur des clips tirés de Getty-Images, Photo-Search, Google Videos, et Inside Mecca (un documentaire de National Geographic). Cependant, l'approche n'a pas été testée en utilisant des paramètres appropriés ni comparée à d'autres méthodes. Le code source de cette approche est disponible publiquement sur github [4].

Dans une autre contribution, (Ali et Shah, 2008) proposent une approche probabiliste pour suivre des individus dans une scène de foule très dense basée sur le modèle de force social qui prend en compte la structure de la scène. Les contraintes de ce modèle sont déterminées par trois Floor Fields inspirés du domaine de la dynamique des évacuations. Les trois Floor Fields sont :

1. Static Floor Field (SFF) : qui représente les endroits vers lesquels la foule se dirige.
2. Dynamic Floor Field (DFF) : qui représente les forces induites de l'interaction d'un individu avec la foule environnante. Ici le modèle de force sociale est calculé à partir de l'advection de particules [5] qui est obtenue lors de l'extraction du flux optique.
3. Boundary Floor Field (BFF) : qui représente les bords de la scène et que l'on déduit grâce au calcul du Finite-Time Lyapunov Exponent (FTLE).

(Ali et Shah, 2008) posent une grille de cellules sur la scène. Chaque cellule de la grille représente une particule ou un piéton. Les trois Floor Fields sont calculées sur les particules. Ces contraintes permettent de prédire les futures localisations des piétons. (Ali et Shah, 2008) ont entrepris leurs expériences sur les trois séquences du jeu de données Marathon. Leur méthode a été comparée en termes d'accuracy à la méthode Mean-Shift Tracker. Leurs résultats montrent que les auteurs ont rencontré des problèmes de suivi des piétons dans des scènes de foule lorsque leur approche est confrontée à de graves changements d'éclairage et à une occlusion. Cependant, ils arrivent à faire face aux occlusions partielles.

Avec l'émergence du Deep Learning, le développement des trackers a pris un second souffle ces dernières années (Alahi, Ramanathan et Fei-Fei, 2017 ; Sadeghian, Alahi et Savarese, 2017 ; Bera, Kim et Manocha, 2018). L'accent est davantage mis sur un meilleur compromis entre la vitesse et l'accuracy (Bewley et al., 2016). Dans ce qui suit, nous listons quelques trackers qui ont retenu notre attention.

(Bewley et al., 2016) proposent SORT (Simple Online and Realtime Tracking), un outil de suivi en temps réel des piétons, qui repose sur l'usage du filtre de Kalman (Kalman, 1960) avec l'algorithme Hongrois (Kuhn, 1955). Pour réaliser des détections préalables de piétons sur les clips vidéos, ils ont utilisé le Réseau de Neurones

4. https ://github.com/saadali37/Crowd-Flow-Segmentation
5. Une définition de l'advection de particules est proposée par (Aref, 1990)

Convolutifs Faster Region CNN (FrRCNN) (Ren et al., 2015) qui est basé sur l'architecture VGG16. Les piétons détectés sont représentés par des Bounding Boxes. Le filtre de Kalman permet de calculer les composantes de vitesse d'une cible lorsqu'une nouvelle détection lui est associée. L'algorithme Hongrois est utilisé pour attribuer une nouvelle détection à une cible en fonction de la distance IoU Intersection-over-Union. SORT est capable de faire face aux occlusions de courte durée dans les vidéos. Testé sur 11 séquences tirées du jeu de données Multiple Objects Tracking (MOT) (Leal-Taixé et al., 2015), et comparé à 9 autres trackers, l'algorithme SORT s'avère 20 fois plus rapide que l'état-de-l'art. Il obtient un bon score d'accuracy, ce qui rend ses performances très proches de celles des trackers de l'état-de-l'art. Le point faible de SORT sont les occlusions de longue durée, à l'issue desquelles il rate un grand nombre des réidentifications de piétons, ce qui aggrave son score des changements d'identités (Identity Switches).

Dans le prolongement des travaux de (Bewley et al., 2016), (Wojke, Bewley et Paulus, 2017) ont développé DeepSORT, une amélioration de SORT qui inclut désormais une métrique d'association pré-entraînée aidée par des informations sur les apparences et le mouvement. Ceci permet de remédier aux occlusions de longue durée tout en conservant une vitesse de suivi similaire à celle de l'algorithme SORT. La métrique d'association est pré-entraînée sur le jeu de données de ré-identification des personnes MARS (Zheng et al., 2016) à l'aide d'un Réseau de Neurones Convolutifs. Entre SORT et DeepSORT, les changements majeurs se produisent au niveau de la procédure d'affectation de nouveaux états : (Bewley et al., 2016) formulent le problème d'affectation comme suit : le filtre de Kalman prédit les états initiaux. Ensuite, les potentiels nouveaux états sont associés aux états existants en utilisant l'algorithme Hongrois. Avec DeepSORT, (Wojke, Bewley et Paulus, 2017) intègrent les caractéristiques de mouvement et les indices d'apparence à la formulation de la procédure d'affectations de SORT. (Wojke, Bewley et Paulus, 2017) incorporent la caractérisation du mouvement en calculant le carré de la distance de Mahalanobis entre les états prédits par le filtre de Kalman et les états nouvellement calculés. Ils intègrent la fonction d'apparence en calculant la distance cosinus entre le descripteur d'apparence de la dernière détection et les descripteurs d'apparence des pistes existantes. Les Bounding Box, qui servent de descripteurs d'apparence, sont générées par un Réseau de Neurones Convolutifs. Pour réduire davantage les effets préjudiciables des occlusions de longue durée, (Wojke, Bewley et Paulus, 2017) ont renforcé leur procédure d'affectation avec une procédure de correspondance en cascade, qui accorde, lors de l'affectation de nouveaux états, la priorité aux trajectoires encore actives des piétons fréquemment identifiés. Comparé à SORT et à d'autres suiveurs de trajectoires sur le jeu de données MOT16 (Milan et al., 2016), DeepSORT s'avère assez compétitif mais souffre de quelques défauts. Même s'il réduit les changements d'identités (Identity Switches) d'une proportion de 45% par rapport à l'algorithme SORT, il pâtit d'un nombre élevé de faux positifs. Les séquences sur lesquelles la méthode a été évaluée ont des caméras fixes et des caméras piétons mobiles.

(Lamba et Nain, 2019) proposent une méthode de clustering de trajectoires basée sur les contours pour la segmentation d'une scène de foule. Leur but est de détecter des situations d'encombrement pouvant déboucher sur des catastrophes en cas d'un mouvement de foule brusque ou de bousculade. Leur méthode commence par séparer les régions occupées par la foule de l'arrière plan immobile. Par la suite, des points d'intérêt sont extraits de la foule et sont traqués, tout au long d'un clip vidéo, à l'aide de l'algorithme Kanade-Lucas-Tomasi (Tomasi et Kanade, 1991). Les trajectoires sont regroupées à l'aide de la distance de Jaccard en fonction des caractéristiques de position, de densité, de forme, et de direction. La foule en elle-même est segmentée

à l'aide de l'algorithme DBSCAN. La densité de chaque flux de foule est analysée pour détecter des situations de congestions locales. Leur méthode a été entraînée et évaluée sur le jeu de données UCF Crowd Dataset, Collective Motion [6] et Violent Flows. Leurs performances ont été comparées à celles de méthodes classiques, en termes de la métrique de Jaccard, du F score, et de l'Erreur Absolue Moyenne (Mean Average Error (MAE)).

(Li et al., 2019) proposent un algorithme de clustering des trajectoires de piétons pour en déceler des groupes. Ils se basent sur le différentiel de vitesses et de positions entre deux trajectoires pour en évaluer la distance. Dans un premier temps, ils emploient une combinaison entre un algorithme de clustering des pics de densité et un algorithme glouton pour effectuer un regroupement de haut niveau des trajectoires. Dans cette première étape, les auteurs n'ont pas besoin de spécifier le nombre de clusters. Dans un second temps, ils utilisent une variante de la distance moyenne de Hausdorff pour effectuer un clustering plus fin. L'approche de (Li et al., 2019) est évaluée sur un scénario unique d'une scène du monde réel prise par une caméra HD d'un drone qui filme une foule de 67 piétons à une intersection de rues. Ils utilisent le Coefficient de Silhouette (Silhouette Coefficient) et le Taux de Consistence (Consistency Rate) pour évaluer la qualité du clustering le plus fin. Ils comparent leur méthode avec le Fundamental Diagram (FD) (Favaretto, Dihl et Musse, 2016), et les méthodes de (Ge, Collins et Ruback, 2009 ; Ge, Collins et Ruback, 2012), et obtiennent une meilleure accuracy.

(Wu et al., 2017b) proposent d'employer le descripteur Curl and Divergence of Motion Trajectories (CDT) pour caractériser le mouvement dans une scène de foule et exploiter ces caractéristiques pour en déduire la classe. Dans leur approche, ils classent les scènes de foule en cinq catégories :

1. Lane : la foule adopte une configuration linéaire.
2. Clockwise arch : la foule prend la forme d'un arc de cercle et avance dans le sens des aiguilles d'une montre.
3. Counterclockwise arch : la foule toujours sous la forme d'un arc de cercle, évolue dans le sens contraire des aiguilles d'une montre.
4. Bottleneck : la foule aboutit à un goulot d'étranglement.
5. Fountainhead : la configuration d'une foule en forme d'une source vive.

Ils commencent par extraire le flux optique d'un clip vidéo à l'aide de la méthode Lucas-Kanade (Lucas et Kanade, 1981), puis appliquent un clustering sur le flux optique pour obtenir le champ des vecteurs vitesse. Après cela, ils appliquent l'advection de particules (Solmaz, Moore et Shah, 2012) pour décomposer le champ de vecteurs en sous-champs. À partir de ces sous-champs, ils extraient les descripteurs CDT qui décrivent le Curl le long des axes tangents et la divergence le long des axes radiaux. L'application d'un regroupement des caractéristiques sur les descripteurs CDT, permet d'obtenir une caractérisation des mouvements. Cette caractérisation est envoyée en entrée de l'algorithme des Machines de Vecteurs de Support (Support Vector Machine (SVM)) pour classer les différents mouvements de foule à l'aide de la validation croisée d'un contre tous (Leave-One-Out Cross Validation). Ils évaluent leur méthode en termes des métriques (Receiver Operating Characteristic (ROC)) et Area Under the ROC Curve (AUC) sur les jeux de données de UCF Crowd, CUHK Crowd, et une combinaison de UCF Crowd et CUHK Crowd. Leur méthode a été comparée à d'autres méthodes SOTA, où ils obtiennent des résultats compétitifs sauf sur la classe de clips illustrant des goulots d'étranglement.

6. Collective motion : http://mmlab.ie.cuhk.edu.hk/projects/collectiveness/dataset.htm

(Alahi et al., 2016) proposent l'approche Social-LSTM, un Réseau de Neurones Récurrents (Recurrent Neural Network (RNN)) qui contient des unités Long-Short-Term-Memory (LSTM), pour simultanément apprendre et prédire les mouvements des piétons en fonction de la foule environnante. À partir d'un clip vidéo d'une scène de foule, les auteurs obtiennent les positions des piétons sur toutes les images. Ensuite, ils utilisent un modèle RNN-LSTM distinct pour apprendre la trajectoire de chaque piéton. Les LSTM sont connectés les uns aux autres grâce à une couche de Social-Pooling, de sorte que les RNN-LSTM spatialement proches partagent des informations mutuelles et les récupèrent à chaque nouvelle étape. Ils entraînent leur modèle en suivant une stratégie de validation croisée et l'évaluent sur les jeux de données UCY (Lerner, Chrysanthou et Lischinski, 2007) et ETH (Pellegrini et al., 2009). Social-LSTM est évaluée en termes des métriques Erreur de Déplacement Moyen (Average Displacement Error (ADE)), d'erreur de déplacement final (Final Displacement Error (FDE)), d'erreur de déplacement non linéaire moyenne (Average Non Linear Displacement Error (ANLDE)). Ils comparent leur méthode au filtre de Kalman, une méthode d'évitement de collision basée sur le modèle de force sociale (Yamaguchi et al., 2011), le processus gaussien itératif (Iterative Gaussian Process (IGP)) (Peter et al., 2013), un modèle LSTM de base (Vanilla LSTM) dépourvu de la couche du Social-Pooling, et une version plus simple de leur modèle qui ne contient que les cartes d'occupation du terrain par la foule (O-LSTM). Cette dernière capture les positions des voisins à un instant t sans prendre en compte les positions précédentes. Les modèles évalués apprennent les trajectoires pendant 3.2 secondes et s'attellent à prédire les 4.8 secondes suivantes. Les résultats globaux montrent que Social-LSTM obtient de meilleures performances que les autres méthodes dans les deux jeux de données. Toutefois, Social-LSTM est parfois dépassé par IGP qui arrive à exploiter la connaissance préalable sur la destination finale de chaque piéton contrairement aux autres méthodes. O-LSTM obtient les meilleurs résultats sur des scènes moins denses.

(Bartoli et al., 2018) proposent un réseau RNN qui utilise des unités de mémoire de type LSTM pour apprendre le déroulement des trajectoires des piétons dans plusieurs contextes spatiaux et sociaux. L'objectif du réseau est de prédire les futurs déplacements de chaque piéton. Ils augmentent les modèles Social-LSTM et O-LSTM proposés par (Alahi et al., 2016) en leur incorporant un Context-Aware Pooling qui prend en compte l'interaction d'un piéton avec son contexte social et son interaction avec les objets statiques qui se trouvent à sa proximité. Cette approche nécessite une connaissance préalable de la configuration spatiale de la scène. Ils évaluent leur méthode, en termes d'Erreur de Déplacement Moyen (ADE), sur les jeux de données UCY (Lerner, Chrysanthou et Lischinski, 2007) et MuseumVisits (Bartoli et al., 2018). Comme dans (Alahi et al., 2016), les méthodes apprennent les trajectoires pendant 3.2 secondes et s'attellent à prédire les 4.8 secondes suivantes. Renforcée par l'avantage que lui confèrent les cartes d'occupation de la scène, la méthode Context-Aware O-LSTM semble bien fonctionner sur les scènes d'intérieur du Musée (MuseumVisits) car les personnes s'y déplacent souvent en groupe et se dirigent très fréquemment vers les œuvres d'art. Context-Aware Social-LSTM, elle, fonctionne mieux sur les séquences du jeu de données UCY, car ici, les personnes se déplacent seules et la connaissance préalable des points d'entrée et de sortie, utiles pour deviner le but de chaque piéton, ont été mémorisés par le modèle.

De nombreux travaux ont été effectués sur le suivi de piétons (Bewley et al., 2016 ; Wojke, Bewley et Paulus, 2017) et l'analyse des trajectoires (Lamba et Nain, 2019 ; Li et al., 2019 ; Wu et al., 2017b). La publication de nouveaux travaux dans ces deux axes de recherche est devenue très difficile en raison d'une forte

concurrence et d'un état-de-l'art présentant de très bonnes performances. Cet état de blocage semble inciter davantage de chercheurs à s'orienter vers la prédiction des trajectoires, ces dernières années, où la recherche est encore récente (Coscia et al., 2018 ; Tang et al., 2018 ; Alahi et al., 2016 ; Bartoli et al., 2018).

1.4.2.3 Analyse des comportements de groupe

Ici, nous présentons quelques approches utilisées pour l'analyse des comportements de groupe. Aujourd'hui, le Deep Learning y est largement répandu. Cependant, malgré son ancrage actuel, l'usage de ces méthodes a commencé plusieurs années après les premiers succès du Deep Learning en Vision par Ordinateur.

(Shao, Change Loy et Wang, 2014) proposent une approche pour détecter des groupes dans une scène de foule et comprendre leurs comportements. Ils se basent sur la connaissance préalable dénommée transition collective (Collective Transition (CT)) afin de détecter des groupes. Cette connaissance préalable aide à trouver des descripteurs visuels qui décèlent des groupes et à révéler les propriétés de la scène qui ne sont pas influencées par les variations de la densité de foule. Ces propriétés de la scène sont : la stabilité, la collectivité, l'uniformité, et les frictions. Ces descripteurs aident à déterminer les états internes et les comportements des groupes et les catégorisent dans les classes suivantes : gaz, solide, fluide pur, fluide impur. La transition collective est un algorithme de regroupement. Les auteurs commencent par extraire les tracklets de chaque piéton à l'aide du suiveur de caractéristiques Kanade-Lucas-Tomasi (Tomasi et Kanade, 1991). Les clusters initiaux de tracklets sont trouvés en utilisant la méthode de filtrage cohérent (Coherent Filtering) (Zhou, Tang et Wang, 2012). Après cela, des tracklets sont choisies pour représenter les centres des clusters (groupes d'individus). Les tracklets sont regroupées à l'aide de l'algorithme Espérance-Maximisation (EM). Les propriétés intra-groupe et inter-groupes qui sont la stabilité, la collectivité, l'uniformité et le conflit, sont validées à l'aide de l'algorithme des K plus proches voisins (K Nearest Neighbors (KNN)). Afin d'évaluer le détecteur de groupes, des expériences ont été entreprises sur le jeu de données CUHK Crowd, qui a été mis en place par les auteurs. Ils ont comparé leur méthode à une Mixture des Textures Dynamiques (Dynamic Textures Mixture (DTM)) (Chan et Vasconcelos, 2008), au Clustering Hiérarchique (HC) (Ge, Collins et Ruback, 2012), et au Filtrage Cohérent (Coherent Filtering (CF)) (Zhou, Tang et Wang, 2012). Afin d'évaluer les performances de leur détecteur, ils ont utilisé l'information mutuelle normalisée (Normalized Mutual Information (NMI)) (Wu et Schölkopf, 2007), la pureté (Aggarwal, 2004), et l'indice de Rand (RI) (Rand, 1971). Pour la tâche de classification des comportements de groupe, les métriques d'évaluation qui ont été utilisées sont l'accuracy et la matrice de confusion.

(Vahora et Chauhan, 2018) proposent une approche ascendante basée sur le Deep Learning pour identifier des activités de groupe. Leur méthode repose sur la prise en compte des interactions sociales et spatiales. Ils utilisent les Réseaux de Neurones Convolutifs pour extraire des informations sur les postures des individus, leurs actions individuelles, et la configuration de la scène lors de la réalisation d'une action de groupe. Les Réseaux de Neurones Récurrents sont utilisés pour apprendre les variations dans la configuration d'un groupe d'individus. (Vahora et Chauhan, 2018) ont développé deux approches : l'une basée sur les unités LSTM et l'autre basée sur les unités GRU. Après l'évaluation de leurs modèles sur le jeu de données Collective Activity de (Choi, Shahid et Savarese, 2009) et l'avoir comparé à 6 autres approches de l'état-de-l'art, ils ont trouvé que l'approche reposant sur les unités GRU obtient les meilleures performances en termes d'accuracy.

(Shu, Todorovic et Zhu, 2017) proposent le Réseau de Neurones Récurrents de (Confidence-Energy Recurrent Neural Network (CERN)) qui utilise des unités de mémoire LSTM pour apprendre les actions humaines individuelles, les interactions humaines, ainsi que les activités de groupe. Ils ont évalué les deux variantes de leur approche sur les jeux de données Collective Activity (Choi, Shahid et Savarese, 2009) et Volleyball (Ibrahim et al., 2016), et les ont comparées à d'autres méthodes. Lors de ces expériences, la variante CERN-2 obtient les meilleures performances en termes d'accuracy.

(Zitouni, Sluzek et Bhaskar, 2020) perçoivent dans une foule deux types d'éléments : les piétons isolés et les groupes de piétons. Dans ce contexte, les groupes sont séparés en groupes ordinaires qui s'illustrent par des individus aux comportements homogènes et des groupes mixtes qui s'illustrent par des individus aux comportements hétérogènes. (Zitouni, Sluzek et Bhaskar, 2020) proposent une approche algorithmique où les piétons, les têtes des piétons, et les groupes de piétons sont d'abord détectés puis traqués par un suiveur de trajectoires. Par la suite, ils utilisent la méthode de détection Gaussian Mixture Models of Dynamic Textures (GMM-of-DT) (Zitouni, Bhaskar et Al-Mualla, 2016) qui se base sur le filtre de Kalman afin de classer les comportements de foule en 4 catégories : actions individuelles, activités de groupe, actions découlant d'un comportement chef de file-suiveurs (leader-follower), et interactions sociales. L'approche a été validée sur 6 séquences du jeu de données PETS-2009 (Ferryman et Shahrokni, 2009). Elle a été évaluée, en utilisant le F-Score, sur des séquences de Parking-Lot (Dehghan, Modiri Assari et Shah, 2015) et Town Center (Benfold et Reid, 2011).

(Bisagno, Zhang et Conci, 2018) proposent Group-LSTM, une approche qui permet de détecter des groupes de piétons et de prédire leurs trajectoires. Tout d'abord, ils utilisent l'approche de filtrage cohérent (Coherent Filtering (CF)) (Zhou, Tang et Wang, 2012) pour regrouper les trajectoires et former des groupes en fonction des trajectoires des individus et de leur contexte spatial. Selon eux, les piétons marchant dans la même direction appartiennent forcément au même groupe. Ils utilisent une version augmentée de l'approche Social-LSTM (Alahi et al., 2016) pour prédire les trajectoires de groupes. Alors que Social-LSTM prédit uniquement les trajectoires des piétons en associant un LSTM pour chacun d'entre eux, (Bisagno, Zhang et Conci, 2018) ajoutent une couche cachée de Social-Pooling pour centrer l'attention sur tout le groupe de piétons afin de prédire la trajectoire du groupe. Ils évaluent leur méthode sur les jeux de données UCY et ETH, en se basant sur l'Erreur de Déplacement Moyen (ADE) et l'Erreur de Déplacement Final (FDE). Ils comparent leur méthode à l'approche Social-LSTM (Alahi et al., 2016), sa variante proposée par (Gupta et al., 2018), et au filtre de Kalman. En termes d'ADE, (Bisagno, Zhang et Conci, 2018) obtiennent les meilleurs résultats, mais en termes de FDE, ils ne parviennent pas toujours à trouver la position finale d'un groupe, contrairement à d'autres méthodes.

L'analyse des comportements de groupe ne capte pas l'attention de la recherche autant que le suivi de trajectoires et la reconnaissance d'actions dans des scènes individuelles. Ce manque d'intérêt peut être imputable au manque de données annotées illustrant des activités de groupe. Publier dans cet axe de recherche requiert de passer outre ce manque de données en réalisant des expériences sur des jeux de données non destinés initialement à l'analyse des comportements de groupe comme le font (Bisagno, Zhang et Conci, 2018 ; Zitouni, Sluzek et Bhaskar, 2020) sur ETH, UCY, et PETS-2009, et de travailler essentiellement sur des approches issues de l'apprentissage non-supervisé.

1.4.3 Détection d'anomalies

La détection et la prédiction d'anomalies peuvent faire recette dans n'importe quel thème de l'analyse de foule. Cet axe de recherche retient de plus en plus l'attention dans l'analyse de foule en raison de ses diverses applications en vidéo-protection et en gestion des mouvements de foule (Zhan et al., 2008).

Inspirés par l'efficacité du mélange de textures dynamiques (Mixture of Dynamic Textures (MDT)) dans la représentation d'une vidéo et le clustering des vidéos (Chan et Vasconcelos, 2008), (Mahadevan et al., 2010) proposent une approche non supervisée basée sur MDT pour représenter les situations normales dans les scènes de foule. Ce modèle considère les valeurs aberrantes spatio-temporelles comme des anomalies. Plus précisément, les anomalies temporelles représentent des événements rares, et les anomalies spatiales sont tout ce qui est saillant visuellement. Les métriques utilisées pour évaluer le modèle sur le jeu de données UCSD Anomaly sont la courbe ROC lorsque la comparaison entre la vérité terrain et l'anomalie détectée est effectuée au niveau de la trame et au niveau du pixel. (Mahadevan et al., 2010) utilisent aussi une autre métrique déduite de la courbe ROC, l'EER (Equal Error Rate). Cette dernière calcule le pourcentage de trames mal classées. Le MDT dépasse la méthode de la force sociale de (Mehran, Oyama et Shah, 2009), une méthode de surveillance du flux optique (Adam et al., 2008), la Mixture du Flux Optique (Kim et Grauman, 2009), et une combinaison entre les méthodes de (Mehran, Oyama et Shah, 2009) et de (Kim et Grauman, 2009) . L'inconvénient majeur de cette approche est sa forte dépendance aux données d'entraînement.

(Mehran, Oyama et Shah, 2009) utilisent le modèle de la force sociale pour détecter les anomalies dans les scènes de foule. Les auteurs projettent une grille de particules sur la première image d'une scène de foule, puis appliquent une opération d'advection de particules en suivant les fluctuations du flux optique. Un champ vectoriel des forces est extrait. Il s'appuie sur les lois du modèle de force sociale qui est appliqué sur les interactions qui se produisent entre les particules de la scène. À partir de ce champ vectoriel, un sac de mots est construit pour représenter les différents comportements d'une foule. L'allocation de Dirichlet latente (Latent Dirichlet Allocation (LDA)), une technique issue du traitement du langage naturel, est entraînée pour reconnaître les comportements normaux sur le sac de comportements. Enfin, la méthode Espérance-Maximisation (EM) est utilisée pour différencier les comportements normaux et anormaux. Un comportement anormal est observé dans les régions où se focalisent les flux de force. La méthode est évaluée, en utilisant la courbe ROC, sur les jeux de données UMN et Web Dataset. (Mehran, Oyama et Shah, 2009) démontrent que l'approche basée sur le modèle de force sociale surpasse les méthodes basées uniquement sur le flux optique. L'inconvénient majeur de cette approche est son incapacité à reconnaître de nouveaux exemples pris à partir d'un angle ou d'une position de caméra différent de l'angle de la position de la caméra des exemples sur lesquels elle a été entraînée.

(Hassner, Itcher et Kliper-Gross, 2012) proposent des descripteurs de flux violents (Violent Flows (ViF)) pour détecter la violence dans les scènes de foule. Ces descripteurs sont extraits des variations des amplitudes des vecteurs de flux optique. Les auteurs proposent d'utiliser une représentation sac-de-mots de type Bag-of-Features des scènes avec le descripteur ViF. Par la suite, ils entraînent une machine à vecteurs de support (SVM) à classer des scènes en scènes violentes ou non violentes. Afin d'évaluer la tâche de classification, ils utilisent les métriques d'accuracy moyenne (ACC) $\pm$ un écart-type (SD) et l'Aire Sous la Courbe ROC (AUC). (Hassner, Itcher et Kliper-Gross, 2012) ont développé un jeu de données *ad hoc* pour

la détection de violence sur lequel ils évaluent les effets du descripteur ViF et le comparent à d'autres descripteurs SOTA. Ici, le descripteur ViF s'avère très utile pour cette tâche de classification. Toutefois, comme le soulignent les auteurs, il n'est pas aussi précis sur le jeu de données Hockey Fights où il est supplanté par d'autres descripteurs faisant appel au Scale-Invariant Feature Transform (STIP) (Laptev, 2005).

(Singh, Patil et Omkar, 2018) ont développé un système de surveillance par drone (DSS) pour identifier les individus violents en temps réel grâce à un service de calcul dans le cloud. Dans un premier temps, le DSS détecte les piétons via le réseau Feature Pyramid Network (FPN) (Lin et al., 2017). Ensuite, ils estiment la posture des piétons grâce au réseau ScatterNet Hybrid Deep Learning (SHDL). Le réseau SHDL est composé d'une combinaison d'un réseau ScatterNet placé en front-end (Singh et Kingsbury, 2017) et d'un réseau de Régression placé en back-end. Le SHDL crée un squelette constitué de 14 noeuds pour chaque piéton pour modéliser sa posture. Ce squelette est ensuite utilisé pour distinguer 6 classes d'actions, dont 5 sont des classes d'actions violentes. Cette tâche de classification se fait à l'aide d'une Machine à Vecteurs de Support (SVM). Le DSS obtient de meilleures performances qu'une méthode issue de l'état-de-l'art proposée par (Penmetsa et al., 2014) en termes d'accuracy, en la dépassant de plus de 10% sur le jeu de données proposé par (Singh, Patil et Omkar, 2018) dénommé Aerial Violent Individual (AVI) dataset. Toutefois, il serait intéressant d'observer les performances de cette approche sur d'autres jeux de données publiquement disponibles.

(Ravanbakhsh et al., 2016) ont développé une approche non supervisée qui détecte des anomalies locales en combinant des informations de déplacement avec l'apparence. Les informations de déplacement et d'apparence sont obtenues à l'aide d'un Réseau de Neurones Convolutifs (CNN). Leur approche se déroule en trois étapes : à l'aide du réseau Binary Fully Convolutionnel Network (BFCN), ils extraient des cartes binaires des séquences envoyées en entrée. Ces cartes binaires sont utilisées pour calculer une distance similaire à la mesure de commotion (Mousavi et al., 2015), appelée le Temporal CNN Pattern (TCP). En combinant ces informations avec le flux optique, ils produisent des segments de mouvement raffinés. Le réseau BFCN est composé de couches de neurones convolutifs (Fully Convolutional Network (FCN)) et d'une couche de quantification binaire (Binary Quantization Layer (BQL)). Bien que les poids du FCN soient obtenus à partir d'un modèle AlexNet pré-entraîné (Krizhevsky, Sutskever et Hinton, 2012), les poids du BQL sont obtenus à partir d'une méthode de hachage. Ils ont évalué leurs méthodes selon deux modes : (1) capacité de détection d'anomalies au niveau des trames (frame-level), et (2) capacité de détection d'anomalies au niveau des pixels (pixel-level). Dans le premier mode, (Ravanbakhsh et al., 2016) s'assurent que le modèle arrive à bien identifier les trames anormales, tandis que dans le second mode, ils vérifient si le modèle localise l'anomalie avec précision au niveau de l'image. Les expériences ont été menées sur les jeux de données UMN SocialForce et UCSD Anomaly. Les mesures d'évaluation qui ont été employées sont la courbe de ROC (Receiver Operating Characteristics) et l'aire sous la courbe ROC (AUC). Le modèle rencontre parfois des difficultés à détecter des anomalies lorsqu'un objet est minuscule, partiellement dissimulé, ou arbore un mouvement inhabituel qui serait vu comme un mouvement normal dans certaines situations, par exemple ; une voiture qui aurait une vitesse similaire à celles des piétons alors qu'elle devrait se déplacer plus vite.

(Ramos et al., 2017) proposent une méta-heuristique pour détecter les anomalies dans les scènes de foule. Ils extraient le flux optique de la scène pour caractériser le déplacement des piétons. La méta-heuristique Artificial Bacteria Colony (ABC) est

utilisée pour trouver des régions d'intérêt (ROI) marquées par une forte affluence. Ils entraînent des cartes topologiques (Self-Organized Maps (SOM)) sur la population de ABC, son stock alimentaire, et ses centres de gravité pour déceler des événements inhabituels. La fonction de coût de la méta-heuristique se base sur une mesure de similarité entre les motifs de comportements. Au cours de la procédure de recherche de la solution optimale, il est à noter que la population de ABC est mise à jour à travers un processus de sélection naturelle dont la fonction de coût est en fonction du stock alimentaire. L'utilisation du flux optique pour déceler les zones de mouvement prémunit la méthode contre les changements brusques dûs au bruit blanc et les fluctuations de l'intensité lumineuse. Afin d'évaluer la solution optimale, des expériences ont été menées sur le jeu de données UMN SocialForce. Avec des quantités expérimentalement déterminées sur la taille de la population de bactéries et du nombre de neurones (pour la carte SOM), la solution optimale réalise ses calculs assez rapidement, traitant chaque image en $0,033$ secondes en moyenne. La solution optimale surpasse en termes d'aire sous la courbe ROC (AUC) la méthode de (Mehran, Oyama et Shah, 2009) de 18%. Toutefois, nous regrettons que (Ramos et al., 2017) ne comparent pas leur approche à des méthodes plus récentes. En guise de perspectives, les auteurs proposent d'utiliser le modèle de force sociale de (Mehran, Oyama et Shah, 2009) à la place du flux optique, et d'utiliser la méta-heuristique ABC dans l'optimisation des volumes spatio-temporels.

(Singh et al., 2020) proposent Aggregation of Ensembles (AOE), une agrégation de quatre méthodes de classification qui rassemblent des ensembles de trois Réseaux de Neurones Convolutifs (CNN) ajustés sur des jeux de données de mouvements de foule pour détecter par le vote majoritaire des anomalies sur des vidéos de scènes de foule. Une méthode de classification est appliquée sur chaque ensemble. Un ensemble de CNN est composé des 3 modèles pré-entraînés suivants : AlexNet (Krizhevsky, Sutskever et Hinton, 2012) pré-entraîné sur CIFAR-10, GoogleNet (Hu et al., 2018) et VGGNet (Simonyan et Zisserman, 2014), tous deux pré-entraînés sur ImageNet. Les CNN ont été utilisés comme extacteurs de caractéristiques qui sont envoyées en entrée de quatre méthodes de classification : un SVM linéaire, un SVM quadratique, un SVM cubique, et une fonction Softmax. Les caractéristiques sont extraites d'un lot d'images sélectionnées à partir d'une vidéo. Si plus de 10% des images d'un lot sont classées comme anormales, la vidéo est considérée comme anormale. L'ajustement des modèles, l'entraînement de l'aggrégateur AOE, et son évaluation ont été entrepris sur les jeux de données suivants UCSD Anomaly Detection et Avenue [7].

(Qasim et Bhatti, 2019) proposent un descripteur tridimensionnel des flux optiques extraits des vidéos. Dans leur approche, le calcul du flux optique s'opère sur 7 images consécutives tirées d'un clip vidéo. Par la suite, les 3 caractéristiques calculées sont :

1. la somme de l'amplitude du flux optique soumise seuillée,
2. l'entropie conjointe des amplitudes des flux optiques de 2 images consécutives. L'entropie conjointe permet de détecter les changements brusques (par exemple : une dispersion rapide des piétons),
3. la variance d'un cuboïde spatio-temporel obtenue à partir d'un registre des amplitudes du flux optique.

En plus de ce descripteur, une Machine à Vecteur de Support (SVM) est utilisée pour détecter les anomalies dans les vidéos. L'objectif de (Qasim et Bhatti, 2019) est de

7. http://www.cse.cuhk.edu.hk/leojia/projects/detectabnormal/dataset.html

proposer un bon compromis entre précision et calcul temps réel. Leur méthode est évaluée en termes d'accuracy sur le jeu de données UMN.

(Hao et al., 2019) proposent d'appliquer un filtre de Gabor sur des textures spatio-temporelles afin de détecter des comportements anormaux dans des scènes de foule. À partir d'un clip vidéo, un volume spatio-temporel (Spatio-Temporal Volume (STV)) est construit (Adelson et Bergen, 1985) duquel sont extraites des textures spatio-temporelles (Spatio-Temporal Texture (STT)). Les STT sont des tranches verticales ou horizontales de STV le long de l'axe temporel. Le filtre de Gabor est appliqué sur la STT pour supprimer le bruit et l'arrière plan. Parmi les STT filtrées, les auteurs sélectionnent la STT qui maximise l'entropie. Les caractéristiques de foule sont obtenues en appliquant une matrice de co-occurrence des niveaux de gris (GLCM) (Haralick, Shanmugam et Dinstein, 1973) sur les STT sélectionnées. Les STT sélectionnées sont d'abord converties en niveaux de gris, auxquelles un GLCM est appliqué. Par la suite, quatre caractéristiques sont extraites : le contraste, le deuxième moment angulaire, l'entropie, et la variance. Ces caractéristiques aident à modéliser une signature STT. Celle-ci est envoyée en entrée à des méthodes de classification. Dans ce travail, (Hao et al., 2019) comparent la qualité des signatures STT aux descripteurs de texture TAMURA (Ranjan et Agrawal, 2016). Chacune de ces descriptions est transmise en entrée des méthodes de classification suivantes : K plus proches voisins (KNN), la classification Naïve Bayésienne, l'Analyse Discriminante (DAC), la Forêt Aléatoire, et la Machine à Vecteurs de Support (SVM). La comparaison entre les caractéristiques TAMURA et GLCM, réalisée sur le jeu de données UMN, laisse entrevoir une certaine complémentarité entre les deux descripteurs de textures. Tandis que GLCM est utile pour détecter des situations de panique, TAMURA semble avoir un regard aiguisé pour détecter les situations de congestion.

(Lin et al., 2019) proposent un algorithme d'apprentissage multi-instances (Social Multiple Instance Learning (SMIL)) basé sur un réseau à deux branches pour détecter des anomalies dans des scènes de foule :

— La 1ère branche du réseau extrait des caractéristiques spatio-temporelles des clips en RVB. Cette branche est constituée d'un Réseau de Neurones Convolutifs 3D pré-entraîné sur les jeux de données Kinetics et UCF-101. Les caractéristiques extraites de ce premier réseau sont par la suite envoyées en entrée d'un module uni-dimentionnel d'attention connecté à un Réseau de Neurones entièrement connecté (Fully Connected Network (FCN)).

— La seconde branche extrait des caractéristiques des cartes de force sociale obtenues après l'application de l'approche de (Mehran, Oyama et Shah, 2009) sur les clips vidéos.

Leur approche a été entraînée en utilisant une fonction de coût basée sur le classement multi-instances (MIL Ranking). Ils ont évalué leur approche sur le jeu de données UCF Crime (Sultani, Chen et Shah, 2018) et l'ont comparée à 4 autres méthodes en termes des métriques ROC et l'Aire Sous la Courbe ROC (AUC).

(Xie, Zhang et Cai, 2019) proposent un algorithme de reconnaissance de comportements anormaux dans des scènes de foule basé sur le calcul du flux optique et l'utilisation du modèle de force sociale. Ils commencent par calculer le flux optique d'un clip vidéo en utilisant la méthode de Lucas-Kanade. Après cela, le flux optique est projeté sur un espace géospatial 2D en fonction des paramètres de la caméra. Cette opération les aide à extraire des points d'intérêt d'un clip vidéo. Par la suite, ils utilisent ces points de particules pour calculer les forces d'interactions sociales. Enfin, les forces de chaque trame sont additionnées et comparées à un seuil déterminé empiriquement pour décider si une trame est normale ou anormale.

La détection des anomalies a depuis toujours attiré l'attention des chercheurs en analyse de foule. Mais, comme il n'existe pas de définition de l'anomalie faisant l'unanimité en analyse de foule, aucun des travaux présentés dans cette section n'est généralisable à tous les contextes des mouvements de foule.

1.5 Les sources de données vidéo

Nous pouvons trouver deux types de sources de données vidéo :

1. des clips vidéo issus des jeux de données publics ou privés qui sont utilisés à des fins d'apprentissage, de validation, et de test,
2. des clips issus de la vidéo-surveillance en temps réel qui permettent de créer des jeux de données mais qui nécessitent un effort d'annotation.

Dans cette section, nous énumérerons quelques sources de données de vidéo-surveillance en direct.

1.5.1 Vidéo-surveillance en direct

Plusieurs sources de vidéo-surveillance en direct sont disponibles et librement accessibles sur Internet. Les scènes proposées peuvent être utilisées pour de nombreuses tâches telles que la détection et le suivi des piétons/véhicules, le comptage du nombre de personnes, ou l'analyse des comportements de foule. Nous listons ci-dessous quelques sources de vidéo-surveillance en direct que nous avons rencontrées au cours de nos recherches. Le Tableau 1.3 résume les caractéristiques de ces sources de données en temps réel.

— **Vidéo-surveillance du trafic routier au Royaume-Uni :** Le système de vidéo-protection britannique autorise le libre accès à des caméras contrôlant une cinquantaine de voies routières et auto-routières au Royaume-Uni. Les cadres fournis peuvent être utilisés principalement pour la détection de véhicules. Le suivi de trajectoires des véhicules ne semble pas possible car, une seule image au maximum est transmise chaque minute. Pour y remédier, (Lyu et al., 2017) ont mis en place UA-DETRAC [8], un jeu de données vidéo, dont la fréquence de trames est de plusieurs images par minute. Ce qui lui permet d'être utilisé pour des tâches de détection et de suivi de véhicules.

— **Earthcam :** Un site web qui fait office de répertoire de mono-caméras fixes qui filment plusieurs scènes des quatre coins du monde en temps réel. Ce répertoire nous offre une gamme de possibilités allant de l'analyse des comportements de foule, à la détection de piétons, au suivi de leurs trajectoires, et d'autres tâches liées au calcul des statistiques de foule.

— **La Mecque en direct :** La vidéo-surveillance en temps réel transmise librement par le système multi-caméras de la Mecque peut permettre de réaliser de multiples tâches en analyse de foule allant du calcul des statistiques de foule à l'analyse des comportements de foule. Des événements telles que la Omra ou le Hajj annuel offrent des occasions sans pareille pour évaluer des approches d'analyse de mouvements de foule massive. Les caméras, mises à disposition dans ce système, permettent d'accéder à différentes vues mais elles ne sont pas toujours fixes ce qui peut corser le travail de certaines tâches d'analyse des mouvements de foule. Ce système transmet des images en haute définition en temps réel.

8. UA-DETRAC : https://detrac-db.rit.albany.edu/

Référence bibliographique	Axe de recherche	Jeux de données utilisés	?DL	Code source
(Chan, Liang et Vasconcelos, 2008)	Comptage des personnes	UCSD Anomaly Detection	×	Closed source
(Shao, Change Loy et Wang, 2014)	Détection de groupes et analyse des comportements de groupe	CUHK Crowd	×	Closed source
(Ali et Shah, 2008)	Suivi et prédiction de trajectoires	Marathon-1, Marathon-1, Marathon-1, Train Station	×	Closed source
(Ali et Shah, 2007)	Analyse de trajectoires et détection d'anomalies	Self-made and Inside Mecca documentary	×	Open source
(Baccouche et al., 2011)	Reconnaissance d'actions humaines	KTH dataset	✓	Closed source
(Siva et Xiang, 2010)	Détection d'actions dans des scènes de foule	CMU and i-LIDS datasets	×	Closed source
(Hassner, Itcher et Kliper-Gross, 2012)	Détection d'actions violentes	Violent Flows (ViF)	×	Open source
(Mehran, Oyama et Shah, 2009)	Détection d'anomalies	UMN and Web datasets	×	Closed source
(Mahadevan et al., 2010)	Détection d'anomalies	UCSD anomaly dataset	×	Closed source
(Marsden et al., 2017)	Calcul des statistiques de foule, Détection d'anomalies	UMN, UCF CC 50, WWW Crowd	✓	Closed source
(Sindagi et Patel, 2017)	Comptage de personnes	UCF CC 50, ShanghaiTech	✓	Open source
(Tran et al., 2018)	Reconnaissance d'actions humaines	Sports-1m, Kinetics, HMDB-51, UCF-101	✓	Open source
(Carreira et Zisserman, 2017)	Reconnaissance d'actions humaines	HMDB-51, UCF-101	✓	Open source
(You et Jiang, 2018)	Reconnaissance d'actions dans des scènes peu denses	Self-made	✓	Closed source
(Bewley et al., 2016)	Suivi des trajectoires de piétons	MOT Challenge 2015	×	Open source
(Wojke, Bewley et Paulus, 2017)	Suivi des trajectoires et ré-identification de piétons	MOT Challenge 2016	✓	Open source
(Singh, Patil et Omkar, 2018)	Détection de comportements anormaux	Aerial Violent Individual (AVI) "Self-made"	✓	Closed source
(Ravanbakhsh et al., 2016)	Détection de comportements anormaux	UCSD, UMN	✓	Closed source
(Ramos et al., 2017)	Détection de comportements anormaux	UMN	×	Closed source
(Vahora et Chauhan, 2018)	Analyse des comportements de groupe	Collective activity	✓	Closed source
(Shu, Todorovic et Zhu, 2017)	Analyse des comportements de groupe	Collective activity, Volleyball	✓	Closed source
(Wei et al., 2020)	Reconnaissance des mouvements de foule	CUHK Crowd, normal-abnormal crowd	✓	Closed source
(Wang et O'Sullivan, 2016)	Reconnaissance des mouvements de foule	Synthetic, Edinburgh (Forum), MIT Carpark, Train Station	×	Closed source
(Yan, Zhu et Yu, 2019)	Reconnaissance des mouvements de foule	WorldExpo'10	✓	Closed source
(Ullah et al., 2016)	Reconnaissance des mouvements de foule	WorldExpo'10	×	Closed source
(Ullah et al., 2019)	Reconnaissance des mouvements de foule	CUHK Crowd	✓	Closed source
(Liu, Salzmann et Fua, 2019)	Calcul des statistiques de foule	ShanghaiTech, WorldExpo'10, UCF Crowd Counting, UCF QNRF, Venice dataset	✓	Closed source
(Wan et Chan, 2019)	Calcul des statistiques de foule	ShanghaiTech A/B, WorldExpo'10, UCF QNRF	✓	Closed source
(Wang et al., 2019)	Calcul des statistiques de foule	ShanghaiTech A/B, WorldExpo'10, UCF QNRF, Grand Theft Auto 5 dataset	✓	Open source
(Lamba et Nain, 2019)	Analyse des mouvements	UCF Crowd, Collective motion, Violent Flows	×	Closed source
(Li et al., 2019)	Analyse des mouvements	Self-made (canteen scene)	×	Closed source
(Wu et al., 2017b)	Analyse des mouvements	UCF Crowd, CUHK Crowd	×	Open source
(Alahi et al., 2016)	Prédiction des trajectoires	UCY and ETH	✓	Open source
(Bartoli et al., 2017)	Prédiction des trajectoires	UCY and Museum dataset	✓	Closed source
(Zitouni, Sluzek et Bhaskar, 2020)	Analyse des comportements de groupe	PETS dataset, Parking Lot, Town Center	×	Closed source
(Bisagno, Zhang et Conci, 2018)	Analyse des comportements de groupe	UCY and ETH	✓	Closed source
(Singh et al., 2020)	Détection de comportements anormaux	UCSD Ped 1, UCSD Ped 2, and the Avenue dataset	✓	Closed source
(Qasim et Bhatti, 2019)	Détection de comportements anormaux	UMN	×	Closed source
(Hao et al., 2019)	Détection de comportements anormaux	UMN	×	Closed source
(Lin et al., 2019)	Détection de comportements anormaux	UCF Crime dataset	✓	Closed source
(Xie, Zhang et Cai, 2019)	Détection de comportements anormaux	Self-made	×	Closed source

TABLE 1.2 – Présentation synthétique des travaux vus en analyse de foule. La colonne ?DL précise si le travail de recherche emploie ou non des méthodes issues du Deep Learning (✓) ou non (×)

Vidéo-surveillance en direct	Qualité de l'enregistrement	Caméras utilisées	Tâches d'analyse de foule	Site web
UK road traffic	Bonne résolution, 1 image/minute	Statique, 50 vues	Détection et peut être le suivi de véhicules	https://trafficcameras.uk/roads/
Earthcam	Bonne résolution, fréquence d'images en temps réel	Statique	Toutes les tâches liées à l'analyse de foule	https://www.earthcam.com/
Mecque en direct	720p	Statique et dynamique	Analyse des mouvements de foule massive	https://makkahlive.net/tv.aspx?r=14
Vatican en direct	480p	Statique, une seule vue	Toutes les tâches liées à l'analyse de foule	https://www.youtube.com/watch?v=q5wc5aIpjk4
Place centrale de Monthey	Résolution moyenne, ~5 images/seconde	Statique, 2 vues	Détection et suivi de trajectoires des piétons et des véhicules	http://www.idelec.ch/webcam/monthey/cam1 ou (.../cam2)

TABLE 1.3 – Présentation synthétique des systèmes de vidéo-protection en direct publiquement accessibles.

— **Vatican en direct :** Positionnement d'une mono-caméra piéton fixe observant la place Saint-Pierre. Cette place accueille souvent des rassemblements de diverses densités.
— **Monthey Place Centrale :** Cette place est vidéo-protégée par deux caméras qui émettent des images d'une résolution moyenne toutes les secondes. Les images sont affectées par des changements d'intensité lumineuse et certaines conditions météorologiques pouvant bruiter la qualité des images fournies. Les scènes filmées fournissent également divers défis aux tâches de détection et de suivi des trajectoires des piétons ou de véhicules tels que l'occlusion, et parfois, la congestion.

1.5.2 Jeux de données

Nous n'avons pas l'intention de dresser une liste exhaustive des jeux de données existants [9]. Une grande partie des jeux de données que nous introduisons, dans cette section, sont fréquemment utilisés pour des tâches liées à l'analyse de foule telles que la détection et le suivi des trajectoires des piétons avec KITTI et MotChallenge, la classification et la reconnaissance d'actions avec UCF-101 et HMDB-51, la détection de piétons seulement avec Inria Person et Caltech Pedestrian. Les jeux de données utiles pour les domaines centraux de l'analyse de foule telles que la détection d'actions dans des scènes de foule, la reconnaissance des comportements de foule sont rares à trouver. Les Tableaux 1.4 et 1.5 résument le contenu de cette section 1.5.2.

1.5.2.1 Jeux de données pour le calcul des statistiques de foule

Multi Task Crowd : Ce jeu de données a été développé par (MARSDEN et al., 2017). Ils l'ont utilisé pour entraîner l'architecture ResnetCrowd sur les trois tâches suivantes : le comptage des personnes dans une foule, l'estimation de la densité d'une scène de foule, et la détection d'actions violentes. Le jeu de données est composé d'un ensemble d'images obtenues à partir du jeu de données WWW Crowd [10].

UCF Crowd 50 : Utilisé pour le comptage de personnes dans une foule, UCF Crowd 50 [11] est constitué d'une cinquantaine d'images. Les images sont tirées de FLICKR [12], et sont occupées par une foule dont la taille varie entre 94 et 4543 individus.

GTA 5 Crowd Counting (GCC) : Proposé par (WANG et al., 2019), ce jeu de données synthétiques est créé à l'aide de Script Hook V [13]. Script Hook V est une librairie C++ appliquée au jeu vidéo Grand Theft Auto 5 (GTA 5). Les images proviennent d'une centaine de scènes d'intérieur et d'extérieur de Los Santos, une

9. Par souci d'exhaustivité, nous vous invitons à explorer ces trois sites web :
— http://homepages.inf.ed.ac.uk/rbf/CVonline/Imagedbase.htm
— http://riemenschneider.hayko.at/vision/dataset/index.php
— https://www.di.ens.fr/~miech/datasetviz/

10. WWW Crowd : https://computervisiononline.com/dataset/1105138602
11. UCF Crowd 50 : http://crcv.ucf.edu/data/ucf-cc-50/
12. FLICKR : https://www.flickr.com/
13. Script hook v. http://www.dev-c.com/gtav/scripthookv/

ville fictive inspirée de la ville de Los Angeles, en Californie, un état des États-Unis d'Amérique. Les personnes dans les scènes sont diverses et variées, et générées à partir de 265 modèles de personnes différentes. Les scènes dans le jeu GTA 5 sont limitées à une capacité d'accueil de 256 personnes. Le jeu de données GCC contient 15212 images. La définition de chaque image est de 1080×1920. Le jeu de données offre pas mal de défis au niveau météorologique au vu des 7 conditions météorologiques qui peuvent être observées : temps clair ou très ensoleillé, pluvieux, modérément nuageux ou très couvert, brumeux. Les scènes peuvent être capturées à différents instants d'une journée.

1.5.2.2 Jeux de données pour l'analyse et la prédiction des trajectoires

Town Center : proposé par (Benfold et Reid, 2011), ce jeu de données est une vidéo enregistrée à partir d'une caméra de vidéo-protection qui filme des piétons marchant au centre-ville. Le jeu de données est principalement utilisé pour le suivi de trajectoires des piétons. A chaque trame, il y a environ 16 piétons. La qualité de la vidéo est bonne. Sa définition est de 1920×1080 pixels pour 25 images/seconde.

i-Lids for AVSS 2007 : Le jeu de données i-Lids [14] contient deux types d'enregistrements vidéo : le premier provient d'une station de métro et le second provient d'un trafic routier. Ces vidéos peuvent être utilisées pour la détection et le suivi de véhicules, et la détection et le suivi des trajectoires des piétons. Elles sont également employées pour la détection des comportements anormaux (par exemple : l'abandon d'un bagage dans une station de métro). La définition des images de la vidéos est de 720×576 pixels pour 25 Hertz.

MOTChallenge Dataset : MOTChallenge 2016 Dataset [15] est une extension de la version précédente de 2015 proposée par (Leal-Taixé et al., 2015) (Milan et al., 2016). Il se compose de 14 séquences, contenant des scénarios de scènes de foule semblables aux scènes de PETS09-S2L1. Le jeu de données peut représenter un défi pour des méthodes de détection de piétons et de suivi de leurs trajectoires au vu des conditions de capture des vidéos qu'il met en place : mobilité de certaines caméras, changement d'intensité lumineuse, variété des angles de vue. Les éléments présents dans la scène peuvent aller des piétons mobiles ou assis, à des véhicules, et à des éléments obstructifs, etc. Les auteurs du jeu de données incitent les concepteurs de méthodes de détection et de suivi de trajectoires qui travaillent sur le jeu de données MOT Challenge à évaluer leurs méthodes selon les métriques CLEAR expliquées par (Kasturi et al., 2009), en plus d'autres métriques telles que celles évoquées dans les travaux de (Huang, Nevatia et Li, 2009).

KITTI : Les vidéos du jeu de données KITTI [16] ont été obtenues à partir d'une station de VM en mouvement wagon qui enregistrait pendant 6 heures à 10-100 Hz, dans la ville de Karlsruhe en Allemagne (Geiger et al., 2013). Plusieurs capteurs ont été utilisés pour l'enregistrement de ces scènes tels qu'une caméra stéréo couleur et en niveaux de gris, et un scanner laser 3D Velodyne. Les images en couleur et en niveaux de gris sont stockées au format de fichiers PNG 8 bits. Le jeu de données est utilisé, entre autres, pour la détection et le suivi d'objets. Huit classes d'objets peuvent être suivies et détectées dans des conditions difficiles, car les objets peuvent être statiques ou dynamiques et la caméra bouge constamment. Les 8 classes d'objets pouvant être observées sont : des voitures, des vans, des camions, des piétons, des individus assis sur des bancs, des cyclistes, des tramways, et le reste des objets non-identifiés sont

14. i-Lids : http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html
15. MOT Challenge : https://motchallenge.net/
16. KITTI : http://www.cvlibs.net/datasets/kitti/index.php

inclus dans la classe Divers. Les annotations fournies se présentent sous la forme de tracklets de Bounding Boxes 3D. La taille totale des données fournies est de 180 Go.

Toulouse Campus Surveillance Dataset : Le jeu de données Toulouse Campus Surveillance Dataset [17] peut être utilisé, entre autres, pour la détection/le suivi d'objets et la reconnaissance d'événements audio (Malon et al., 2018). Le jeu de données provient d'un système multi-caméras et peut donc être utile pour évaluer des méthodes fusion d'informations provenant de caméras ciblant les mêmes scènes. Il peut aussi servir à la ré-identification des piétons. Certaines caméras sont fixes, d'autres sont mobiles. Le système a permis de générer 50 vidéos réparties en deux scénarios. Chaque vidéo est disponible en trois définitions différentes : 1920×1080, 960×540, 640×360. Les annotations de la détection et le suivi de trajectoires ne sont fournies que pour le premier scénario.

The PathTrack MOT : À l'aide du crowdsourcing permis par Amazon Mechanical Turk, (Manen et al., 2017) ont pu proposer le jeu de données annoté PathTrack MOT [18]. Ce jeu de données a été généré suite à la création de l'annotateur PathTrack. Il fournit l'annotation de 720 séquences vidéo qui mettent en scène les mouvements de piétons à partir de scènes de vidéo-protection capturées par des caméras statiques et dynamiques. Les séquences contiennent 16287 trajectoires annotées.

Optical Flow Dataset : Également nommé TUB CrowdFlow Dataset, le Optical Flow Dataset [19] a été créé par (Schröder et al., 2018). Les auteurs ont généré graphiquement le jeu de données en utilisant le moteur graphique Unreal Engine. Ils simulent des mouvements de foule dans cinq scénarios différents. La foule y est capturée par deux caméras statique et dynamique. Le jeu de données contient 10 séquences dont les longueurs varient de 300 à 450 images. Les séquences sont caractérisées par une fréquence d'images de 25 Hz et une résolution Haute-Définition. Les scènes peuvent contenir de 371 à 1451 individus. Les auteurs ont également vérifié que les résultats obtenus par les méthodes état-de-l'art sur ce jeu de données peuvent être transférés vers des jeux de données obtenus du monde réel tels que le UCF Crowd Tracking de (Ali et Shah, 2008). Dans le jeu de données, des annotations vérité-terrain sont fournies. Ces annotations renseignent sur le flux optique, et les trajectoires denses et éparses des individus.

UCF Crowd Tracking : Le jeu de données UCF Crowd Tracking [20] a été créé par (Ali et Shah, 2008). Le jeu de données est constitué de trois séquences appelées Marathon-1, Marathon-2 et Marathon-3. La longueur de ces séquences varie de 333 à 492 images. Pour évaluer leurs méthodes, les auteurs ont annoté les trajectoires de 199, 120, et 50 individus respectivement pour chacune des trois séquences.

1.5.2.3 Jeux de données pour la détection de piétons

Inria Person : Le jeu de données Inria Person [21] est utilisé pour la détection de piétons. Le jeu de données contient des annotations sous la forme de boîtes englobantes encadrant les piétons. Les auteurs mettent également à disposition des images positives (boîtes englobantes) de piétons redimensionnées à la définition 64×128 pixels, ainsi que des images originales négatives. Le jeu de données contient 1805 images de personnes figurant dans des situations pouvant représenter un certain défi

17. Toulouse Campus Surveillance Dataset : http://ubee.enseeiht.fr/dokuwiki/doku.php?id=public:tocada
18. PathTrack MOT : https://data.vision.ee.ethz.ch/daid/MOT/pathtrack_release_v1.0.zip
19. Optical Flow Dataset : https://github.com/tsenst/CrowdFlow
20. UCF Crowd Tracking : https://www.crcv.ucf.edu/data/tracking.php
21. Inria Person : http://pascal.inrialpes.fr/data/human/

pour les algorithmes de détection de piétons : différentes postures et orientations du piéton ainsi qu'un large éventail d'arrière-plans (Dalal et Triggs, 2005).

Caltech Pedestrian : Le jeu de données Caltech Pedestrian [22] est couramment utilisé pour la détection des piétons (Dollar et al., 2012). Il peut également être utilisé pour entraîner des méthodes de détection de piétons à contrer les occlusions. Le jeu de données contient 10 heures de capture vidéo de 640×480 pixels enregistrées à partir d'une mono-caméra autoportée sur un véhicule se déplaçant dans la rue. Ces 10 heures de capture vidéo ont permis de générer 250.000 images. 2300 piétons différents apparaissent dans le jeu de données. Les occurrences de ces piétons ont été annotées sous la forme de 350000 boîtes englobantes.

Common Object in Context (COCO) : (Lin et al., 2014) proposent le jeu de données COCO [23]. Il est principalement utilisé pour la détection d'objets , des KeyPoints, et la segmentation d'images. En plus des piétons, 79 autres classes d'objets figurent dans les images du jeu de données. Il contient 330000 images, et plus de 200000 d'entre elles sont annotées.

1.5.2.4 Jeux de données pour la reconnaissance d'actions dans des scènes individuelles

UCF-101 : (Soomro, Zamir et Shah, 2012) proposent UCF-101 [24], une extension du jeu de données UCF-50. Le jeu de données se compose de 13320 clips qui représentent 27 heures d'enregistrement vidéo, téléchargées de Youtube [25]. La définition des images de chaque clip vidéo est de 320×240 pixels. Il dispose d'une fréquence de 25 images/seconde. Les clips sont catégorisés en 101 classes d'actions. Les clips sont sujets à l'obstruction de l'arrière-plan, à des mouvements de la caméra, aux changements d'éclairage, aux occlusions partielles des objets, et à des images de mauvaise qualité.

HMDB-51 : (Kuehne et al., 2011) proposent HMDB-51 [26]. Le jeu de données se compose de 6766 clips vidéo annotés récupérés de Youtube ou de diverses productions cinématographiques. Les clips vidéo ont tous été normalisés à une hauteur de 240 pixels. La largeur de chaque clip vidéo est redimensionnée de manière à conserver un ratio hauteur/largeur qui préserve la qualité de la vidéo. La fréquence de tous les clips est de 30 images/seconde. Les clips sont catégorisés en 51 classes d'actions. Les méta-informations du jeu de données dressent pour chaque clip vidéo une liste de renseignements telle que l'angle de prise de vue de la caméra, l'occurrence d'occlusions, la présence d'un mouvement de la caméra (qui affecte les 2/3 des clips du jeu de données), la qualité de la vidéo (très bonne qualité, qualité moyenne, ou faible qualité), ainsi que le nombre d'individus figurant sur le clip vidéo.

Kinetics : (Kay et al., 2017) proposent le jeu de données Kinetics [27]. Ce jeu de données similaire à UCF-101 et HMDB-51. Il est principalement utilisé pour la classification des clips vidéo. Il se compose de 400 classes d'actions. Chaque classe peut contenir de 400 à 1150 clips vidéo. Chaque clip vidéo est tiré de Youtube et dure en moyenne une dizaine de secondes. Les défis présentés par ce jeu de données sont les changements d'éclairage, l'encombrement de l'arrière-plan, les vibrations et la

22. Caltech Pedestrian : http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/
23. COCO : http://cocodataset.org/#detection-2018
24. UCF-101 : http://crcv.ucf.edu/data/UCF101.php
25. Youtube : https://www.youtube.com/
26. HMDB : https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/
27. Kinetics : https://deepmind.com/research/open-source/open-source-datasets/kinetics/

mobilité de la caméra, les ombres, les reflets, etc. Le processus d'annotation a reposé sur le service Amazon Mechanical Turk (MTurk) [28]. Au moins chaque clip contient le nom d'une action qui s'y produit, mais certains clips peuvent se retrouver associés avec plusieurs actions qui s'y sont déroulées.

Sports-1m : (Karpathy et al., 2014) proposent le jeu de données Sports-1m [29]. Ce jeu de données contient 1 million de vidéos Youtube qui durent en moyenne 5 minutes et 36 secondes. Comme le nom du jeu de données le laisse entendre, les vidéos portent uniquement sur des activités sportives. Il y a, dans le jeu de données, 487 classes d'activités différentes. Chaque classe est illustrée par 1000 à 3000 clips vidéo, et approximativement 5% des clips vidéo possèdent plus d'une étiquette. Cela veut dire que sur certains clips plusieurs activités différentes ont pu être réalisées.

1.5.2.5 Jeux de données pour la détection d'anomalies

Aerial Violent Individual (AVI) : Proposé par (Singh, Patil et Omkar, 2018) pour entraîner leur réseau ScatterNet Hybrid Deep Learning (SHDL) pour l'estimation de la posture d'une personne, ce jeu de données contient 2000 images. Les scènes capturées ne sont pas denses et peuvent contenir de 2 à 10 personnes. Dans le jeu de données, chaque individu est annoté avec 14 KeyPoints permettant une analyse précise de sa posture. Les images ont été capturées par un drone selon quatre altitudes différentes. Ce jeu de données ne semble pas publiquement disponible pour le moment.

UMN SocialForce et Web Dataset [30] : UMN SocialForce se compose de 11 séquences vidéo qui décrivent toutes une situation de départ ordinaire qui se termine mal. Le jeu de données Web se compose de 20 séquences vidéo. 8 de ces séquences contiennent des événements anormaux tels que les situations de panique ou des affrontements, et 12 de ces séquences décrivent une situation ordinaire de piétons qui marchent sans manifester le moindre signe de panique.

UCSD Anomaly Detection : UCSD Anomaly Dataset [31] est couramment utilisé pour la détection d'anomalies et il se compose d'approximativement 100 clips vidéo. Ces vidéos sont divisées en deux sous-ensembles de données : Peds1 et Peds2 (Chan, Liang et Vasconcelos, 2008). Les anomalies sont liées à des éléments anormaux apparaissant dans le clip vidéo comme l'apparition de véhicules dans une voie piétonne. Cependant, les anomalies concernent également les piétons qui décrivent des mouvements inhabituels. La vérité-terrain indique l'occurrence d'une anomalie sur la trame via un flag binaire. La vérité-terrain fournit également la localisation de la boîte englobante de l'élément anormal.

Agoraset : Créé par (Allain, Courty et Corpetti, 2012), ce jeu de données peut être utilisé pour le suivi de la trajectoire des piétons, l'analyse des événements anormaux, et l'estimation de la densité d'une scène de foule. Les simulations ont été générées à l'aide d'un modèle basé sur les forces Lagrangiennes proposées par (Helbing, Farkas et Vicsek, 2000). Le jeu de données se compose de 7 scénarios qui peuvent être déclinés en plusieurs versions selon les modifications qu'il est possible d'y apporter : variation de l'état de la foule (mouvement ordinaire ou situation de panique), l'intensité de la luminosité, l'ajout des ombres, etc.

28. Amazon Mechanical Turk : https://www.mturk.com/
29. Sports-1m : https://cs.stanford.edu/people/karpathy/deepvideo/
30. UMN SocialForce et Web Dataset : http://crcv.ucf.edu/projects/Abnormal_Crowd/
31. UCSD Anomaly Detection : http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm

Violent Flows (ViF) : (Hassner, Itcher et Kliper-Gross, 2012) ont proposé The Violent Flows Dataset [32] en réponse à la rareté de jeux de données illustrant des comportements violents, Le jeu de données est composé de 246 clips vidéo provenant de Youtube. Les images des vidéos ont été redimensionnées à 320×240 pixels. La durée moyenne de chaque clip vidéo est de 3.60 secondes. (Hassner, Itcher et Kliper-Gross, 2012) ont rendu disponible un code source pour calculer les descripteurs de violence qui est publiquement disponible sur Github [33].

Caviar : Le jeu de données Caviar [34] peut être utilisé pour la reconnaissance d'anomalies dans des scènes de foule. Les scènes disponibles dans le jeu de données sont légèrement denses. L'un des défauts du jeu de données est que toutes les scènes sont scriptées (ED, 2003). Les séquences se déclinent en deux scénarios en fonction du lieu de leur tournage :

— une scène intérieure provenant du Hall d'entrée de l'Inria Labs de Grenoble et,
— Un centre commercial du Portugal.

De la première scène, 28 clips vidéo ont été produits à l'aide d'une seule caméra. De la seconde scène, 26 clips vidéo ont été produits à l'aide de deux caméras permettant d'observer une même scène de deux points de vue différents.

Motion Emotion : Le jeu de données Motion Emotion [35] a été créé par (Rabiee et al., 2016b ; Rabiee et al., 2016a), et est utilisé pour la détection d'anomalies dans les mouvements et les émotions humaines. Le jeu de données se compose d'approximativement 44000 images qui composent 31 vidéos. Ici l'annotation du jeu de données ne se fait pas au niveau des clips vidéo mais au niveau des images. Ces images sont classées en 5 types de mouvements : la panique, les accrochages, l'évitement d'obstacle, la congestion, et des comportements normaux. Les images décrivent également 6 types d'émotions de foule Celles-ci décrivent : la colère, la joie, l'excitation, la peur, la tristesse, et l'indifférence (ou la neutralité). Les scènes sont enregistrées à partir d'une caméra orientée vers le bas pour enregistrer des personnes qui marchent. La définition des images des vidéos est de 554×235 pixels. La fréquence des vidéos est de 30 images/seconde. Les scènes sont toujours occupées par une foule très dense ou faiblement disparate.

UCF-Crime : Le jeu de données UCF-Crime [36] a été créé par (Sultani, Chen et Shah, 2018). Le jeu de données se compose de 1900 vidéos de surveillance, longues et non découpées, totalisant 128 heures d'enregistrement acquises à partir de Youtube et Liveleak [37] via des requêtes textuelles exprimées en différentes langues. UCF-Crime peut être utilisé pour deux tâches : la détection d'anomalies car nous pouvons trouver des comportements normaux et anormaux ; et la classification des activités anormales, car les comportements anormaux peuvent être classés en 13 activités anormales qui sont : l'abus de faiblesse, les arrestations, les incendies criminels, les agressions, les accidents de la route, les cambriolages, les explosions, les accrochages, les vols violents, les tirs à l'arme à feu, les vols, les vols à l'étalage, et le vandalisme. Lorsque les auteurs ont évalué les méthodes sur ce jeu de données, ils ont pu fixer la fréquence à 30 images/seconde et la définition des images à 240×320 pixels.

CCTV-Fights : Le jeu de données CCTV-Fights [38] a été créé par (Perez, Kot

32. Violent Flows (ViF) : https://www.openu.ac.il/home/hassner/data/violentflows/
33. Descripteurs de violence : https://talhassner.github.io/home/projects/violentflows/index.html
34. Caviar : http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/
35. Motion Emotion Dataset (MED) : https://github.com/hosseinm/med
36. UCF-Crime : https://webpages.uncc.edu/cchen62/dataset.html
37. https://www.liveleak.com/
38. CCTV-Fights : https://rose1.ntu.edu.sg/dataset/cctvFights/

et Rocha, 2019). Le jeu de données se compose de 1000 clips vidéo annotés temporellement qui ont nécessité plus de 17 heures d'enregistrement vidéo. Ces vidéos sont constituées de 280 clips vidéo, dont la durée varie entre 5 secondes à 12 minutes, et de 720 clips vidéo qui ne proviennent pas des systèmes de vidéo-protection (Non-CCTV [39]), dont la durée varie entre 3 secondes et 7 minutes. Les définitions et les résolutions de ces vidéos sont variables.

Crowd-11 : Le jeu de données Crowd-11 a été créé par (Dupont, Tobias et Luvison, 2017). Il se compose de 6272 séquences vidéo capturées. En moyenne, chaque clip vidéo est composé de 100 images chacun. Le clip vidéo dure au maximum 5 secondes. La résolution des vidéos est variable. Le jeu de données est principalement destiné à la classification des comportements de foule et à la détection de la violence. Les séquences peuvent être classées en 11 catégories, à savoir : Pas de foule, Flux laminaire, Foule turbulente, Deux flux de foule qui se croisent, Deux flux de foule qui convergent, Deux flux de foule qui divergent, Électrons libres, Foule congestionnée, Foule statique mais agitée, Deux flux de foule en conflit. Les clips vidéo de ce jeu de données proviennent de deux types de sources de vidéos :

— Les sites web de partage de vidéos tels que Youtube, Pond5 et Gettyimages.
— D'autres jeux de données qui sont : Agoraset, UMN SocialForce, Violent Flows, CUHK Crowd, WWW Crowd Attributes, Shanghai WorldExpo'10 Crowd, Hockey combats et films, PETS-2009

Le jeu de données est faiblement annoté. En effet, la seule information dont nous disposons est la classe de chaque clip vidéo. Dans le cadre de la classification des vidéos de foule, ce jeu de données est assez riche en vidéos et varié en classes pour mettre en place des méthodes récentes issues du Deep Learning.

1.5.2.6 Jeux de données pour la détection de groupes et pour l'analyse des comportements de groupes

The Friends Meet (FM) : (Bazzani, Cristani et Murino, 2012) proposent le jeu de données The Friends Meet [40] pour évaluer l'approche Decentralized Particles Filter qu'ils ont utilisée pour le suivi simultané des trajectoires des piétons et des groupes de piétons. The Friends Meet permet également d'évaluer des méthodes de détection de piétons et des groupes de piétons. Le jeu de données contient 53 séquences composées de 16286 images, impliquant 3 à 16 individus par scène. Il y a deux types de séquences :

— 28 séquences synthétiques, constituées de 200 images chacune.
— Des vidéos tirées du monde réel capturées dans des scènes d'extérieur.

Le jeu de données décrit plusieurs situations de formations de groupes : apparition ou disparition d'un groupe de la scène, division ou fusion d'un groupe, et un groupe sous la forme d'une file d'attente.

CUHK Crowd Dataset : Créé par (Shao, Change Loy et Wang, 2014 ; Shao, Loy et Wang, 2017), le jeu de données CUHK Crowd [41] contient 474 clips vidéo représentant 215 scènes de foule. La durée des clips n'est pas homogène. Le jeu de données est principalement utilisé pour les tâches suivantes : regroupement des piétons en groupes ou segmentation de la foule en petites groupes. Le jeu de données est également utilisé dans l'analyse des comportements de groupe, pour la détection d'anomalies, et pour le calcul des statistiques de foule. Des annotations ont

39. Closed-Circuit TeleVision (CCTV) (Perez, Kot et Rocha, 2019)
40. The Friends Meet : https://pavis.iit.it/datasets/fmdataset
41. CUHK Crowd : https://amandajshao.github.io/projects/CUHKcrowd_files/cuhk_crowd_dataset.htm

été fournies par les auteurs pour localiser des groupes, décrire le comportement de chaque groupe, et apporter une classe pour chaque clip vidéo.

MuseumVisitors : Le jeu de données MuseumVisitors [42], créé par (Bartoli et al., 2015a), a été enregistré par un système de trois caméras IP disposant d'une définition de 1280×800 pixels d'une fréquence d'enregistrement de 5 images/seconde, à l'intérieur du Musée national du Bargello à Florence, en Italie. Ce jeu de données peut représenter un défi pour un bon nombre d'algorithmes de détection de piétons et de groupes. Il est caractérisé par un nombre fréquent d'occlusions et de changements d'intensité lumineuse. Le positionnement des trois caméras induit à la variation de l'échelle des objets apparaissant dans la scène. Ce jeu de données est destiné à la détection de piétons et de groupes, l'estimation du sens du regard, l'analyse des comportements de groupe et de piétons, et la ré-identification des piétons. Ce jeu de données contient deux scénarios :

1. Des visiteurs seuls qui observent des œuvres d'art.
2. Des groupes de visiteurs qui observent des œuvres d'art.

Les annotations fournies sont des boîtes englobantes autour des visiteurs. Si une personne est partiellement dissimulée, sa partie visible est encadrée d'une deuxième boîte englobante. L'annotation fournit également un identifiant unique pour chaque groupe et chaque piéton sur toutes les boîtes englobantes.

Behave : (Blunsden et Fisher, 2010) ont trouvé que la plupart des jeux de données disponibles sont utilisés pour la détection et le suivi de trajectoires de piétons, et la reconnaissance d'actions dans des scènes individuelles. Constatant un manque de jeu de données décrivant des comportements de groupe normaux ou anormaux, ils ont décidé de créer le jeu de données Behave [43]. Ce jeu de données est constitué de 4 clips vidéo enregistrés sous le format WMV, qui peuvent se décliner en 76800 images. La définition des images des clips vidéo est de 640×480 pixels, et chaque vidéo dispose d'une fréquence de 25 images/seconde. Dans tout le jeu de données, il y a 125 personnes impliquées dans la réalisation des actions. L'annotation du jeu de données consiste en un encadrement des piétons par des boîtes englobantes. À ce titre, les auteurs ont généré 83545 boîtes englobantes à l'aide de l'outil Viper-GT [44]. Par ailleurs, les interactions inter et intra-groupes ont été classées en 10 catégories. Les défis offerts par ce jeu de données sont les changements de luminosité et les occlusions récurrentes.

SALSA : (Alameda-Pineda et al., 2016) proposent le jeu de données SALSA [45] pour la détection de groupes et à l'analyse des comportements de groupe. Le but premier des auteurs est de permettre l'étude des groupes conversationnels autonomes (Free-standing Conversational Groups (FCG)). Dans le jeu de données, deux vidéos sont capturées dans une scène d'intérieur. Dans chacune d'elles, 18 personnes sont impliquées. Leurs comportements ne sont pas scriptés. Les vidéos durent en moyenne une soixantaine de minutes. Les défis présentés par ce jeu de données sont la faible résolution, les variations de l'intensité lumineuse, et les occlusions qui surviennent très souvent. Les annotations fournies renseignent sur la personnalité de chaque individu, sa position, l'orientation de la tête et de son corps, à quel groupe appartient-il, et quel type de formation ce groupe adopte. Les annotations sur la personnalité d'un individu lui attribue un score d'extraversion, d'amabilité, de stabilité émotionnelle, et de créativité. Ces scores ont été obtenus en se basant sur les réponses apportées

42. MuseumVisitors : https://www.micc.unifi.it/resources/datasets/museumvisitors/
43. Behave : http://groups.inf.ed.ac.uk/vision/BEHAVEDATA/INTERACTIONS/
44. Viper-GT : http://viper-toolkit.sourceforge.net/
45. SALSA : http://tev.fbk.eu/salsa

à un questionnaire de personnalité Big Five (John et Srivastava, 1999) adressé à chaque participant avant d'entamer les enregistrements.

1.5.3 Conclusion sur les sources de données existantes

Comme vous pouvez l'observer à partir de la palette de jeux de données présentés dans cette section, la plupart des jeux de données publiquement disponibles sont utilisés pour l'une des tâches suivantes : la reconnaissance d'actions dans des scènes individuelles, la détection et le suivi des trajectoires des piétons, la détection et le suivi des trajectoires des groupes, le comptage du nombre de personnes dans une foule et l'estimation de sa densité, des cas limités de classification de mouvements de foule et de détection d'anomalies, et très peu de cas pour la reconnaissance des activités de groupe. Cependant, il y a un manque de jeu de données qui illustrent les mouvements de foule dans les rues ou les avenues des grandes villes. À notre connaissance, il n'existe aucun jeu de données pouvant être utilisé pour l'analyse et la prédiction des comportements de foule.

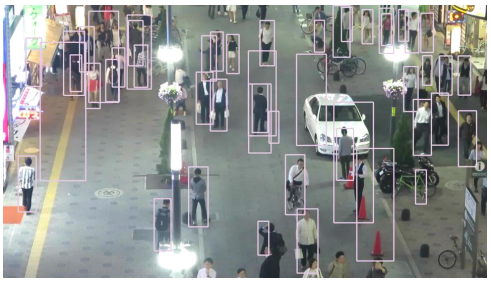
L'accès aux données vidéo, rendu possible par la disponibilité d'une multitude de systèmes de vidéo-protection émettant en temps réel (c.f. Section 1.5.1), étend nos possibilités. En effet, les rassemblements se produisent plusieurs fois par jour aux quatre coins du monde. Parfois avec un fort Niveau de Service (LoS) autour de la Kaaba, à la Mecque en Arabie Saoudite, devant le Mur des Lamentations, à Jérusalem en Israël (Earthcam), à Times Square, à New York, ou parfois avec des Niveaux de Service moyens ou bas sur la place Saint-Pierre, au Vatican, etc. Toutefois, la création de jeux de données annotés issus de ces systèmes peut nécessiter une autorisation préalable des responsables de ces systèmes et des autorités compétentes pour respecter au mieux la confidentialité des données récoltées et la vie privée des personnes apparaissant sur les vidéos.

Devant ce manque de données pertinentes pour l'analyse des mouvements de foule massive, nous avons décidé, pour la suite de nos travaux, d'appliquer nos approches sur le jeu de données Crowd-11. Nous estimons que de part la quantité des vidéos qu'il offre et la diversité de ses classes, c'est le jeu de données qui répond le mieux à nos attentes pour l'étude de la classification des vidéos de foule dans le cadre de l'apprentissage profond (Deep Learning).

1.6 Les annotateurs

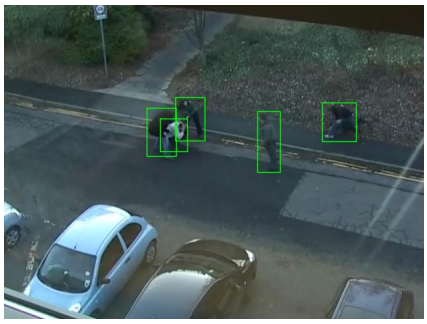
Comme nous l'avons observé dans la section précédente, il existe une multitude de jeux de données pour les tâches apparentées à l'analyse de foule. Toutefois, trop peu de jeux de données pour l'analyse des comportements de foule et l'analyse des comportements de groupe. Pour y remédier, des annotateurs ont été mis en place et permettent, à l'aide des systèmes de crowdsourcing massif, d'annoter des données brutes récupérées des sites de diffusion de la vidéo-surveillance en direct et de créer de nouveaux jeux de données à même de répondre aux nouveaux besoins de l'apprentissage profond et de la Big Data. Dans cette section, nous dressons une liste d'annotateurs qui peuvent être utilisés pour créer des jeux de données qui satisferaient les tâches de l'analyse de foule. Le Tableau 1.6 résume les caractéristiques de quelques annotateurs étudiés.

(A) SALSA (ALAMEDA-PINEDA et al., 2016)



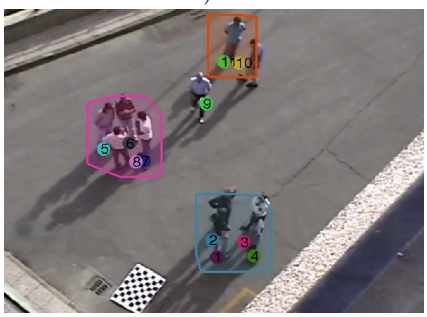
(B) MOTChallenge 2016 (MILAN et al., 2016)

(C) Agoraset (ALLAIN, COURTY et CORPETTI, 2012)

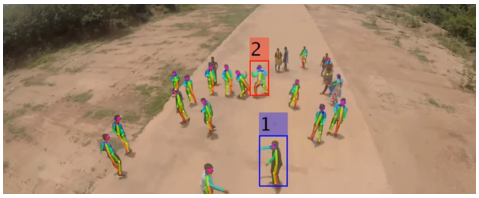


(D) Behave (BLUNSDEN et FISHER, 2010)

(E) MuseumVisitors (BARTOLI et al., 2015a)



(F) The Friends Meet (BAZZANI, CRISTANI et MURINO, 2012)



(G) Aerial Violent Individual (AVI) (SINGH, PATIL et OMKAR, 2018)

(H) UCF-101 (SOOMRO, ZAMIR et SHAH, 2012)

FIGURE 1.12 – Images tirées des jeux de données présentés dans cette thèse

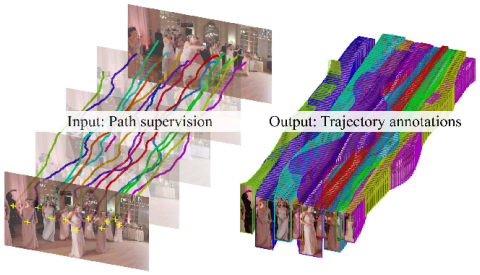
Jeu de données	Date de publication	Type de données	Capteurs utilisés	Caractéristiques
Aerial Violent Individual (AVI)	2018	Img	Drone	2000 Img's, 5124 actions violentes, 10863 piétons, 2-10 piétons/img
Town Centre	2009	Vid	MonoCam	1 Vid, ~16 piétons/img, 71500 positions de têtes annotées
I-Lids AVSS 2007 (Task 1)	2007	Vid	MonoCam	3 Vid's, 9718 régions annotées
UCSD Anomaly Detection	(MAJ 2013)	Vid	MonoCam	98 Vid's, chaque vid contient ~200 img
UCF Crowd 50	2013	Img	MultiCam	50 Img's, ~1280 piétons/img, 63705 VT
UMN SocialForce	2009	Vid	MonoCam	11 Vid's de 3 scènes.
Web Dataset	2009	Vid	MonoCam	20 Vid's, 8 décrivant des événements anormaux et 12 des événements normaux
MOT Challenge	2014 (MAJ 2016)	Vid	MonoCam	14 Vid's
Multi Task Crowd	2017	Img	MonoCam	100 Img's, 50/50 actions violentes/non-violentes, 0 to +150 piétons/img
Agoraset (Simulé)	2012	Vid	MonoCam	7 scénarios sujets à différentes simulations
CUHK Crowd	(MAJ 2014)	Vid	MonoCam	474 Vid's sur 215 scènes de foule
KITTI	2013	Vid	MonoCam	180GB de données, 6 heures d'enregistrement
Toulouse Campus Surveillance	2018	Vid	MultiCam	50 Vid's de 2 scénarios. VT pour 1 scénario.
UCF-101	2012	Vid	MonoCam	13320 vid's, 27 heures d'enregistrement. 101 classes d'actions
HMDB-51	2011	Vid	MonoCam	6766 Vid's annotées. 51 classes d'actions
PathTrack MOT	2017	Vid	MonoCam	720 Vid's, 16.287 VT
INRIA Person	2005	Img	MonoCam	1805 Img's
Kinetics	2017	Vid	Plusieurs	400 actions, 400-1150 Vid's pour chaque action, ~10s de durée chacune
Caltech Pedestrian	2009 (MAJ 2018)	Vid	MonoCam	10 heures d'enregistrement. Apparition de ~2300 piétons, 350, 000 VT de BEs
Violent Flows (ViF)	2012	Vid	MonoCam	246 vid's, ~3.6s de durée chacune
Sports-1m	2014	Vid	MonoCam	1m vid's, ~5min36s de durée, 487 classes d'actions, 1K-3K vid's/classe
Caviar	2003 (MAJ 2004)	Vid	Plusieurs	28 vid's d'une MonoCam, 26 vid's des MultiCam
Behave	2010	Vid	MultiCam	4 vid's, 76800 img's, ~125 piétons, 83545 VT
The Friends Meet (FM)	2012	Vid	MonoCam	53 vid's, 16286 img's, 3-16 piétons impliqués
MuseumVisitors	2015	Vid	MultiCam	2 scénarios
SALSA	2016	Vid	MonoCam	2 scénarios, 18 piétons, vid's durant 60 minutes chacune
COCO	2015	Img	MonoCam	330K img's, 80 classes d'objets, 200K VT
Crowd-11	2017	Vid	MonoCam	6,272 clips, 11 classes, all clips labelled
Motion Emotion	2016	Vid	MonoCam	31 vid's ; 44.000 img's ; classes : 5 de mouvements, 6 d'émotions ; VT fournie
UCF Crime	2018	Vid	MonoCam	1900 vids, 13 classes de crimes, VT fournie
CCTV Fights	2019	Vid	MonoCam	1000 vids, les 2 classes : normal/abnormal, VT fournie
Crowd-Flow	2018	Vid	MonoCam	10 vids, VT FU et trajectoires fournies
UCF crowd tracking	2008	Vid	MonoCam	3 vids, VT trajectoires fournies
GTA5 Crowd Counting (GCC)	2019	Img	MonoCam	15,212 img's, VT fournie, scènes diversifiées

TABLE 1.4 – Synthèse des jeux de données (Partie 1). **Précisions sur la colonne "Type" :** Vid/Img sont des abréviations de Vidéo et Image respectivement. **Précisions sur la colonne "Capteurs utilisés" :** MonoCam/MultiCam sont des contractions des systèmes mono-caméra et multi-caméras. *Plusieurs* signifie qu'il y a plusieurs types de capteurs impliqués dans l'enregistrement d'une scène. **Précisions sur la colonne "Caractéristiques" :** VT sont les initiales de Vérité-terrain et fait référence aux annotations. BE sont les initiales de Boîtes Englobantes. FU sont les initiales du Flux Optique

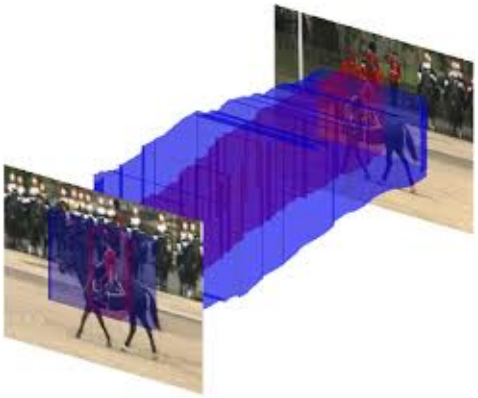
Jeu de données	Qualité	Applications	Disponibilité	Référence
Aerial Violent Individual (AVI)	Non renseignée	Détection et reconnaissance des actions de groupe violentes	Privé	(Singh, Patil et Omkar, 2018)
Town Centre	1920x1080 px, 25 Img's/s	Détection et suivi des trajectoires des piétons	Publie	(Benfold et Reid, 2011)
LLids AVSS 2007 (Task 1)	720x576 px, 25 Hertz	Détection et suivi des trajectories des piétons, Détection d'anomalies	Publie	(AVSS, 2007)
UCSD Détection d'anomalies	Non renseignée	Détection d'anomalies	Publie	(UCSD, 2013)
UCF Crowd 50	Non renseignée	Calcul des statistiques de foule	Publie	(Idrees et al., 2013)
UMN SocialForce	Non renseignée	Reconnaissance des comportements anormaux	Publie	(Mehran, Oyama et Shah, 2009)
Web Dataset	Non renseignée	Reconnaissance des comportements anormaux	Publie	(Mehran, Oyama et Shah, 2009)
MOT Challenge	Non renseignée	Détection et suivi des trajectoires de piétons et de groupes	Publie	(Milan et al., 2016)
Multi Task Crowd	Non renseignée	Reconnaissance des comportements anormaux/Calcul des statistiques de foule	Privé	(Marsden et al., 2017)
Agoraset (Simulated)	Non renseignée	Détection et suivi des trajectoires de piétons et de groupes et analyse de leurs comportements	Publie	(Allain, Courty et Corpetti, 2012)
CUHK Crowd	Non renseignée	Group/Détection d'anomalies	Unavailable	(Shao, Change Loy et Wang, 2014)
KITTI	Img's 8-bit sous format PNG	Détection et suivi des trajectoires des objets	Publie	(Geiger et al., 2013)
Toulouse Campus Surveillance	1920x1080 px	Détection et suivi des trajectoires des objets	Publie	(Malon et al., 2008)
UCF-101	320x240 px, 25 Img's/s	Détection et reconnaissance d'actions dans des scènes individuelles	Publie	(Soomro, Zamir et Shah, 2012)
HMDB-51	240 px height, 30 Img's/s	Détection et reconnaissance d'actions dans des scènes individuelles	Publie	(Kuehne et al., 2011)
PathTrack MOT	Non renseignée	Détection et suivi des trajectoires des piétons	Publie	(Manen et al., 2017)
INRIA Person	64x128 px	Détection et ré-identification des piétons	Publie	(Dalal et Triggs, 2005)
Kinetics	Non renseignée	Détection et reconnaissance d'actions dans des scènes individuelles	Publie	(Kay et al., 2017)
Caltech Pedestrian	640x480 px	Détection et suivi des trajectoires des piétons	Publie	(Dollar et al., 2012)
Violent Flows (ViF)	320x240 px	Reconnaissance des actions violentes	Publie	(Hassner, Itcher et Kliper-Gross, 2012)
Sports-1m	Non renseignée	Détection et reconnaissance des actions sportives	Publie	(Karpathy et al., 2014)
Caviar	Non renseignée	Reconnaissance des comportements de groupe	Publie	(EC, 2004)
Behave	640x480 px, 25 Img's/s	Reconnaissance des comportements de groupe	Publie	(Blunsden et Fisher, 2010)
The Friends Meet (FM)	Non renseignée	Suivi simultané des trajectoires de piétons et de groupes	Publie	(Bazzani, Cristani et Murino, 2012)
MuseumVisitors	1280x800 px, 5 Img's/s	Suivi simultané des trajectoires de piétons et de groupes, et analyse de leurs comportements	Publie	(Bartoli et al., 2015a)
SALSA	Non renseignée	Détection de groupes et reconnaissance de leurs comportements	Publie	(Alameda-Pineda, et al., 2016)
COCO	Non renseignée	Segmentation d'images, détection d'objets et des points d'intérêt	Publie	(Lin et al., 2014)
Crowd-11	220 × 400 to 700 × 1250, fréquence d'Img's/s variable	Classification des scènes de foule et Détection d'anomalies	Partiellement	(Dupont, Tobias et Luvison, 2017)
Motion Emotion	554 × 235, 30 Img's/s	Classification des scènes de foule et Détection d'anomalies	Publie	(Rabbie et al., 2016b; Rabbie et al., 2016a)
UCF Crime	240 × 320, 30 Img's/s	Détection d'anomalies et classification des comportements anormaux	Publie	(Sultani, Chen et Shah, 2018)
CCTV Fights	Définitions variables	Détection d'anomalies	Publie	(Perez, Kot et Rocha, 2019)
Crowd-Flow	Haute Définition, 300-450 Img's/vid	Analyse des mouvements de foule	Publie	(Schröder et al., 2018)
UCF crowd tracking	Faible définition, 333-492 Img's	Analyse des mouvements de foule	Publie	(Ali et Shah, 2008)
GTA5 Crowd Counting (GCC)	1080 × 1920 px	Calcul des statistiques de foule	Publie	(Wang et al., 2019)

Table 1.5 – Présentation synthétique des jeux de données (Partie 2).

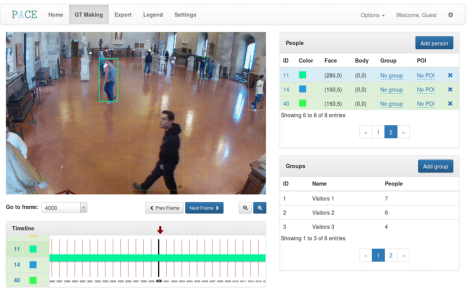
(A) LabelMe UI (Russell et al., 2008)



(B) PathTrack GT process (Manen et al., 2017)



(C) SpotOn GT process (Mettes, Gemert et Snoek, 2016)



(D) Pace UI (Bartoli et al., 2017)

Figure 1.13 – Exemples d'Interfaces Utilisateurs (User Interfaces (UI)) et du processus d'annotation (Ground Truthing (GT) process) de quelques annotateurs

1.6.1 Annotateurs d'images

(Russell et al., 2008) proposent LabelME, un outil d'annotation d'images publiquement disponible sous la forme d'une application Web. Leur objectif est de permettre à tout le monde de créer des jeux de données d'images pouvant être utilisés dans des tâches de détection d'objets, la classification d'images, et la segmentation d'images.

Dans le même sillage, (Dutta et Zisserman, 2019) proposent VGG Image Annotator (VIA), un outil Web permettant de définir et de décrire des régions dans une image en utilisant six formes différentes : une boîte englobante, un point, et des polygones pour créer des masques. L'outil permet une annotation préliminaire des images en appliquant des détecteurs d'objets avant de modifier manuellement les annotations, ce qui accélère la tâche d'annotation manuelle des images. L'outil permet également de réaliser un suivi de visages. Le code source de VIA est publiquement accessible [46].

(Andriluka, Uijlings et Ferrari, 2018) proposent l'outil d'annotation d'images Fluid Annotation. Cet annotateur repose sur trois principes :

1. Une phase de pré-annotation réalisée à l'aide d'un modèle statistique issu de l'apprentissage automatique.
2. Une annotation d'images exhaustive en une seule passe, c'est à dire, l'encadrement d'objets dans des boîtes englobantes ou dans des masques et la segmentation d'images.
3. Une responsabilisation de l'outil d'annotation dans l'optique de réduire l'intervention de l'expert humain, et ne cantonner le travail de ce dernier qu'à la revue du rendu final de l'annotation.

Cet effort de pré-annotation réduit considérablement l'erreur et l'intervention des utilisateurs humains dans le processus d'annotation. À titre comparatif, (Andriluka, Uijlings et Ferrari, 2018) assurent que l'annotation avec LabelMe peut prendre 3× plus de temps qu'avec l'outil qu'ils proposent.

Bien qu'il soit trois fois plus rapide que LabelMe et plus pratique que VIA, Fluid Annotation ne semble pas effectuer d'annotation vidéo contrairement à VIA.

1.6.2 Annotateurs de groupes et de comportements de groupes

(Bartoli et al., 2017) proposent PACE un outil open source d'annotation collaborative de scènes de foule, qui est une amélioration de l'outil WATSS proposé par (Bartoli et al., 2015b). PACE est destiné aux systèmes multi-caméras filmant des scènes d'intérieur, et vise principalement à annoter des activités de groupe. Développé en Javascript, PACE dispose de deux back-ends :

— Un back-end codé en PHP qui sert à gérer une Base de Données relationnelle,
— Un serveur REST codé en Python utilisé comme API pour exécuter des tâches de traitement d'images et de vidéos.

L'outil PACE permet de localiser une personne et tout en déterminant sa personnalité à travers deux boîtes englobantes imbriquées, mentionner les parties occluses, indiquer l'appartenance ou non à un groupe d'individus, et préciser l'orientation de son regard et de son corps. Bien que PACE soit basé sur le travail collaboratif, l'outil utilise également l'annotation prédictive : lorsqu'un piéton est localisé sur les images t_{0-k} à t_0, son étiquette est déduite sur les images t_1 à t_{1+m} [47] en utilisant le suivi de

46. Code source de VIA : https://gitlab.com/vgg/via
47. k et m étant des entiers positifs

trajectoires basé sur un filtre de Kalman (Kalman, 1960). Les algorithmes *"prédictifs"* de détection de mouvement et de piétons sont basés sur les méthodes Mixture of Gaussians (MoG) (Godbehere, Matsukawa et Goldberg, 2012) et Histogram of Gradients (HoG) (Dalal et Triggs, 2005).

1.6.3 Annotateurs de trajectoires

(Manen et al., 2017) proposent PathTrack, un annotateur de trajectoires dans le but de créer des jeux de données plus volumineux et plus riches que les jeux de données MOTChallenge. Les annotations sont produites par les utilisateurs qui regardent une scène vidéo-surveillée et suivent des cibles avec un curseur. Les annotations sont ensuite transformées en boîtes englobantes image par image. L'utilisation de PathTrack a réduit de moitié le taux d'erreur de classification pour une méthode de correspondance de personnes entraînée sur le jeu de données MOTChallenge 2015 (Leal-Taixé et al., 2015), et a amélioré les performances de NOMT réduisant les changements d'identité (Identity Switches) de 18%. À l'aide de l'outil Pathtrack, les auteurs ont mis en place le jeu de données volumineux PathTrack MOT que nous avons présenté dans la section 1.5.2. Malheureusement, cet annotateur n'est pas encore publiquement disponible.

1.6.4 Annotateurs d'actions dans des scènes individuelles

(Mettes, Gemert et Snoek, 2016) proposent Spot On, un annotateur d'actions dans les clips vidéo. Spot On émet des propositions de boîtes englobantes qui encadrent des zones de déroulement d'une action en un certain laps de temps. Cet annotateur n'est pas encore publiquement accessible.

1.6.5 Discussion sur les annotateurs

Nous avons vu dans cette section quelques annotateurs utilisés qui pourvoient aux besoins de tâches importantes en analyse de foule telles que la détection des piétons, le suivi de leurs trajectoires, la reconnaissance des actions dans des scènes de foule, et la reconnaissance des comportements de groupe.

Actuellement, de nombreux outils d'annotation destinés à la détection/suivi des trajectoires d'objets, et à la segmentation d'images sont en cours de développement (Andriluka, Uijlings et Ferrari, 2018 ; Dutta et Zisserman, 2019). Cependant, à l'exception de Spot On et PACE, peu d'annotateurs sont utilisés pour la reconnaissance d'actions dans des scènes de foule, l'analyse des comportements de foule, et d'autres tâches plus spécialisées telles que la détection de groupes et la reconnaissance des comportements de groupe. En outre, nous n'avons trouvé aucun annotateur public ou privé qui s'intéresse à des sujets moins étudiés mais non moins très d'actualité en analyse de foule tels que l'analyse des mouvements de foule massive, la reconnaissance des comportements de foule, l'estimation de la densité d'une scène, etc.

Afin d'accompagner la recherche en analyse des mouvements de foule, il est important de mettre en place des annotateurs de détection de têtes dans des foules massives qui permettent de suivre les trajectoires des têtes afin d'extraire les dynamiques d'une foule. Il est également important de créer des annotateurs d'actions individuelles et de groupe dans des scènes de foule. Par ailleurs, les annotateurs qui permettent de classer des clips occurant sur un court laps de temps issus des scènes de foule permettant d'enrichir des jeux de données tels que Crowd-11 ou d'en créer des nouveaux.

Annotateur	Date de publication	Usage	Disponibilité	Référence
LabelMe	2008	Détection d'objets, segmentation d'images	Public	(Russell et al., 2008)
VIA	2017	Détection d'objets, segmentation d'images	Public	(Dutta et Zisserman, 2019)
Fluid Annotation	2018	Détection d'objets, segmentation d'images	Public	(Andriluka, Uijlings et Ferrari, 2018)
PathTrack	2017	Détection et suivi de piétons	Private	(Manen et al., 2017)
Pace	2015	Détection de piétons, de groupes, d'orientation du regard et du corps	Public	(Bartoli et al., 2017)
SpotOn	2016	Détection et reconnaissance d'actions	Private	(Mettes, Gemert et Snoek, 2016)
WATSS	2015	Détection de piétons et de groupes	Public	(Bartoli et al., 2015b)

Table 1.6 – Présentation synthétique des annotateurs observés

1.7 Conclusion

Le déploiement de systèmes de vidéo-protection intelligents va de pair avec le développement de villes intelligentes afin de parer à tout problème de sécurité. Le recours à ces systèmes nécessite le déploiement d'algorithmes capables d'analyser en temps réel des images issues des systèmes de vidéo-protection. Comme la vidéo-protection se répand le plus souvent dans l'espace public des villes où les rassemblements sont plus fréquents (Krausz et Bauckhage, 2012), le développement de méthodes d'analyse de foule s'avère utile pour un certain nombre de municipalités.

Le but de ce chapitre est de fournir un aperçu des travaux existants en analyse de foule tout en tenant compte des tendances récentes dans ce domaine. Nous avons exploré les articles d'état-de-l'art sur l'analyse de foule (c.f. Section 1.2). Nous avons également étudié les travaux récents sur la détection des piétons et des groupes dans la section 1.3, ainsi que sur les domaines de l'analyse de foule dans la section 1.4. Nous avons dressé une liste des jeux de données les plus utilisés, et en raison de leur rareté dans les tâches de l'analyse de foule massive, nous avons étudié les possibilités offertes par les annotateurs actuellement utilisés en analyse de foule. Dans ce cadre, nous n'avons pas trouvé d'annotateurs récents qui puisse pourvoir en données annotées des thèmes importants en analyse de foule tels que le calcul des statistiques de foule, la reconnaissance des actions dans des scènes de foule, et l'analyse des comportements de foule.

À travers notre travail de recherche, nous avons constaté que le recours à des méthodes issues de l'apprentissage profond ne porte presque aucunement sur les tâches de détection de groupes et d'analyse des comportements de groupe, malgré la prévalence de ces méthodes en Vision par Ordinateur. Par ailleurs, nous dénotons un faible intérêt de la recherche pour l'analyse de foule massive, le suivi des mouvements de foule, et la détection ainsi que la prédiction d'anomalies dans les scènes de foule, en raison de la rareté de jeux de données pertinents.

À la lumière des connaissances fournies par ce chapitre, nous préconisons d'orienter les recherches futures vers l'augmentation d'annotateurs existants par des modules spécialisés qui ciblent des tâches peu communes en analyse de foule telles que la détection d'actions dans des scènes de foule, le comptage du nombre de personnes ou de groupes réalisant des actions prédéfinies, la prédiction des mouvements de foule, et l'anticipation des actions anormales.

Au début de ces travaux de thèse, nous avions l'intention de travailler sur la reconnaissance des comportements de foule afin de prédire des situations de conflit. Au fur et à mesure de la progression de nos recherches, nous avons constaté que cette tâche est double ; elle débute par une classification et se poursuit par la prédiction. La prédiction d'anomalies ou d'actions violentes dans des scènes de foule ne peut se faire sans une très bonne maîtrise de la classification ou la détection d'actions violentes. La réussite de la classification des comportements de foule nécessite certains prérequis : (1) l'existence de méthodes d'apprentissage efficaces, (2) l'accès à des infrastructures permettant d'entraîner ces méthodes d'apprentissage de modèles de classification, et (3) la disponibilité de données pertinentes à classer.

1. Aujourd'hui, les méthodes issues de l'apprentissage profond peuvent sérieusement prétendre à produire de très bons modèles de classification.
2. Grâce au concours du Mésocentre de Strasbourg, nous avons accès à la puissance de calcul nécessaire pour développer des modèles de classification et les tester sur des données.
3. Reste le troisième prérequis, difficile à satisfaire, qui représente un jeu de données massives annotées et variées en comportements de foule.

Malgré la difficulté de satisfaire ce dernier prérequis, nous avons trouvé un jeu de données assez varié mais peu volumineux, sur lequel nous avons décidé d'entraîner et de développer des méthodes de classification. Nous avons décidé d'emprunter cette voie dans la recherche afin de créer des conditions propices à de futures recherches sur la prédiction des comportements de foule et la prédiction des conflits.

Quoique peu volumineux et faiblement annoté, le jeu de données Crowd-11 (Dupont, Tobias et Luvison, 2017) représente, à ce jour, l'un des jeux de données les plus exhaustifs pour la classification des comportements de foule. Ce jeu de données représente un modèle, en termes de diversité de classes, pouvant constituer une base solide à de nouveaux jeux de données encore plus volumineux. Dans les chapitres suivants, nous décrivons les méthodes de classification que nous avons avons appliquées sur Crowd-11.

Chapitre 2

Apports de l'apprentissage par transfert et des cartes de détection de têtes dans la classification des scènes de foule

2.1 Introduction

Une équipe du CEA (Commissariat à l'énergie atomique et aux énergies alternatives) a créé un jeu de données appelé Crowd-11 (Dupont, Tobias et Luvison, 2017). Ce jeu de données, de plus de 6000 séquences vidéo, constitue une contribution majeure pour l'analyse du comportement de foule car il décrit une dizaine de comportements observables dans la voie publique.

Dans ce chapitre, nous avons appliqué, dans un premier temps, l'apprentissage par transfert pour classer les séquences vidéo de foule. Dans ce cadre, notre tâche consiste à étiqueter une vidéo. Pour ce faire, nous avons ajusté un modèle issu de l'architecture *TwoStream Inflated 3D ConvNet* (*TwoStream-I3D*) (Carreira et Zisserman, 2017) pré-entrainé sur les jeux de données ImageNet (Deng et al., 2009) et Kinetics (Kay et al., 2017), sur ce qui a pu être récupéré du jeu de données Crowd-11. Le modèle *TwoStream-I3D* ajusté a été comparé à un modèle issu de l'architecture *3D Convolutional Networks* (*C3D*) (Tran et al., 2015), pré-entrainé sur le jeu de données Sports-1m avant d'être ajusté sur Crowd-11. Suite aux résultats que nous avons obtenus, nous avons vérifié si la même stratégie de classification puisse être appliquée au jeu de données de reconnaissance d'actions UCF Crime.

Dans un second temps, nous avons étudié l'apport des cartes de détection de têtes dans la classification des scènes de foule. Nous avons utilisé le détecteur de têtes LSC-CNN (Sam et al., 2020) pour extraire les cartes de détection de têtes qui sont utilisées pour classer des scènes de foule.

Ce chapitre est organisé comme suit : Dans la section 2.2, nous avons introduit l'apprentissage par transfert dans le cadre de la classification de vidéos, et nous avons présenté les architectures pour lesquelles nous l'avons appliqué. Le jeu de données Crowd-11 est présenté plus en détails dans la section 2.2.1. Dans la section 2.3, nous avons évalué les méthodes employées pour l'apprentissage par transfert sur le jeu de données UCF Crime. Dans la section 2.4, nous avons étudié l'apport de la détection de têtes pour la classification des scènes de foule.

2.2 Apport de l'apprentissage par transfert pour la classification des scènes de foule

2.2.1 Le jeu de données Crowd-11

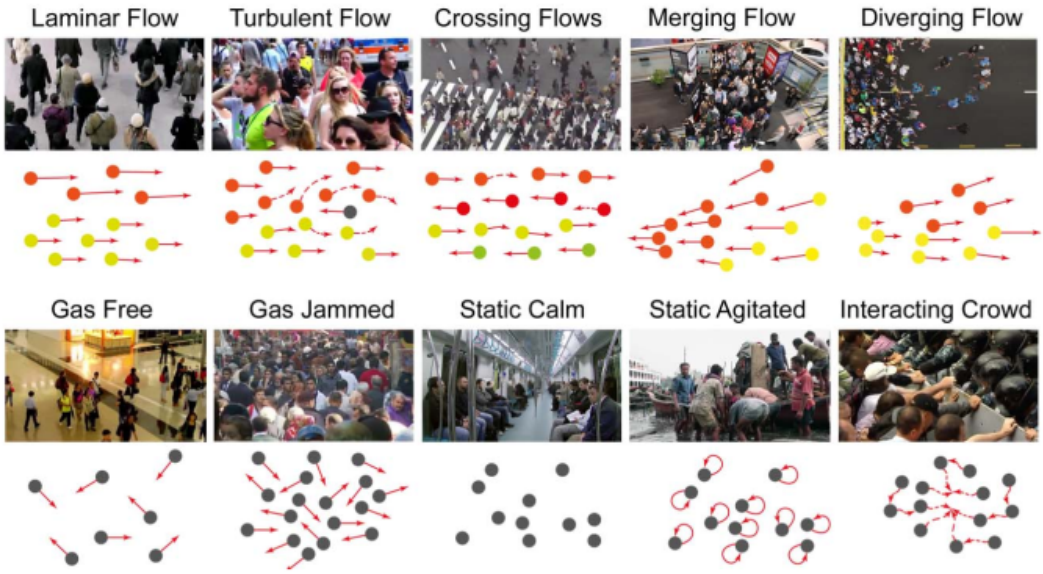


FIGURE 2.1 – Illustration des classes de mouvements du jeu de données Crowd-11, tirée de l'article de (DUPONT, TOBIAS et LUVISON, 2017). La 11ème classe, représentant les scènes vides, n'y est pas incluse

Créé par une équipe du CEA-LIST (DUPONT, TOBIAS et LUVISON, 2017), ce jeu de données récent et totalement annoté, contient plus de 6000 séquences vidéo. Les séquences vidéo disposent de résolutions variables allant de 220×400 à 700×1250, et proviennent à la base d'une multitude de sources pré-existantes. Les vidéos sont classées en 11 catégories illustrées dans la figure 2.1.

Dans ce qui suit, nous décrivons les comportements correspondant aux 11 classes contenues dans le jeu de données Crowd-11 :

0. **Gas Free :** Individus marchant dans toutes les directions sans rencontrer d'obstacles.
1. **Gas Jammed :** Foule congestionnée.
2. **Laminar Flow :** Individus marchant dans une seule direction.
3. **Turbulent Flow :** Foule marchant dans une seule direction perturbée par un individu marchant à contresens.
4. **Crossing Flows :** Deux foules qui se croisent.
5. **Merging Flows :** Deux foules qui convergent.
6. **Diverging Flow :** Une foule qui se subdivisent en deux foules.
7. **Static Calm :** Une foule d'individus statiques et calmes.
8. **Static Agitated :** Une foule d'individus statiques et agités.
9. **Interacting Crowd :** Deux foules d'individus qui s'opposent. Cette classe contient des scènes de conflits.
10. **No Crowd :** Aucune présence humaine dans la scène.

Les vidéos proviennent principalement de trois sites d'hébergement de vidéos et qui sont Youtube [1], Pond5 [2], et GettyImages [3].

Le reste provient des jeux de données suivants : UMN SocialForce, AgoraSet, PETS-2009, Violent-Flows, Hockey Fights and Movies, WWW Crowd, CUHK Crowd, et Shanghai WorldExpo'10 Crowd (Zhang et al., 2015).

La plupart de ces jeux de données sont publiquement disponibles et facilement accessibles. Toutefois, certains ne le sont plus tels que WWW Crowd, CUHK Crowd, et Shanghai WorldExpo'10 Crowd (Zhang et al., 2015). À cause de cela, nous n'avons pas pu récupérer le jeu de données Crowd-11 dans sa totalité. Ce qui a pu être récupéré représente approximativement 90% du jeu de données initial. Une estimation de la répartition des séquences récupérées par classe permet de constater qu'il n'y a pas eu une perte majeure par rapport au jeu de données initial, comme nous pouvons l'observer dans la table 2.1.

Étiquette	Nom de la classe	#vidéos (qté originale)	#vidéos récupérées
0	Gas Free	529	477
1	Gas Jammed	520	508
2	Laminar Flow	1304	1189
3	Turbulent Flow	892	862
4	Crossing Flows	763	717
5	Merging Flow	295	267
6	Diverging Flow	184	189
7	Static Calm	737	686
8	Static Agitated	410	351
9	Interacting Crowd	248	153
10	No Crowd	390	370

Table 2.1 – Tableau comparatif entre le nombre de vidéos récupérées et le nombre de vidéos original par classe pour le jeu de données Crowd-11.

2.2.2 Apprentissage par transfert

Le but de l'apprentissage par transfert est de transmettre les connaissances apprises par un modèle à partir d'un jeu de données source vers un jeu de données cible (Pan et Yang, 2010). Dans des travaux récents, l'apprentissage par transfert pour la classification des clips vidéo a été appliqué pour la reconnaissance d'actions dans des scènes individuelles (Carreira et Zisserman, 2017 ; Tran et al., 2015). Dans cette situation, l'objectif est de transférer les connaissances acquises d'un jeu de données source vers un jeu de données cible appartenant au même domaine. (Dupont, Tobias et Luvison, 2017) ont appliqué cette opération en transférant les connaissances apprises par un modèle d'un jeu de données source de reconnaissance d'actions vers un jeu de données cible illustrant des scènes de foule. Afin de surpasser les problèmes liés à l'apprentissage par transfert, en passant d'un domaine à un autre, nous appliquons l'apprentissage par transfert en lançant la procédure d'ajustement sur un nombre important d'époques (entre 30 et 40).

1. Youtube : https://www.youtube.com/
2. Pond5 : https://www.pond5.com/
3. GettyImages : https://www.gettyimages.fr/

2.2.2.1 Architectures implémentées

Nous avons sélectionné trois modèles à affiner de deux architectures : *C3D* et *TwoStream-I3D*. Le choix de l'architecture *TwoStream-I3D* est principalement motivé par les bons résultats obtenus par ses modèles par rapport aux modèles *C3D* lorsqu'ils effectuent la reconnaissance d'actions dans des scènes individuelles à partir des jeux de données UCF-101 et HMDB-51 (Carreira et Zisserman, 2017). L'équipe du CEA ayant obtenu les meilleurs résultats avec l'architecture *C3D*, son choix dans nos expériences est naturel car nous n'avons pas été en mesure de récupérer le jeu de données Crowd-11 dans son intégralité. Un modèle *C3D* pré-entrainé sur Sports-1m a obtenu ses meilleurs résultats en classant les vidéos de Crowd-11 (Dupont, Tobias et Luvison, 2017). Ce modèle représente donc pour nous le résultat de base à améliorer au cours de nos expériences. Plus de détails sur les architectures implémentées peuvent être trouvés dans cet article (Bendali-Braham et al., 2019).

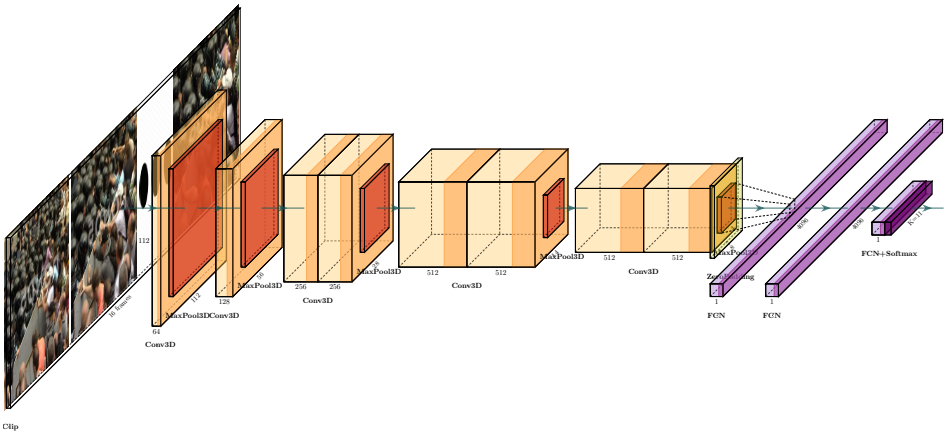


Figure 2.2 – Illustration de l'architecture 3D Convolutional Neural Network

2.2.2.1.1 Réseaux de neurones *3D Convolutional Neural Network* Nous avons décidé de ré-implémenter une version des réseaux de neurones convolutifs 3D correspondant à l'architecture décrite par (Tran et al., 2015).

Comme nous l'avons déjà mentionné, l'équipe du CEA obtient sa meilleure performance avec *C3D* après avoir pré-entrainé le modèle sur le jeu de données Sports-1m (Karpathy et al., 2014). Les réseaux C3D sont illustrés dans la figure 2.2

2.2.2.1.2 Réseaux de neurones *Two-Stream Inflated 3D* Carreira et Zisserman proposent l'architecture *Two-Stream Inflated 3D Neural Network* (2S-I3D) (Carreira et Zisserman, 2017). Cette architecture a été utilisée pour apprendre la reconnaissance d'actions dans des scènes individuelles, où elle a obtenu de très bons résultats par rapport à *C3D*. Nous l'utilisons pour apprendre à reconnaître les scènes de foule. Une illustration d'une branche I3D de cette architecture est proposée dans la figure 2.3, et l'illustration de l'architecture à deux branches 2S-I3D est proposée dans la figure 2.4.

Carreira et Zisserman ont pré-entrainé un modèle *TwoStream-I3D* sur ImageNet (Deng et al., 2009) et Kinetics (Kay et al., 2017). En testant ce modèle sur les jeux de données UCF-101 et HMDB-51, ils ont considérablement dépassé les performances des modèles *C3D* qui ont été pré-entraînés sur Sports-1m (Carreira et Zisserman, 2017). Dans notre cas, nous avons décidé de transférer les connaissances acquises

d'une branche RVB de l'architecture *I3D* sur les jeux de données sources ImageNet et Kinetics vers le jeu de données cible Crowd-11. Nous avons fait la même chose pour le modèle *TwoStream-I3D* en transférant les connaissances apprises de la branche RVB et de la branche flux optique de l'architecture au jeu de données cible. Nous avons extrait le flux optique de chaque clip vidéo en utilisant l'algorithme TV-L1 (Zach, Pock et Bischof, 2007).

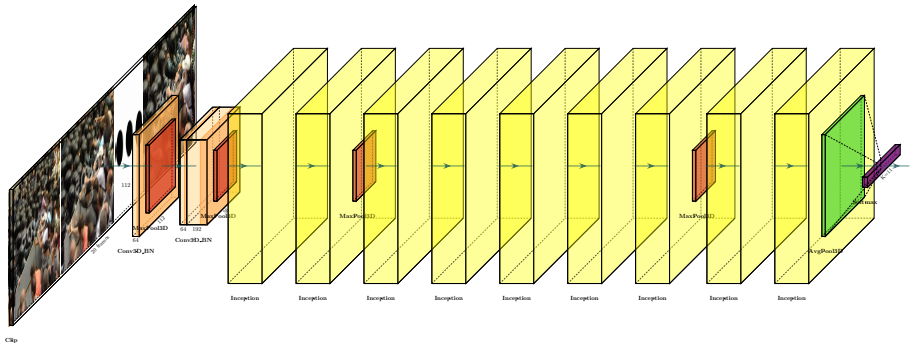


Figure 2.3 – Illustration de l'architecture Inflated 3D

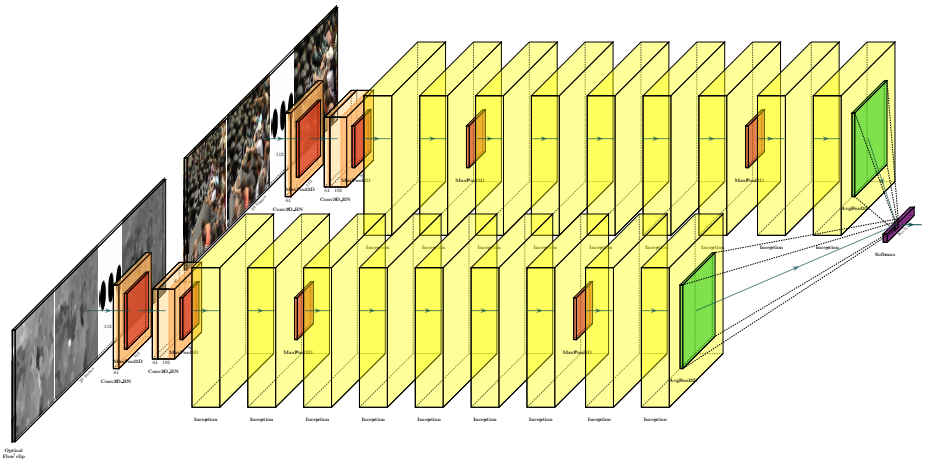


Figure 2.4 – Illustration de l'architecture TwoStream Inflated 3D (2S-I3D). L'architecture est à deux branches. La 1ère branche, en haut, reçoit en entrée des vidéos RVG (Rouge Vert Bleu). La 2ème branche, en bas, reçoit en entrée des vidéos en flux optique

2.2.3 Expérimentations sur Crowd-11

Dans les expériences que nous avons réalisées, nous avons décidé pour chaque architecture d'affiner un modèle pré-entrainé et d'entraîner un modèle à partir de zéro sur Crowd-11. Dans le cas du modèle pré-entraîné *C3D*, le pré-entraînement a été réalisé sur le jeu de données Sports-1m. Dans le cas des modèles *I3D*/*TwoStream-I3D*, le pré-entraînement a été effectué sur ImageNet, puis sur la version RVB de Kinetics pour la branche RVB, et la version flux optique de Kinetics pour la branche du flux optique.

En prenant en compte les paramètres d'apprentissages trouvés sur Tran et al. (2015) et Carreira et Zisserman (2017) respectivement pour les modèles *C3D* et *TwoStream-I3D*, nous avons choisi d'appliquer la descente du gradient stochastique

(SGD) comme fonction d'optimisation, et avons fixé le taux d'apprentissage initial à $0,003$. La fonction de perte choisie pour ces expériences est l'entropie croisée catégorielle. Afin d'être très proche des hyperparamètres utilisés pour *C3D* par (Dupont, Tobias et Luvison, 2017), nous avons divisé le taux d'apprentissage par 10 toutes les 4 époques. Cependant, nous n'avons pas reproduit cette opération lors de l'entraînement des modèles *I3D* et *TwoStream-I3D*. Pour ces derniers, nous avons choisi de diviser le taux d'apprentissage par 10 uniquement si la valeur de l'erreur augmente sur l'ensemble de validation. Pendant la phase d'entraînement, le nombre d'époques a été fixé à 40 pour les modèles *C3D* et à 30 pour les autres, afin de maximiser les chances des modèles *C3D* d'obtenir de meilleurs scores. Un modèle est enregistré à la fin de chaque époque. À la fin de la phase d'apprentissage, nous avons choisi de sauvegarder le modèle minimisant la fonction de perte lors de la phase de validation. Lors de l'affinement des modèles, nous avons décidé de ne geler aucune couche des réseaux, car les jeux de données sources sur lesquels nos modèles ont été pré-entraînés diffèrent beaucoup de ceux que nous voulons apprendre. Par conséquent, nous avons décidé de rétropropager la mise à jour des poids des réseaux sur l'ensemble des architectures des réseaux lors des phases d'apprentissage. Contrairement à (Dupont, Tobias et Luvison, 2017) nous n'avons pas appliqué de méthodes d'augmentation des données pour entraîner nos modèles. Sachant que l'augmentation des données est une méthode de régularisation, nous voulons voir si nos modèles ne souffrent pas d'un sur-apprentissage sur la version basique du jeu de données (Dvornik, Mairal et Schmid, 2018). Par ailleurs, nous voulons déterminer quelles classes nuisent à l'apprentissage de nos modèles, sans parer à ce problème en utilisant l'augmentation des données. Comme nous comptons tester plusieurs méthodes d'augmentation des données vidéo, nous préférons nous consacrer à ce problème ultérieurement.

2.2.3.1 Validation croisée à 5 échantillons

Notre version de Crowd-11 est composée de 1641 scènes. Ces scènes ont été divisées en 5769 clips vidéo. Pour éviter que des échantillons se chevauchent, nous avons décidé de conserver tous les clips d'une même scène dans un même échantillon. Lorsque nous sélectionnons une scène à ajouter à un échantillon, notre sélection fait en sorte de maintenir une similarité approximative des échantillons en termes de nombre de clips par classe. Pour entraîner ou ajuster nos modèles, nous avons divisé le jeu de données en 5 échantillons, et avons décidé d'appliquer la validation croisée 5 fois. Pour chaque itération de la validation croisée, nous avons choisi 3 échantillons pour constituer l'ensemble d'apprentissage, un pour constituer l'ensemble de validation et un dernier pour l'ensemble de test. À chaque itération de la validation croisée, l'ensemble de test change. L'ensemble de validation est choisi de manière aléatoire parmi les 4 échantillons restants.

Comme nous avons appliqué une validation croisée 5 fois pour chacun de nos trois modèles en prenant en compte les deux conditions d'entraînement : l'entraînement à partir de zéro, et l'ajustement d'un modèle pré-entraîné ; nous avons lancé 30 procédures d'entraînement [4].

4. Le code source de cette section est disponible ici :
https://github.com/MounirB/Crowd-movements-classification

2.2.3.2 Discussion des résultats obtenus

Modèle	Condition d'entraînement	Accuracy
Notre *C3D*	Sans pré-entraînement	31.88%
C3D Dupont et al.	Sans pré-entraînement	46.9%
Notre *C3D*	Pré-entrainé	58.29%
C3D Dupont et al.	Pré-entrainé	61.6%

TABLE 2.2 – Comparaison entre notre version de *C3D* et celle de (DUPONT, TOBIAS et LUVISON, 2017)

Architecture	Condition d'entraînement	Moyenne	Min	Max
I3D	Sans pré-entraînement	47.01%	40%	53.36%
C3D	Sans pré-entraînement	31.88%	28.82%	36.43%
TwoStream-I3D	Sans pré-entraînement	47.85%	43.91%	52.42%
I3D	Pré-entrainé	58.97%	56.33%	60.17%
C3D	Pré-entrainé	58.29%	57.19%	60%
TwoStream-I3D	Pré-entrainé	**68.2%**	**66.01%**	**70.34%**

TABLE 2.3 – Accuracy obtenue à la suite de la validation croisée avec K=5

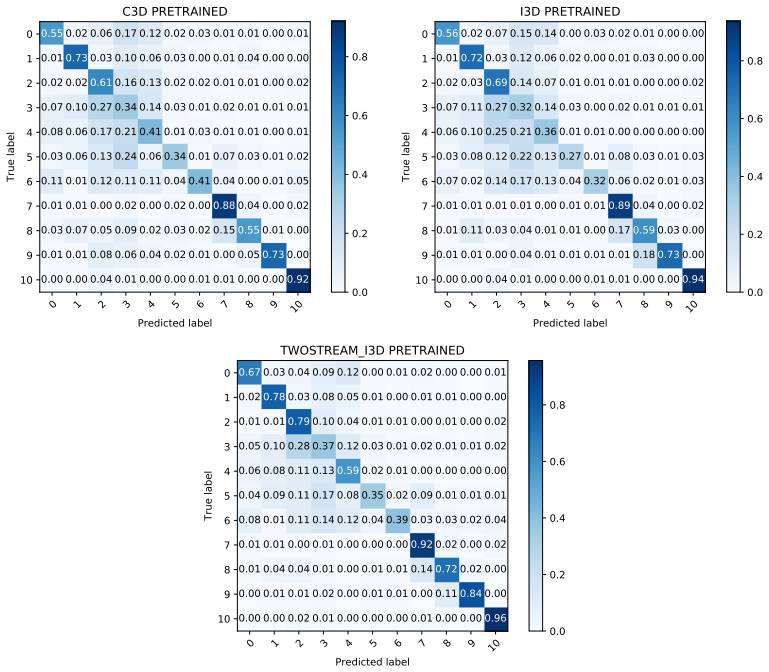


FIGURE 2.5 – Matrices de confusion globales, des modèles pré-entrainés, calculées à la suite de la validation croisée à 5 échantillons

Selon les résultats affichés sur la table 2.2, nous observons que le modèle *C3D* entrainé à partir de zéro n'est pas aussi performant que le modèle entraîné par DUPONT, TOBIAS et LUVISON (2017). Cela peut avoir plusieurs raisons : une possible différence entre les hyperparamètres que nous utilisons et ceux qui sont utilisés par les auteurs de Crowd-11 pour l'entraînement de leur modèle, la différence entre nos deux jeux de données, et le fait que nous n'ayons pas recours à l'augmentation des données

vidéo. Selon les résultats affichés dans la table 2.3, nous constatons que les modèles *C3D* et *I3D* obtiennent des résultats presque identiques lors de la classification des clips vidéo lors de phase de test. *C3D* n'est dépassé que d'environs 0.60% d'accuracy par le modèle *I3D*. Cette légère différence de performances peut s'expliquer par le fait que l'architecture *C3D* doit entraîner 78 millions de paramètres, tandis que l'architecture *I3D* compte 12 millions de paramètres ainsi qu'une structure profonde. De plus, nous observons que le modèle *TwoStream-I3D* arrive bien à tirer profit du flux optique lors de l'affinement. Cela n'est pas le cas lorsqu'il est entraîné à partir de zéro. Globalement, les modèles *TwoStream-I3D* obtiennent les meilleurs scores.

À partir des matrices de confusion affichées sur la figure 2.5, nous observons que chaque modèle éprouve des difficultés face aux mêmes classes indicées de 3 à 6, qui sont respectivement : *Turbulent Flow*, *Crossing Flows*, *Converging Flow* et *Diverging Flow*. Nous constatons, également, que les clips appartenant à ces classes, y compris la classe *Laminar Flow*, sont fréquemment confondus. Alors que la classe *Laminar Flow* n'est pas une grande source de confusion, car la foule y suit une direction unique, les multiples transitions clés observables dans les quatre autres classes peuvent perturber la décision du classifieur. Par exemple, nous observons que la classe *Merging Flow* n'est pas confondue avec la classe *Diverging Flow*, ce qui montre que le classifieur apprend bien à différencier entre ces deux comportements. Cependant, ces deux classes sont fréquemment confondues avec la classe *Crossing Flows*. Lorsqu'une foule se croise avec une autre, des comportements de convergence et de divergence sont observés. De plus, alors que *Crossing Flows* est composée par ≈ 850 clips, les classes *Merging Flow* et *Diverging Flow* sont composées par ≈ 200 clips chacune (comme indiqué dans la table 2.1). Cette situation peut amener deux classes à être englouties par une classe plus globale, telle que la classe *Crossing Flows*.

2.3 Reconnaissance d'actions sur le jeu de données UCF Crime

UCF Crime est un jeu de données vidéo issu de la vidéo-surveillance proposé par (Sultani, Chen et Shah, 2018). La moitié des vidéos filmées contiennent des comportements anormaux. L'autre moitié des vidéos portent sur des comportements normaux. En plus de la classe décrivant les scènes normales, 13 comportements anormaux sont illustrés dans les vidéos de ce jeu de données. Ces comportements sont les suivants :

1. Abuse : comportements abusifs sur personnes faibles ou n'étant pas en mesure de se défendre ;
2. Arrest : scène d'arrestation ;
3. Arson : incendie criminel ;
4. Assault : agression ;
5. Road Accident : accident de la route ;
6. Burglary : cambriolage ;
7. Explosion : explosion ;
8. Fighting : bagarre ;
9. Robbery : vol à l'arrachée ;
10. Shooting : scène de tirs ;
11. Stealing : vol sans violence ;

12. Shoplifting : vol à l'étalage ;
13. Vandalism : actes de vandalisme.

Ce jeu de données contient 1900 vidéos qui totalisent 128 heures d'enregistrement.

2.3.1 Contexte (État de l'art)

Les travaux récents sur le jeu de données UCF Crime se regroupent en trois catégories. La 1ère catégorie, riche en travaux, se focalisent sur la détection d'anomalies seulement (Zhong et al., 2019 ; Landi, Snoek et Cucchiara, 2019 ; Zaheer et al., 2020 ; Ullah et al., 2020 ; Lv et al., 2021 ; Aqeel et al., 2020). La 2ème catégorie comprend les travaux qui mêlent détection d'anomalies et reconnaissance d'actions anormales (Sultani, Chen et Shah, 2018 ; Sun et al., 2020). La 3ème catégorie, plus rare, concentre les travaux portant sur la reconnaissance d'actions anormales seulement (Girdhar, Johri et Virmani, 2020 ; Qi, Liu et Fu, 2020).

(Sultani, Chen et Shah, 2018), les auteurs du jeu de données UCF Crime, ont proposé deux approches : une destinée à la détection d'anomalies et l'autre destinée à la reconnaissance d'actions dans des scènes normales et anormales. L'approche qu'ils proposent pour la détection d'anomalies est basée sur la méthode Multiple Instance Learning, où les instances sélectionnées le sont après un classement des instances dans chaque sac de clips (Bag of Video segments). Ils considèrent chaque vidéo comme un sac de clips. Il y a des sacs positifs ; qui contiennent des clips anormaux, et des sacs négatifs ; qui ne contiennent que des clips de scènes normales. 32 clips sont élus pour chaque sac. La fonction de classement (ranking instances à l'intérieur de chaque sac) se sert de la hinge loss complétée par les propriétés de smoothness et de sparsity afin de classer les clips dans chaque sac. Les deux clips qui obtiennent les meilleurs scores sont sélectionnés pour représenter chaque sac. Un clip ou instance est représentée par un vecteur de caractéristiques extraites des 16 images représentant un clip. L'extracteur de caractéristiques est le réseau C3D sur lequel les auteurs appliquent une régularisation L2. Ils comparent leur méthode à une méthode de classification binaire basique, aux méthodes de (Hasan et al., 2016 ; Lu, Shi et Jia, 2013) qu'ils surpassent en termes du score Area Under the Curve (AUC). Dans le cadre de la classification de clips en 14 classes, les auteurs proposent deux méthodes de base pour les recherches futures :

— La 1ère méthode est basée sur l'extraction des caractéristiques à l'aide d'un modèle C3D qui s'entraîne sur des clips de 16 frames. Les caractéristiques extraites sont envoyées en entrée d'un algorithme KNN pour la classification.
— La seconde méthode est basée sur le réseau TCNN (Tube of Interest CNN). La particularité de ce réseau CNN est qu'il dispose d'un Tube of Interest qui est une couche de pooling qui remplace la couche de pooling usuelle de type 3d max pooling de la 5ème couche d'un réseau C3D.

Les auteurs ont obtenu respectivement 23% et 28% d'accuracy sur la 1ère et la seconde méthode en évaluant la classification sur UCF Crime.

(Zhong et al., 2019) ont mis en place une approche qui permet d'apprendre simultanément un algorithme de classification binaire des clips en actions normales et anormales, ainsi qu'un algorithme de nettoyage de bruit (Noise cleaner). Ce couple algorithmique a pour tâche la détection d'anomalies dans les scènes de UCF Crime, dans le cadre d'un apprentissage supervisé sous la contrainte de la faible annotation (Weakly Supervised Anomaly Detection). Le nettoyage de bruit des clips anormaux repose sur l'usage d'un réseau Graph Convolution Neural Nets. (Zhong et al., 2019) sont les premiers à utiliser les Graph Convolution Nets dans le cadre du traitement de

vidéos. Dans leur approche, ils commencent par extraire des caractéristiques spatio-temporelles à l'aide de l'algorithme de classification. Ces caractéristiques sont, par la suite, envoyées à deux Graph Convolutional Nets qui travaillent simultanément : le 1er calcule la similarité entre les caractéristiques, et le second calcule leur consistence temporelle. Le but des deux GCN est de produire des caractéristiques avec le moindre bruit possible. Celles-ci sont à leur tour envoyées en entrée de l'algorithme de classification qui affinera ses résultats de classification. Ils ont employé deux types d'algorithmes de classification, des réseaux C3D pré-entraînés sur Sports-1m, et des réseaux TwoStream-TSN pré-entraînés sur Kinetics-400. Ils obtiennent leurs meilleures performances avec les réseaux TwoStream-TSN. Ils mènent leurs expériences sur les jeux de données UCF Crime où C3D est dépassé par la méthode (Sultani, Chen et Shah, 2018) toutefois le modèle TwoStream TSN dépasse toutes les autres méthodes en termes de score Area Under the Curve (AUC). Ils obtiennent également les meilleurs résultats sur ShanghaiTech (Zhang et al., 2016c) et UCSD Peds.

Comme (Sultani, Chen et Shah, 2018), (Sun et al., 2020) font de la détection d'anomalies à travers une modification de l'approche MIL de (Sultani, Chen et Shah, 2018) et de la classification des clips de UCF Crime. Dans le cadre de la classification, (Sun et al., 2020) proposent la méthode Discriminative Anomalous Clip Miner (DACM). Dans cette approche, ils commencent par découper une vidéo en n clips. Un modèle 2-D BN-Inception, pré-entraîné sur le jeu de données Kinetics-400, est employé pour l'extration des caractéristiques d'images sélectionnées de chaque clip à l'aide de la stratégie de Sparse-Sampling. La moyenne de ces caractéristiques est envoyée en entrée d'un réseau ResNet 3D pour la caractérisation du contexte temporel. Ensuite, le module d'attention CL-CAM utilise les caractéristiques apprises pour classer l'action décrite dans la vidéo. (Sun et al., 2020) ont comparé leur méthode de classification aux méthodes de (Sultani, Chen et Shah, 2018 ; Hou, Chen et Shah, 2017), et une combinaison des méthodes (Hou, Chen et Shah, 2017 ; Zhu et Newsam, 2019). Ils obtiennent les meilleurs scores d'accuracy à hauteur de 35.1%.

Pour l'approche MIL de (Sultani, Chen et Shah, 2018), (Sun et al., 2020) proposent de modifier le système de sélection des clips anormaux basé sur le classement des instances anormales à l'intérieur des sacs anormaux. Dans ce cadre, ils proposent deux méthodes : Attentive MIL et Contrastive Pattern Learning (CPL). Dans l'Attentive MIL, la méthode MIL est modifiée en remplaçant la ranking loss avec une attentive ranking loss pour optimiser le modèle et exploiter la majorité des clips. Le but de cette stratégie est de sélectionner des clips qui reflètent plus précisément l'anomalie contenue dans les vidéos. Par ailleurs, (Sun et al., 2020) proposent l'approche CPL pour détecter les anomalies à partir des clips sélectionnés. CPL regroupe les clips en groupes positifs contenant les clips anormaux et négatifs contenant les clips normaux. Un réseau, de pattern learning, est utilisé pour associer les caractéristiques des deux groupes à des vecteurs (embeddings). Ils emploient la distance euclidienne au carré pour séparer les vecteurs ayant des signes opposés et rapprocher les vecteurs de même signe. Lors de la phase d'évaluation, un algorithme KNN est employé pour classer les vecteurs. Les expériences qu'ils ont menées sur UCF Crime montrent qu'ils obtiennent les meilleures performances en termes de détection d'anomalies et de classification des actions anormales. Le détecteur d'anormales proposé (CPL) est évalué avec le score AUC. Au cours de leurs expériences, (Sun et al., 2020) ont comparé la méthode Attentive MIL et CPL aux méthodes de (Hasan et al., 2016 ; Lu, Shi et Jia, 2013 ; Sultani, Chen et Shah, 2018 ; Zhong et al., 2019 ; Zhu et Newsam, 2019). La méthode Attentive MIL est dépassée par Motion-aware et (Zhong et al., 2019). La méthode CPL réalise les meilleurs résultats, et une combinaison de CPL et

Motion-aware (Zhu et Newsam, 2019) améliore les performances de CPL.

(Qi, Liu et Fu, 2020) proposent de réaliser la détection d'anomalies en appliquant le few-shot learning sur le jeu de données UCF Crime+. UCF Crime+ est une version modifiée de UCF Crime où les vidéos sont découpées en clips de 10 secondes chacun. (Qi, Liu et Fu, 2020) utilisent un modèle I3D, pré-entraîné sur le jeu de données Kinetics-400 et ajusté sur UCF-101, pour extraire des caractéristiques spatio-temporelles de chaque clip de UCF Crime+ représenté par 64 images. Par la suite, ils utilisent le réseau de neurones prototypical pour appliquer le few-shot learning sur les caractéristiques extraites des clips UCF Crime+. Le Prototypical Network (PN) est un encodeur qui lie chaque classe à son support-set dans un espace vectoriel (embedding space). Par la suite, ils classent les vecteurs de l'embedding space à l'aide d'un algorithme KNN. Le réseau few-shot learning (prototypical) a été entraîné sur 32 classes (100 vidéos chacune) du jeu de données UCF-101. Dans leur travail, ils ont proposé deux améliorations à leur approche. Tout d'abord, ils utilisent la Triplet Hard Loss (THL) afin de renforcer les distances inter-classes et intra-classes des clusters de prototypes. Par ailleurs, ils emploient la Feature Pyramid Fusion (FF) aux caractéristiques extraites du modèle I3D afin d'améliorer la qualité des caractéristiques extraites. Sur UCF Crime+, la méthode est entraînée selon la stratégie de classification 20-way épisodes de 5-shot chacune. Ils ont comparé trois versions de leur méthode sur l'ensemble de test de UCF Crime+.

— La version de base avec le réseau prototypical seul (PN) ;
— Une version améliorée avec seulement la triplet hard loss (PN + THL) ;
— Une version contenant, en plus des deux dernières améliorations, la Feature Pyramid Fusion (PN+THL+FF).

Les résultats de leurs expériences montrent qu'ils obtiennent leur meilleure accuracy avec l'approche qui contient toutes les améliorations avec la stratégie de classification 5-way 20-shots. Ils obtiennent une accuracy de 67.9%.

(Girdhar, Johri et Virmani, 2020) proposent l'approche Incept_LSTM, une hybridation d'un modèle Inception v3 pré-entraîné sur un jeu de données de classification d'images et le modèle LSTM, pour la reconnaissance des actions anormales. Dans cette approche, le réseau Inception v3 est utilisé pour l'extraction des caractéristiques et la classification des images extraites. L'output de l'extraction de caractéristiques spatiales sont passées au travers de deux unités LSTM empilées. La 1ère unité LSTM est utilisée pour apprendre le contexte temporel, la 2ème unité LSTM est utilisée pour améliorer les performances d'accuracy. L'apprentissage ainsi que l'évaluation ont été réalisés sur 4 classes du jeu de données UCF Crime. Ces classes sont : Assault, Burglary, Fighting, Normal. Ils ont évalué leur approche selon les métriques d'accuracy, matrice de confusion, et score F1. Leur approche atteint 91% d'accuracy lors de la validation. Ils ont comparé leur méthode à la méthode LSTM Ensemble (Ordóñez et Roggen, 2016) qu'ils ont battue et ont obtenu un score F1 de 0.89.

(Landi, Snoek et Cucchiara, 2019) emploient des tubes spatio-temporels contenant des actions pour représenter les événements anormaux dans des vidéos. Ils proposent un pipeline algorithmique qui extrait ces tubes préalablement identifiés à l'aide de l'annotateur Vatic (Vondrick, Patterson et Ramanan, 2013). Par la suite, ils caractérisent ces tubes à l'aide d'un modèle TwoStream-I3D pré-entrainé sur les jeux de données Kinetics et ImageNet, qui apprend également à classer ces tubes de manière binaire en tubes normaux ou anormaux à l'aide d'un réseau de neurones constitué d'une couche de convolutions et de 4 couches de Fully-Connected Network (FCN). Ils entraînent leur modèle sur un jeu de données extrait de UCF Crime, nommé UCFCrime2Local. Ce jeu de données contient 100 vidéos anormales provenant des 6 classes : Arrest, Assault, Burglary, Robbery, Stealing, et Vandalism, et 200

vidéos normales. Ce jeu de données est augmenté d'une information sur la localisation des actions anormales à l'aide des Bounding Boxes qui sont ajoutées via l'annotateur Vatic. Dans ces tubes, l'action débute lors de l'apparition de tous les acteurs réalisant une action et se termine à la fin de l'action. Leurs expériences, évaluées à l'aide des métriques ROC et AUC, montrent que l'emploi de la localité améliore de 18.61% la performance par rapport à l'usage de la vidéo complète lors de l'apprentissage.

(Zaheer et al., 2020) proposent une approche qui associe classification et regroupement binaires pour la détection d'anomalies. Leur algorithme cible les jeux de données vidéo faiblement annotés (weakly video-labeled datasets). Dans ces jeux de données, une étiquette est associée à une vidéo au lieu de l'être pour chaque frame de la vidéo ce qui engendre beaucoup de bruit lors du processus de détection d'anomalies. Il y a du bruit dans les vidéos anormales, car l'anomalie occupe une petite portion de la vidéo. Les auteurs proposent de découper la vidéo en plusieurs fragments de K frames qui ne se chevauchent pas. Les fragments (ou clips) sont envoyés en entrée d'un extracteur de caractéristiques, qui est un modèle C3D pré-entraîné sur Sports-1m. Les caractéristiques sont transmises à un réseau de neurones constitué de deux couches FCN (Fully Connected Network) chacune suivie par une activation de type ReLU. Une classification binaire des fragments normaux et anormaux est faite à l'issue de la couche FC2. Les caractéristiques de la couche FC1 sont envoyées en entrée à l'algorithme K-means (2-means) qui a pour but de rapprocher les clusters issus des vidéos totalement normales et d'éloigner les clusters issus des vidéos anormales. Dans les clusters des vidéos anormales, un des deux clusters ne contient que du bruit qui constitue des passages normaux de la vidéo considérée comme anormale. Le réseau de classification FCN et le clustering K-means complètent leurs connaissances lors du processus d'entraînement afin de mieux détecter les fragments anormaux dans les vidéos anormales et supprimer le bruit qui y est représenté par les fragments normaux. Les auteurs ont évalué leurs méthodes en utilisant les métriques AUC et ROC. Sur UCF Crime, ils ont comparé leur approche à des méthodes SOTA, et ils ont obtenu de meilleures performances que (Sultani, Chen et Shah, 2018 ; Hasan et al., 2016 ; Lu, Shi et Jia, 2013 ; Zhang, Qing et Miao, 2019), et des résultats similaires aux performances de (Zhong et al., 2019). Ils ont battu toutes les méthodes précédemment citées sur UCSD Ped2 et ShanghaiTech (Zhang et al., 2016c). Ils ont obtenu les scores frame-level AUC de 79.54%, 84.16%, 94.47% respectivement sur les jeux de données UCF Crime, ShanghaiTech (Zhang et al., 2016c), et UCSD Ped2.

(Ullah et al., 2020) proposent une approche de one-shot learning basée sur l'apprentissage d'un Réseau de Neurones Siamois (Siamese 3D CNN) pour la détection d'anomalies dans les vidéos de UCF Crime. Leur approche se décline en deux étapes ; une étape d'apprentissage et une étape de validation. Dans l'étape d'apprentissage, ils extraient d'abord les caractéristiques spatio-temporelles de clips de 16 frames représentées en niveaux de gris. Cette extraction est faite à partir d'une paire de clips. Le premier clip est le clip de référence pour lequel on connaît la classe et le deuxième clip est tiré d'une opération de sliding window sur la vidéo. Les caractéristiques extraites sont comparées au niveau d'un réseau d'appariement (matching network) qui établit une distance absolue entre les caractéristiques extraites de chaque clip. Par la suite, une fonction sigmoïde est appliquée sur cette distance. Lors de la phase de validation, les performances du réseau d'entraînement sont évaluées selon la méthode N-way. Un clip dont le réseau prédit la classe est comparé à N clips pour lesquels le réseau décide s'ils sont similaires ou différents. Le modèle est censé produire les scores de similarité les plus élevés pour les clips d'une même classe ou des scores de dissimilarité élevés pour les clips dont les classes sont différentes. Le modèle Siamois 3D CNN proposé est comparé aux modèles Siamois C3D et Siamois I3D sur UCF Crime, et le modèle

des auteurs réalise les meilleures performances en termes des métriques AUC, ROC, accuracy moyenne, zone mémoire occupée par les modèles (taille en mégabytes), et complexité temporelle. Le modèle siamois des auteurs est dépassé par le modèle C3D dans le cas 10-way (accuracy moyenne), mais il dépasse les autres modèles en termes d'accuracy moyenne (5-way, 14-way (ici *way* fait référence au nombre de classes différentes)), du nombre de paramètres entraînables, et en termes de taille du modèle en mb. La contreperformance de tous les modèles est due au bruit présent dans le jeu de données ; abondance de passages normaux dans les vidéos considérées initialement comme anormales, où l'action anormale se déroule sur un court laps de temps.

(Lv et al., 2021) soutiennent que les méthodes de l'État-de-l'art ne modélisent pas assez le contexte temporel des anomalies, ce qui ne leur permet pas de bien localiser des anomalies dans les vidéos anormales ne bénéficiant pas d'une annotation temporelle accrue. En contrepartie, ils proposent la Weak-Supervised Anomaly Localisation (WSAL), une méthode de localisation des anomalies dans une vidéo anormale faiblement annotée "Weakly-labeled video". En l'occurrence, l'étiquetage porte sur la classe de la vidéo, sans préciser exactement le moment où survient l'anomalie dans la vidéo. Leur méthode repose sur le High-order Context Encoding (HCE). Cet encodage du contexte consiste à combiner la prise en compte des indices sémantiques spatiaux contenus dans la vidéo ainsi que des indices temporels caractérisés par les variations temporelles dans le but de localiser l'occurrence d'une anomalie. Ils renforcent leur approche par une méthode de suppression du bruit appelée Weak-Supervision Enhancement Strategy (WSE). La WSE consiste à simuler du bruit induit par les changements brusques d'environnement, les défaillances matérielles, et la longueur des vidéos contenant plusieurs passages normaux. En simulant ces situations, la stratégie WSE permet à l'approche WSAL de mieux repérer l'occurrence du bruit et de l'exclure des vidéos anormales. Les vidéos sont découpées en segments et sont envoyées en entrée d'un réseau TSN (version BN-Inception). Les caractéristiques extraites sont traitées dans un module High-order Context Endoding (HCE) qui génère des scores d'anomalies des indices sémantiques et des variations temporelles. Les scores prédits sont aggrégés dans une fonction de coût MIL Margin qui se base sur les étiquettes vidéo. Par ailleurs, (Lv et al., 2021) proposent le jeu de données TAD (Traffic Anomaly Detection) pour détecter des anomalies lors du trafic routier. Ils mènent également des expériences sur le jeu de données UCF Crime. Ils comparent leur méthode à celles de (Sultani, Chen et Shah, 2018 ; Zhong et al., 2019) qu'ils surpassent en termes des métriques ROC et AUC. Ils établissent, ainsi, un nouveau score État-de-l'art de 67% d'AUC.

(Aqeel et al., 2020) proposent une méthode basée sur l'extraction des caractéristiques en utilisant un Convolutional AutoEncoder et un Generative Adversarial Network (GAN). Les caractéristiques sélectionnées sont utilisées pour classer les clips de manière binaire, en classes normale et anormale, à l'aide d'un des algorithmes de classification binaire suivants : Gaussian Naïve Bayes, SVM, Arbres de Décisions, Régression Linéaire, Random Forests, et encore une fois le modèle Convolutional AutoEncoder. Tout d'abord, les vidéos d'entraînement de UCF Crime sont converties en images. Celles-ci sont envoyées en entrée d'un Convolutional Autoencoder pour en extraire des caractéristiques permettant de détecter des anomalies. Les GAN utilisent les caractéristiques pour améliorer la résolution des frames. Les techniques de classification reçoivent en entrée les vidéos reconstruites pour y détecter des anomalies. Ils ont mené leurs expériences sur le jeu de données UCF Crime. Dans ces expériences, ils ont comparé les performances des six méthodes de classifications en termes d'AUC, ROC, et EER. Dans ce cadre, le réseau CAE semble réaliser les meilleures performances en termes d'accuracy.

2.3.2 Expériences réalisées sur UCF Crime

Nous avons entraîné de zéro et ajusté des modèles issus des architectures vues dans la section 2.2. Pour chacun des 6 modèles entraînés ou ajustés, une validation croisée à 4 plis a été appliquée en nous basant sur la division des données proposée par les auteurs de UCF Crime. Pour rappel, les modèles entraînés sont les suivants :

1. Le modèle C3D entraîné de zéro sur UCF Crime ;
2. Le modèle C3D pré-entraîné sur Sports-1m et ajusté sur UCF Crime ;
3. Le modèle I3D entraîné de zéro sur UCF Crime ;
4. Le modèle I3D pré-entraîné sur ImageNet et Kinetics-400 et ajusté sur UCF Crime ;
5. Le modèle TwoStream-I3D (2S-I3D) entraîné de zéro sur UCF Crime ;
6. Le modèle 2S-I3D pré-entraîné ImageNet et Kinetics-400 et ajusté sur UCF Crime ;

À travers les boîtes à moustaches illustrées dans les Figures 2.6 et 2.7, nous présentons les résultats obtenus par ces modèles dans la classification des 14 classes de vidéos du jeu de données UCF Crime.

Comme lors des expériences sur Crowd-11, le modèle obtenant les meilleures performances en termes d'accuracy et du taux d'erreur lors de l'évaluation est le modèle TwoStream-I3D ajusté. Toutefois, les performances obtenues par nos modèles sont mauvaises comparées à celles obtenues par (Sultani, Chen et Shah, 2018). Nous perdons 8% d'accuracy comparé à leur approche de base. Une première raison à cela est que contrairement à (Sultani, Chen et Shah, 2018), nous n'utilisons pas un modèle statistique issu du Deep Learning pour extraire des caractéristiques et ensuite une méthode pour exploiter ces caractéristiques lors de classification des vidéos.

Tels que (Sultani, Chen et Shah, 2018), une partie des travaux vus dans la section 2.3.1, qui travaillent sur la classification des actions anormales, appliquent un pré-traitement massif sur le jeu de données avant de procéder à l'apprentissage de la classification, ce que nous n'avons pas voulu tester dans notre approche d'apprentissage de bout-en-bout de la classification de ces vidéos.

(Zhong et al., 2019) débutent par une étape de détection de ces passages anormaux dans les vidéos et les nettoient des passages considérés comme normaux avec le Graph Convolutional Label Noise Cleaner, avant de procéder à l'apprentissage de la classification **binaire** des vidéos pré-traitées qui sont classées simplement en actions normales ou anormales. (Girdhar, Johri et Virmani, 2020) obtiennent de très bonnes performances, mais sur une version modifiée de UCF Crime qui ne reprend que 4 classes de la totalité du jeu de données. (Qi, Liu et Fu, 2020) pré-traitent UCF Crime en le découpant en vidéos de 10 secondes. Ils obtiennent également de bons scores lors de l'apprentissage de la classification sur 5 classes du jeu de données UCF Crime. (Sun et al., 2020) utilisent un module d'attention dans leur réseau d'apprentissage de la classification couplée avec un mécanisme de sélection des caractéristiques à apprendre de chaque vidéo-clip. Ils détiennent jusqu'ici le meilleur score en termes de classification des 14 classes de vidéos de UCF Crime avec 35.1% d'accuracy.

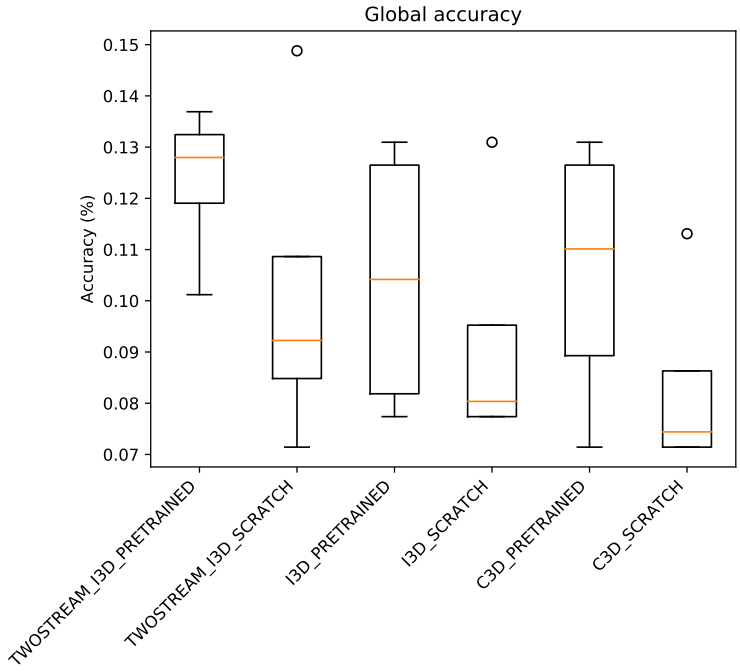


FIGURE 2.6 – Illustration de l'accuracy globale pour les 6 modèles utilisés pour la classification des clips de Crowd-11 et qui sont évalués ici dans la classification des clips de UCF Crime

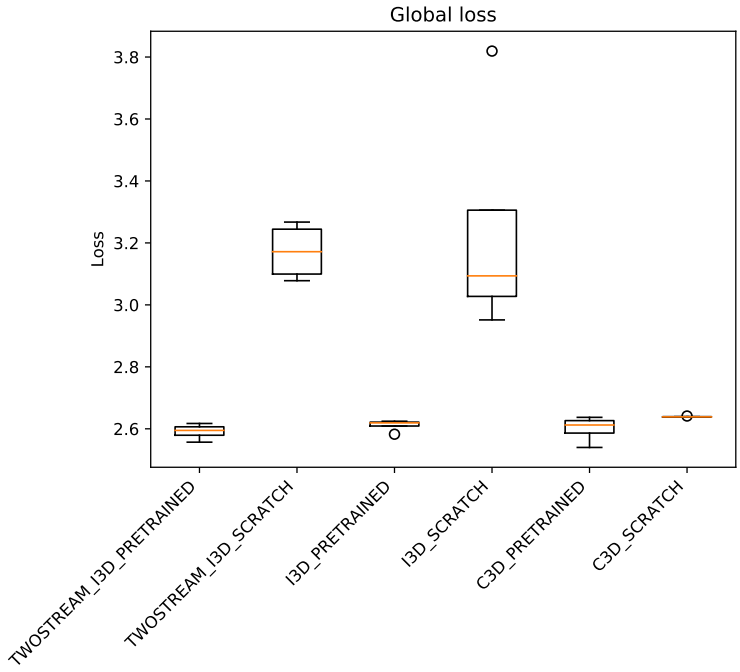


FIGURE 2.7 – Illustration de l'erreur globale pour les 6 modèles utilisés pour la classification des clips de Crowd-11 et qui sont évalués ici dans la classification des clips de UCF Crime

2.4 Apport de la détection de têtes dans la classification des scènes de foule

Dans cette section, nous explorons l'apport des cartes de détection de têtes pour la classification des vidéos de foule. Dans ce registre, nous avons décidé d'utiliser un détecteur de têtes pré-entraîné pour localiser dans un premier temps toutes les têtes dans une scène de foule. Cette détection conduit à la création de cartes de détection. Ces détections, image par image, sont converties en clips de cartes de détection qui sont envoyés, par la suite, en entrée à des réseaux CNNs à des branches de réseaux CNNs lors des phases d'apprentissage, de validation, et de test.

2.4.1 Contexte

Dans cette section, nous allons poser le contexte qui nous a incité à choisir une méthode d'extraction de caractéristiques basée sur la détection de têtes et, pour ce faire, la méthode que nous avons choisie. Les résultats de la détection de têtes seront soit utilisés comme méthode principale de pré-traitement dans la caractérisation des scènes de foule ou épauleront d'autres méthodes de pré-traitement telles que le flux optique.

Afin de réaliser la détection de têtes, nous nous sommes orientés vers le détecteur Locate-Size-Count CNN (LSC-CNN) (Sam et al., 2020). C'est un réseau de neurones convolutifs (CNN), qui réalise une détection dense de têtes en localisant et en comptant le nombre de têtes apparaissant dans une scène de foule. La force de ce détecteur est qu'il est capable de s'entraîner sur des annotations sous forme de points et de trouver la taille de la boite de limitation (bounding box) carrée encadrant chaque tête. L'avantage de cette méthode de détection est la disponibilité de son implémentation en open-source sur github [5] par ses auteurs.

LSC-CNN utilise les réseaux VGG-16, pré-entraînés sur ImageNet (Deng et al., 2009), afin d'extraire des caractéristiques d'une image de foule. Cette extraction (feature extraction) génère les informations d'une même image selon différentes échelles (multiple scales). Les caractéristiques de chaque échelle sont envoyées en entrée d'un réseau TFM (Top-down Feature Modulator) où elles sont assemblées pour générer des boites de limitation carrées correspondant à une échelle donnée. Expérimentalement, et selon la métrique de comptage Mean Absolute Error (MAE), les auteurs ont déterminé que la configuration optimale de LSC-CNN est l'usage de 4 réseaux TFM et la proposition de 3 boites de limitation différentes par chaque réseau TFM. Par la suite, la suppression des non-maximas (Non-Maximum Suppression (NMS)) rassemble les propositions provenant des 4 réseaux TFM pour n'élire qu'une seule détection. La vérité terrain est exprimée par des annotations sous forme de points qui désignent les centres des têtes des individus. Lors de la phase d'apprentissage, le module Grid Winner-Take-All (GWTA) est employé comme fonction de coût dans l'algorithme de rétropropagation. Ce module est remplacé par un module de fusion de prédictions lors de la phase d'évaluation.

(Sam et al., 2020) ont évalué LSC-CNN sur les jeux de données ShanghaiTech (Zhang et al., 2016c), UCF-QNRF (Idrees et al., 2018), UCF_CC_50 (Idrees et al., 2013), WorldExpo'10 (Zhang et al., 2015), le jeu de données de détection de véhicules TRANCOS (Guerrero-Gómez-Olmedo et al., 2015), et le jeu de données de détection de visages WiderFace (Yang et al., 2016). Pour évaluer les performances de localisation du modèle, les auteurs ont utilisé les métriques Grid Average Mean

5. Implémentation de LSC-CNN : https://github.com/val-iisc/lsc-cnn

Absolute Error (GAME) (Guerrero-Gómez-Olmedo et al., 2015) et Mean Localization Error (MLE), où ils arrivent à surpasser les performances de la méthode de base CSR-A tirée de CSR-Net (Li, Zhang et Chen, 2018). Ils ont utilisé la métrique mean Average Precision (mAP) pour évaluer sa capacité à déterminer les bonnes tailles des boites de limitation. Dans ce registre, ils arrivent à dépasser les performances de la méthode PSDNN (Liu et al., 2019). Cette dernière évaluation nous permet d'anticiper et de jauger la capacité de LSC-CNN à déterminer la position d'un individu par rapport à la caméra. Une information qui peut aider nos réseaux à mieux comprendre une scène de foule. Les performances en comptage ont été évaluées selon les métriques MAE et Mean Squarred Error (MSE). En comparant leurs performances à PSDNN, entre autres, ils sont dépassées par celle-ci dans le jeu de données ShanghaiTech Part-A (Zhang et al., 2016c).

Dans ce qui suit, nous présentons d'autres détecteurs de têtes, plus récents, mais qui sont sortis lors de l'élaboration des expériences liées à ce chapitre. Ces détecteurs sont ceux de (Abousamra et al., 2021) et (Wang et al., 2021). Leurs implémentations sont publiquement disponibles, mais faute de temps, nous n'avons pas pu les tester.

(Abousamra et al., 2021) proposent la méthode de détection de têtes appelée TopoCount. Cette approche utilise un réseau de type U-Net, muni d'un encodeur de type VGG-16, qui produit des cartes topologiques de probabilités. Les auteurs y introduisent une contrainte topologique permettant de réduire les erreurs de leur détecteur de têtes. Cette contrainte est renforcée par une fonction de coût basée sur la théorie de l'homologie persistente (Edelsbrunner et Harer, 2010). Ils intègrent leur contrainte à une méthode de comptage des individus dans une foule basée sur l'estimation de la densité afin d'en améliorer les performances. TopoCount semble réaliser de bons résultats sur les scènes de foules denses ou clairsemées. TopoCount supplante même les performances de LSC-CNN, en termes de l'accuracy et de la métrique GAME, sur les jeux de données UCF-QNRF et ShanghaiTech Part A et B (Zhang et al., 2016c). Toutefois, contrairement à LSC-CNN, les auteurs de TopoCount n'évaluent pas leur méthode sur sa capacité à estimer les bonnes tailles des boites de limitation. Produire des boites de limitation ayant des tailles appropriées peut aider nos méthodes de classification à estimer la position d'une tête par rapport à la caméra, et de ce fait, améliorer sa compréhension des mouvements opérant dans une scène de foule.

(Wang et al., 2021) proposent le réseau Crowd-SDNet entrainé exclusivement sur des annotations sous forme de points pour détecter les centres des objets et estimer leurs tailles. Le réseau auto-encodeur est composé d'un encodeur de type ResNet-50, pré-entraîné sur ImageNet (Krizhevsky, Sutskever et Hinton, 2012), et d'un décodeur muni de trois couches de convolutions. Le décodeur couplé avec une fonction de fusion de caractéristiques génère des cartes de caractéristiques de haute résolution. Les auteurs utilisent, avant le début de la phase d'apprentissage, une distribution localement uniforme (Locally-Uniform Distribution Assumption (LUDA)) qui propose une représentation initiale des tailles des objets, appelée Pseudo Objects Sizes, en se basant sur les annotations sous forme de points des données vérité-terrain (Ground Truth). Lors de la phase d'apprentissage, la représentation Pseudo Objects Sizes est améliorée à l'aide d'un schéma de perfectionnement basé sur la confiance et prenant en compte l'ordre (Confidence and Order-aware) sous la supervision d'une fonction de coût Crowdedness-aware. Par ailleurs, la localisation de centres des objets est supervisée à l'aide d'une fonction de coût d'entropie croisée focale. La phase d'apprentissage sert à améliorer la capacité du détecteur à localiser les objets, en trouvant leurs tailles, et à estimer leur nombre. La capacité de détection de Crowd-SDNet a été évaluée sur le jeu de données de reconnaissance faciale Widerface (Yang et al., 2016), selon le

protocole d'évaluation proposée par les auteurs de Widerface. Sa capacité de comptage a été évaluée, selon les métriques Mean Absolute Error (MAE) et Root Mean Squared Error (RMSE), sur les jeux de données de comptage de nombre de personnes ShanghaiTech (Zhang et al., 2016c), NWPU-Crowd (Wang et al., 2020), les jeux de données de détection de véhicules CARPK (Hsieh, Lin et Hsu, 2017) et PUCPR+ (De Almeida et al., 2015), et en utilisant la métrique Normalized MAE (NAE) sur WiderFace. Pour évaluer la capacité de localisation des centres des objets, ils ont utilisé les métriques Average Precision (AV) et Mean Localization Error (MLE) pour ShanghaiTech, et le protocole d'évaluation des auteurs de NWPU-Crowd, qui sont la précision, le rappel, et le F1-score, pour ce jeu de données. En comparant les performances de Crowd-SDNet en localisation et en comptage, entre autres, à celles de LSC-CNN et PSDNN (Liu et al., 2019), elles les surpassent dans les jeux de données WiderFace et ShanghaiTech Part A et B.

Dans un autre registre, nous nous inspirons des travaux de (Wang et al., 2016) pour mettre en place de nouveaux réseaux ThreeStream afin d'exploiter plusieurs déclinaisons d'une même information. (Wang et al., 2016) se basent sur les trajectoires des articulations d'un squelette humain pour caractériser des actions, à l'instar de (Ke et al., 2017) qui reconnaissent des actions en suivant les trajectoires des articulations d'un squelette humain ou de (De Smedt, Wannous et Vandeborre, 2016) qui réalisent la reconnaissance des gestes en suivant les trajectoires des articulations d'une main. (Wang et al., 2016) emploient des CNN et proposent d'extraire des cartes Joint Trajectory Maps (JTM) qui sont une projection des séquences 3D du mouvement du squelette sur un plan 2D. Dans ce cadre, la graduation de la couleur décrit l'évolution sur l'axe temporel désignant les trajectoires que prennent chaque partie du squelette : ses membres, ses articulations. Trois cartes JTM sont extraites de chaque clip d'une action individuelle afin de visualiser le squelette, et l'action qu'il accomplit, sous trois angles différents. Un réseau ThreeStream-CNNs est utilisé, par la suite, pour caractériser une action à partir des trois encodages 2D sous forme de cartes JTMs. Afin de reproduire ces différents angles de vue, les auteurs procèdent eux-mêmes à une rotation des squelettes extraits des clips vidéo. Au travers de cette démarche, les auteurs visent à pallier la faiblesse des CNNs face aux variations des angles de vue. Les réseaux CNNs utilisés sont les réseaux AlexNet (Krizhevsky, Sutskever et Hinton, 2012) pré-entrainés sur le jeu de données ImageNet (Russakovsky et al., 2015). Les trois branches du réseau sont fusionnées à l'aide d'une opération de multiplication avant l'étape de classification. (Wang et al., 2016) évaluent leur approche, en termes d'accuracy, sur les jeux de données NTU RGB+D (Shahroudy et al., 2016), MSRC-12 Kinect Gesture (Fothergill et al., 2012), G3D (Bloom, Makris et Argyriou, 2012), et UTD-MHAD (Chen, Jafari et Kehtarnavaz, 2015). Ils confirment à travers ces évaluations l'apport de la rotation, de la couleur graduée pour représenter les trajectoires des parties d'un squelette, de la fusion via la multiplication dans la reconnaissance d'actions. Par ailleurs, les performances de leur approche surpassent des méthodes récentes telles que Cov3DJ (Hussein et al., 2013), SOS (Hou et al., 2016), ELC-KSVD (Zhou et al., 2014) dans les jeux de données G3D, MSRC-12 Kinect Gesture, UTD-MHAD. Pour le jeu de données NTU RGB+D, ils supplantent, entre autres, ST-LSTM + Trust Gate (Liu et al., 2016a) et Part-aware LSTM (Shahroudy et al., 2016).

La méthode de (Wang et al., 2016) nous incite à mettre en place des réseaux ThreeStream-I3D qui sont une extension des réseaux TwoStream-I3D, vus dans les sections précédentes, et dont la troisième branche a vocation à exploiter les cartes de détection de têtes. Toutefois, nous trouvons l'encodage temporel traduit par les cartes JTM inadapté aux mouvements chaotiques qui se déroulent dans une scène de foule.

Dans le contexte d'une scène de foule, nous trouvons les cartes JTMs inadaptées. Une projection en 2D d'une scène de foule, exprimée selon les dimensions spatio-temporelles, risque de rendre une scène de foule illisible au vu du très grand nombre d'acteurs qui s'y trouvent.

2.4.2 Exploitation des détections de têtes pour la classification des scènes de foule

Dans cette section, nous détaillons les types de cartes de détection de têtes (Heads Detection Maps (HDM)) que nous avons extraites à l'aide du détecteur LSC-CNN sur le jeu de données Crowd-11. Ces cartes HDM extraites sont soit employées comme source principale d'information pour les méthodes de classification afin d'analyser les scènes de foule ou bien elles épaulent d'autres types d'informations tels que les clips vidéo en RVB ou leurs versions en flux optique obtenues à l'aide de l'algorithme TVL1.

Les méthodes de classification envisagées dans cette section sont en partie des méthodes déjà vues précédemment telles que les réseaux C3D, I3D, et TwoStream-I3D. Nous y ajoutons le réseau ResNet 3D (R3D). Par ailleurs, nous mettons nous-mêmes en place un nouveau réseau ThreeStream-I3D pour les besoins de cette partie du chapitre.

2.4.2.1 Extraction des cartes de détection de têtes à l'aide du détecteur LSC-CNN

Les cartes de détection de têtes HDM sont extraites à l'aide du détecteur LSC-CNN appliqué sur les clips RVB du jeu de données Crowd-11. Nous extrayons trois types de cartes de détection de têtes :

1. Une carte de détection de têtes HDM binarisée **Heads detection maps WithOut Scales With Binarization (HWOS_WB)** : où les centres des têtes sont représentés par des points blancs : soient des pixels de valeur 255, et le reste en noir : soient des pixels de valeur 0.
2. Une carte de détection de têtes HDM qui prend en compte les 4 échelles du détecteur avec application d'une égalisation d'histogramme **Heads detection maps With Scales With Histogram Equalization (HWS_WHE)** : Les centres des têtes sont en 4 nuances de gris différentes. Le reste de l'image est en noir. Les 4 échelles permettent de donner un aperçu sur la taille d'une tête et en même temps son éloignement de la caméra. L'égalisation d'histogramme vise à faire mieux ressortir la différence d'intensité entre les 4 échelles car à la base la valeur de l'intensité maximale n'est pas très élevée et la différence entre les 4 n'est pas perceptible à l'oeil nu.
3. Une carte de détection de têtes HDM qui prend en compte les 4 échelles du détecteur sans égalisation d'histogramme **Heads detection maps With Scales WithOut Histogram Equalization (HWS_WOHE)** : Contrairement à ce que nous faisons pour les cartes **HWS_WHE**, nous n'appliquons pas ici d'égalisation d'histogramme.

2.4.2.2 Réseaux employés pour la classification

Nous utilisons pour la classification les réseaux C3D, I3D, TwoStream I3D, ThreeStream I3D et les réseaux ResNet 3D. Nous allons d'abord présenter les réseaux que nous n'avons pas encore vus dans ce chapitre avant de détailler le type de données sur lesquelles ils ont été entraînés.

2.4.2.2.1 Réseau ResNet 3D Le réseau ResNet 3D, proposé par (Hara, Kataoka et Satoh, 2017), est constitué de plusieurs blocs résiduels. Chaque bloc résiduel est constitué de deux couches de convolutions 3D. Nous choisissons la version à 34 couches cachées du fait de ses bonnes performances sur le jeu de données de reconnaissance d'actions Sports-1m. Cette version de l'architecture R3D est constituée d'une première couche de convolutions 3D suivie de 16 blocs résiduels, et se termine par une couche FCN avant la fonction de classification Softmax.

2.4.2.2.2 Réseau nouvellement créé : ThreeStream Inflated 3D Nous illustrons le réseau que nous mettons en place pour les besoins de cette section dans la figure 2.8. Ce réseau dispose de trois branches du réseau de type Inflated 3D. Pour rappel, chaque branche de type Inflated 3D dispose d'une base de 2 couches de convolutions 3D suivie par 9 modules Inception qui débouchent sur un Average Pooling. Les trois branches sont par la suite fusionnées à travers une opération de concaténation précédant une opération de classification de type Softmax.

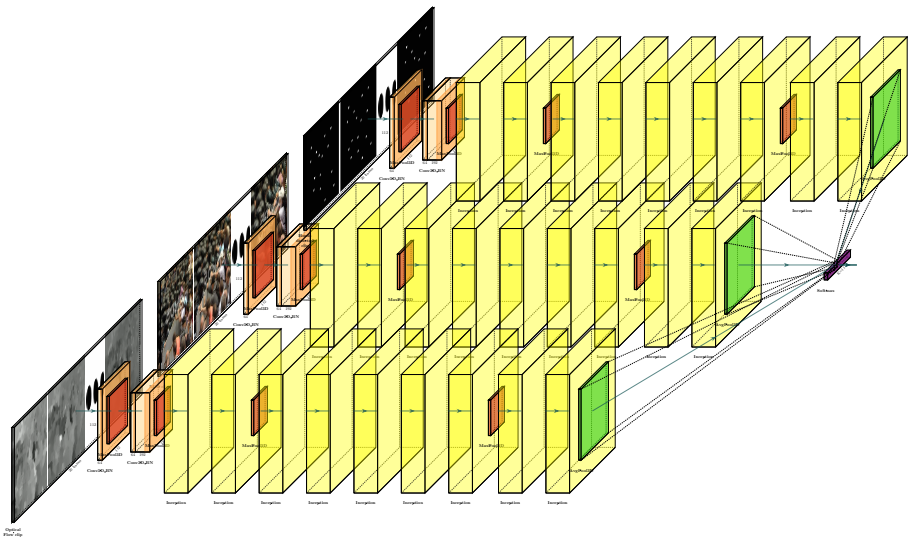


Figure 2.8 – Illustration de l'architecture ThreeStream Inflated 3D

Les illustrations des architectures, Figures 2.2, 2.3, 2.4, 2.8, ont été réalisées à l'aide de l'outil PlotNeuralNet [6] (Iqbal, 2018).

2.4.2.2.3 Données passées en entrée lors des phases d'apprentissage, validation, et test Nous entraînons de zéro ou ajustons les réseaux à branche unique I3D et C3D soit sur les clips vidéo en RVB, ou bien les cartes de détection de têtes Hwos_wb, Hws_whe, ou Hws_wohe. Dans le cas du réseau R3D, comme nous n'avons pas trouvé de poids pré-entrainés, publiquement disponibles, nous avons décidé de seulement considérer la situation de l'entraînement de zéro sur les 4 types de données

6. Outil PlotNeuralNet : https://github.com/HarisIqbal88/PlotNeuralNet

évoqués. Pour rappel, les réseaux C3D ajustés sont pré-entraînés sur le jeu de données Sports-1m et les réseaux I3D ajustés sont pré-entraînés sur ImageNet + Kinetics 400.

Nous entraînons de zéro ou ajustons les réseaux à deux branches TwoStream I3D. La première branche du réseau TwoStream I3D reçoit toujours en entrée les versions RVB des clips de Crowd-11. La deuxième branche de ce réseau peut soit recevoir des clips en version flux optique ou bien un des trois types de cartes de détection de têtes Hwos_wb, Hws_whe, ou Hws_wohe. Si la deuxième branche du réseau TwoStream est ajustée sur les clips en version flux optique, elle aura été pré-entraînée sur la version flux optique du jeu de données Kinetics 400 inclus dans le couple (ImageNet + Kinetics 400). Toutefois, si la deuxième branche du réseau TwoStream s'ajuste sur les cartes de détection, nous utilisons la branche pré-entraînée sur la version RVB du jeu de données Kinetics 400 incluse dans le couple (ImageNet + Kinetics 400).

Nous entraînons à partir de zéro ou ajustons les réseaux à trois branches ThreeStream I3D. La première branche de ce réseau reçoit toujours des clips en version RVB. La deuxième branche reçoit des clips en version flux optique. La troisième branche de ce réseau peut recevoir une des cartes de détection de têtes Hwos_wb, Hws_whe, ou Hws_wohe. Dans le cas de l'apprentissage par transfert la troisième branche est pré-entraînée sur les jeux de données ImageNet + Kinetics 400.

2.4.2.3 Procédure de recherche des hyperparamètres de réseaux supplémentaires de type perceptron multicouche

Afin de compléter la procédure de classification de chaque réseau, nous avons supplémenté chaque modèle étudié, dans cette section, par un perceptron multicouche. Nous utilisons une procédure de recherche exhaustive GridSearch pour trouver les hyperparamètres adéquats pour chaque modèle entraîné ou ajusté. Pour chaque réseau CNN déjà entraîné sur un des 5 ensembles d'apprentissage du jeu de données Crowd-11, un perceptron multicouche s'entraîne sur les distributions de probabilités prédites sur ce même ensemble d'apprentissage par la fonction de classification softmax du réseau de neurones convolutifs.

2.4.3 Expériences

Pour l'extraction des cartes de détection de têtes du jeu de données Crowd-11, nous avons utilisé le modèle LSCCNN pré-entraîné sur le jeu de données de comptage de foule UCF-QNRF. Le jeu de données UCF-QNRF, d'après la description qui en est faite par (Abousamra et al., 2021), est un jeu de données d'images de foule volumineux et très varié destiné principalement au comptage du nombre de personnes dans une scène de foule. Du fait des qualités de ce jeu de données, nous avons décidé de maximiser les chances de notre détecteur de têtes en choisissant ce type de pré-entraînement pour la prédiction des cartes HDM.

2.4.3.1 Apprentissage et évaluation des réseaux CNN sur les cartes de détection de têtes

Nous comparons les performances de modèles ajustés ou entraînés de zéro sur Crowd-11 ayant reçu en entrée un des types de clips HDM à savoir Hwos_wb, Hws_whe, Hws_wohe (c.f. la définition qui est donnée à chacun d'entre eux dans la section 2.4.2.1), ou bien des clips de type RVB ou flux optique. Pour ce dernier type, nous n'entraînons pas nos modèles uniquement dessus car nous avons déjà pu

voir dans nos expériences préliminaires (c.f. section 2.2) l'apport des clips en flux optique dans la classification des scènes de foule. Dans nos expériences, le flux optique épaule à chaque fois un des quatre types de clips énoncés plus haut.

Les modèles sont entraînés ou ajustés pendant 15 époques. Par rapport à ce qui a été fait dans la section 2.2, nous avons décidé de réduire le nombre des époques d'entraînement car nous avons décidé d'entreprendre une multitude d'expériences qui nécessitent un très grand nombre d'heures de calcul. Il est à noter qu'après 15 époques d'entraînement les modèles auront déjà bien appris les différentes classes du jeu de données. Toutefois, nous savons bien qu'une telle réduction peut impacter négativement les performances des modèles.

Les paramètres des données en entrée sont les suivants :

— Les modèles C3D sont entraînés/ajustés sur des lots de 30 clips dont les dimensions sont 16 frames × (112 × 112) pixels.
— Les modèles R3D avec 34 couches sont entraînés de zéro sur des lots de 32 clips dont les dimensions sont 16 frames × (112 × 112) pixels.
— Les modèles I3D sont entraînés/ajustés sur des lots de 16 clips dont les dimensions sont 20 frames × (224 × 224) pixels.
— Les modèles TwoStream I3D et ThreeStream I3D sont entraînés/ajustés sur des lots de 8 clips dont les dimensions sont 20 frames × (224 × 224) pixels.

2.4.3.1.1 Tour d'horizon des hyperparamètres d'apprentissage affectés à chaque réseau : Lors de l'apprentissage des réseaux I3D, TwoStream I3D, et ThreeStream I3D, les modèles sont optimisés à l'aide de l'algorithme de la descente du gradient stochastique (SGD) avec un taux d'apprentissage initial de 0.003 et un momentum de 0.9. Le taux d'apprentissage (LR) est divisé par 10 dès que l'apprentissage atteint un plateau.

Lors de l'apprentissage des modèles C3D, ils sont optimisés à l'aide de la descente du gradient stochastique (SGD) avec un taux d'apprentissage initial de 0.003, et le taux d'apprentissage est divisé par 10 toutes les 4 époques, conformément à l'approche adoptée par (Tran et al., 2015).

Lors de l'apprentissage modèles R3D, ils sont optimisés à l'aide de l'algorithme Adam. Ici, le taux d'apprentissage initial est de 0.001 et le taux d'apprentissage est divisé une seule fois par 10 dès que l'apprentissage atteint un plateau.

Modèle CNN	**Type de clip en entrée**	μ	σ
C3D entraîné de zéro	RVB	32.02	3.07
	Hwos_wb	25.55	1.98
	Hws_whe	26.44	1.06
	Hws_wohe	21.64	0.97
C3D ajusté	RVB	57.61	0.52
	Hwos_wb	43.94	1.67
	Hws_whe	43.60	1.20
	Hws_wohe	41.04	2.39
I3D entraîné de zéro	RVB	53.92	2.30
	Hwos_wb	48.10	1.83
	Hws_whe	45.91	1.48
	Hws_wohe	43.30	1.95
I3D ajusté	RVB	60.74	1.95
	Hwos_wb	45.84	3.13
	Hws_whe	43.97	1.38
	Hws_wohe	41.99	1.32
R3D 34 couches entraîné de zéro	RVB	41.87	2.63
	Hwos_wb	43.07	1.81
	Hws_whe	42.89	1.71
	Hws_wohe	40.36	1.83
TwoStream I3D entraîné de zéro	(RVB, OF)	47.76	5.32
	(RVB, Hwos_wb)	51.17	4.48
	(RVB, Hws_whe)	52.20	4.27
	(RVB, Hws_wohe)	52.47	3.03
TwoStream I3D ajusté	**(RVB, OF)**	**65.98**	**4.79**
	(RVB, Hwos_wb)	60.56	1.55
	(RVB, Hws_whe)	59.24	2.73
	(RVB, Hws_wohe)	60.81	1.98
ThreeStream I3D entraîné de zéro	(RVB, OF, Hwos_wb)	53.70	3.43
	(RVB, OF, Hws_whe)	50.19	3.20
	(RVB, OF, Hws_wohe)	52.49	1.80
ThreeStream I3D ajusté	**(RVB, OF, Hwos_wb)**	**66.09**	**4.16**
	(RVB, OF, Hws_whe)	64.19	4.54
	(RVB, OF, Hws_wohe)	64.68	5.39

TABLE 2.4 – Comparaison entre les modèles de Réseau de Neurones Convolutifs entraînés ou ajustés à classer les vidéos de foule selon le type de données en entrée

2.4.3.2 Apprentissage et évaluation des réseaux MLP sur les distributions de probabilités

Nous comparons les performances des perceptrons multicouches obtenues en s'entraînant sur les probabilités prédites par les fonctions de classification softmax des modèles de réseaux de neurones convolutifs ajustés ou entraînés de zéro sur Crowd-11 ayant reçu en entrée un des types de clips HDM à savoir Hwos_wb, Hws_whe, Hws_wohe (c.f. la définition qui est donnée à chacun d'entre eux dans la section 2.4.2.1), ou bien des clips de type RVB ou flux optique.

Les perceptrons multicouches sont entraînés pour 200 époques. Les hyperparamètres des perceptrons multicouches sont recherchés pour chaque ensemble d'apprentissage de chaque division de Crowd-11 à l'aide de l'algorithme de recherche exhaustive.

L'espace de paramètres P exploré par l'algorithme de recherche, inspiré d'un thread sur StackExchange[7], est comme suit :

7. Parameter space for GridSearch : https://datascience.stackexchange.com/questions/36049/how-to-adjust-the-hyperparameters-of-mlp-classifier-to-get-more-perfect-performa/36087#36087

— Nombre de neurones par couche cachée : $\{(20), (20, 20), (20, 20, 20), (50), (50, 50), (50, 50, 50), (100), (100, 100), (100, 100, 100), (20, 50), (20, 50, 20), (50, 100, 50), (50, 50, 50, 50), (50, 100, 100, 50)\}$;
— Fonctions d'activation : $\{ReLU, TanH\}$;
— Fonctions d'optimisation : $\{SGD, Adam\}$;
— Taux d'apprentissage initiaux : $\{0.05, 10^{-3}, 10^{-4}\}$;
— Types de taux d'apprentissage : $\{Constant, adaptatif\}$.

Afin d'uniformiser les hyperparamètres du perceptron multicouche liés à chaque modèle de réseau de neurones convolutifs, nous avons recherché une combinaison unique des hyperparamètres qui maximisent l'accuracy sur les ensembles d'apprentissages des 5 divisions de Crowd-11. Pour cela, nous avons appliqué la procédure explicitée dans l'équation 2.2 pour trouver la combinaison des hyperparamètres de l'espace des paramètres P maximisant les performances du perceptron multicouche associé à un modèle m.

$$ACC = \begin{bmatrix} a_{11} & \dots & a_{1k} & \dots & a_{1K} \\ \vdots & \ddots & \vdots & & \vdots \\ a_{i1} & \dots & a_{ik} & \dots & a_{iK} \\ \vdots & & \vdots & \ddots & \vdots \\ a_{I1} & \dots & a_{Ik} & \dots & a_{IK} \end{bmatrix} \tag{2.1}$$

$$best\_hyperparameters\_combination_m = \underset{i}{\operatorname{argmax}}(\sum_{k}^{K} ACC) \tag{2.2}$$

— m représente un des 34 modèles de réseaux ayant son propre type de données en entrée ;
— k étant le numéro de la division courante ;
— K étant le nombre de divisions (splits) égal à 5 ;
— i représente une combinaison des hyperparamètres de l'espace des paramètres P ;
— I représente le nombre maximal des combinaisons des hyperparamètres de l'espace des paramètres explorés par l'algorithme de recherche ;
— ACC étant la matrice des accuracies d'entraînement. Chaque élément a_{ik} de la matrice est obtenu après l'apprentissage d'un perceptron multicouche en utilisant une combinaison d'hyperparamètres i sur une distribution de probabilités d'un ensemble d'apprentissage de Crowd-11 tiré d'un split k.

Les hyperparamètres trouvés ainsi que l'accuracy moyenne obtenue pour chaque modèle sont illustrés dans la table 2.5.

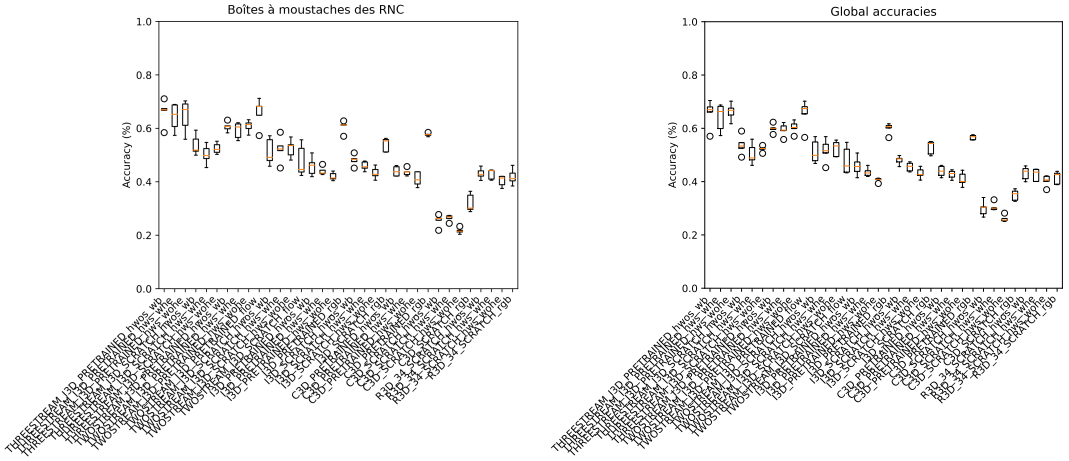


FIGURE 2.9 – Boîtes à moustaches des accuracies de chaque modèle à l'évaluation sur l'ensemble de test des 5 splits de Crowd-11. À droite est l'illustration des modèles RNC. À gauche est l'illustration des modèles PMC

Modèle PMC	**Type de clips**	**Nombre de couches cachées**	**Fonction d'activation**	**Fonction d'optimisation**	**Taux d'apprentissage initial**	**Type du taux d'apprentissage**	μ	σ
PMC C3D entraîné de zéro	RVB	(50, 50, 50)	TanH	Adam	10^{-4}	Constant	34.96	1.85
	Hwos_wb	(100, 100, 100)	TanH	Adam	10^{-4}	Constant	29.92	2.52
	Hws_whe	(100, 100, 100)	TanH	Adam	10^{-3}	Constant	30.48	1.38
	Hws_wohe	(50)	ReLU	Adam	10^{-4}	Constant	26.06	1.11
PMC C3D pré-entraîné	RVB	(100, 100)	TanH	Adam	10^{-3}	Constant	56.61	0.79
	Hwos_wb	(50, 50)	TanH	Adam	10^{-4}	Constant	43.90	1.80
	Hws_whe	(50, 50)	TanH	Adam	10^{-3}	Constant	42.70	1.43
	Hws_wohe	(100, 100)	TanH	Adam	10^{-4}	Constant	40.94	2.29
PMC I3D entraîné de zéro	RVB	(50)	ReLU	Adam	10^{-4}	Constant	52.84	2.30
	Hwos_wb	(50, 50, 50)	TanH	Adam	0.05	Constant	48.01	1.41
	Hws_whe	(100, 100, 100)	TanH	Adam	0.05	Constant	45.54	1.48
	Hws_wohe	(100, 100)	TanH	Adam	10^{-4}	Constant	43.14	1.77
PMC I3D pré-entraîné	RVB	(20, 20)	TanH	Adam	10^{-4}	Constant	60.01	1.82
	Hwos_wb	(20)	TanH	Adam	0.05	Constant	45.81	3.14
	Hws_whe	(50, 50, 50)	TanH	Adam	10^{-4}	Constant	43.54	1.47
	Hws_wohe	(100, 100)	TanH	Adam	10^{-4}	Constant	40.66	0.70
PMC R3D entraîné de zéro	RVB	(50)	TanH	Adam	0.05	Constant	41.50	2.12
	Hwos_wb	(100)	TanH	Adam	10^{-3}	Constant	43.14	2.25
	Hws_whe	(100)	ReLU	Adam	0.05	Constant	42.58	2.09
	Hws_wohe	(100, 100)	TanH	Adam	10^{-3}	Constant	40.37	1.86
PMC 2S I3D entraîné de zéro	(RVB, OF)	(50, 50)	TanH	Adam	10^{-3}	Constant	47.93	4.66
	(RVB, Hwos_wb)	(50)	ReLU	Adam	0.05	Constant	51.26	3.94
	(RVB, Hws_whe)	(20)	TanH	Adam	10^{-3}	Constant	51.69	3.86
	(RVB, Hws_wohe)	(50)	TanH	Adam	0.05	Constant	52.47	2.62
PMC 2S I3D pré-entraîné	**(RVB, OF)**	**(20, 20)**	**ReLU**	**Adam**	**0.05**	**Constant**	**65.54**	**4.72**
	(RVB, Hwos_wb)	(100)	ReLU	Adam	0.05	Constant	59.93	1.46
	(RVB, Hws_whe)	(50, 50)	TanH	Adam	10^{-4}	Constant	59.47	2.14
	(RVB, Hws_wohe)	(50)	ReLU	Adam	0.05	Constant	60.40	2.03
PMC 3S I3D entraîné de zéro	(RVB, OF, Hwos_wb)	(20)	ReLU	Adam	10^{-3}	Constant	53.73	3.17
	(RVB, OF, Hws_whe)	(20)	ReLU	Adam	10^{-3}	Constant	50.36	3.51
	(RVB, OF, Hws_wohe)	(100)	ReLU	Adam	10^{-3}	Constant	52.28	0.95
PMC 3S I3D pré-entraîné	**(RVB, OF, Hwos_wb)**	**(20)**	**TanH**	**Adam**	10^{-4}	**Constant**	**65.72**	**4.58**
	(RVB, OF, Hws_whe)	(20, 50)	TanH	Adam	0.05	Constant	64.09	4.64
	(RVB, OF, Hws_wohe)	**(100)**	**ReLU**	**Adam**	**0.05**	**Constant**	**66.24**	**2.80**

TABLE 2.5 – Comparaison entre les perceptrons multicouches trouvés via l'algorithme de recherche exhaustive pour chaque Réseau de Neurones Convolutifs

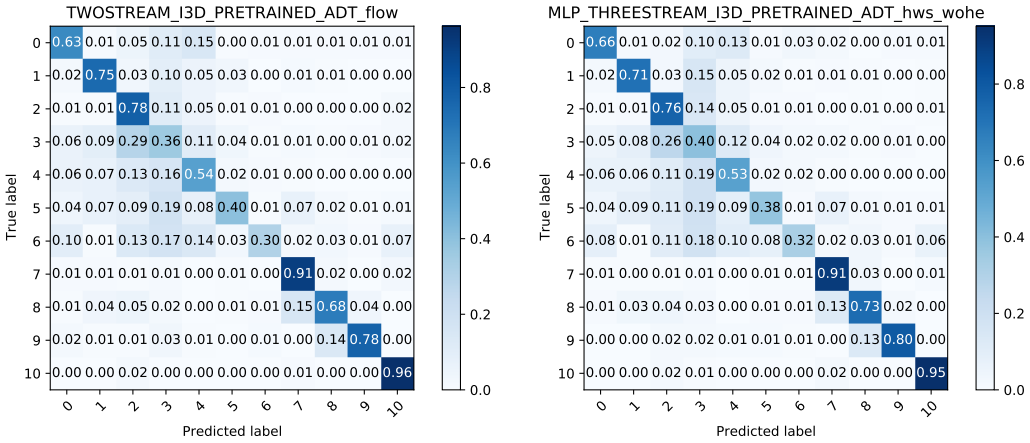


FIGURE 2.10 – Comparaison entre les matrices de confusion globales des réseaux 2S-I3D (RVB, OF) ajusté et 3S-I3D (RVB, OF, Hws_wohe)

2.4.4 Discussion des résultats

Les résultats obtenus par les Réseaux de Neurones Convolutifs (RNC) sont illustrés dans la table 2.4. Les résultats obtenus par les Perceptrons Multicouches (PMC) complémentaires sont illustrés dans la table 2.5. Une comparaison des performances, selon l'accuracy, des modèles RNC et PMC est illustrée dans les boîtes à moustaches de la figure 2.9. Dans la table 2.4, nous apercevons tout d'abord que dans une majeure partie des cas de figure, l'information apportée par les clips de type RVB surpasse largement en qualité celle apportée par les clips de type cartes de détection de têtes HDM. Dans les cartes de détection de têtes, la simplicité des cartes binarisées apporte le plus souvent davantage d'informations que des cartes plus travaillées telles que les cartes contenant une information de profondeur et leurs versions améliorées via l'égalisation de l'histogramme. Nous constatons à partir de la table 2.4 que le réseau 2S-I3D ajusté ayant bénéficié des flux optiques pour sa seconde branche obtient des performances assez proches de celles des réseaux 3S-I3D ajustés. La confrontation est rude entre ce réseau 2S-I3D, qui obtient en moyenne 65.98% d'accuracy avec +- un écart-type de 4.79%, et le réseau 3S-I3D s'étant ajusté sur le triplet contenant des clips de cartes de détection de têtes binarisés, qui obtient en moyenne 66.09% d'accuracy avec +- un écart-type de 4.16%. Même si ce réseau 3S-I3D prend le dessus avec une légère avance, on peut qualifier ces performances de quasi-similaires entre les réseaux ajustés 2S-I3D (RVB, OF) et 3S-I3D (RVB, OF, HWOS_WB).

Dans la table 2.5, nous voyons que pour que chaque Réseau de Neurones Convolutifs (RNC) un Perceptron Multicouche (PMC) a été trouvé via l'algorithme de recherche exhaustive. Les performances, en termes d'accuracy, entre les RNC et les PMC ne sont pas radicalement différents. Nous constatons une légère baisse des performances lors de l'application des PMC entraînés sur les prédictions des probabilités des RNC sur leurs ensembles d'apprentissages respectifs. Toutefois, nous voyons que le PMC du réseau ajusté 3S-I3D (RVB, OF, Hws_wohe) obtient, à hauteur de 66.24% d'accuracy avec +- un écart-type de 2.8, un score qui surpasse celui du réseau brut ajusté 2S-I3D (RVB, OF).

Afin d'avoir tous les éléments en tête avant de procéder à l'interprétation des matrices de confusion exposées dans la figure A.19, il est utile de rappeler l'indice attribué à chaque classe du jeu de données Crowd-11 :

0. **Gas Free :** Individus marchant dans toutes les directions sans rencontrer d'obstacles.
1. **Gas Jammed :** Foule congestionnée.
2. **Laminar Flow :** Individus marchant dans une seule direction.
3. **Turbulent Flow :** Foule marchant dans une seule direction perturbée par un individu marchant à contresens.
4. **Crossing Flows :** Deux foules qui se croisent.
5. **Merging Flows :** Deux foules qui convergent.
6. **Diverging Flow :** Une foule qui se subdivise en deux foules.
7. **Static Calm :** Une foule d'individus statiques et calmes.
8. **Static Agitated :** Une foule d'individus statiques et agités.
9. **Interacting Crowd :** Deux foules d'individus qui s'opposent. Cette classe contient des scènes de conflits.
10. **No Crowd :** Aucune présence humaine dans la scène.

Nous constatons que l'ajout d'un Perceptron Multicouche complémentaire à un Réseau de Neurones Convolutifs a tendance à diffuser davantage l'accuracy du réseau sur les autres classes, même si cela ne mène pas forcément à une amélioration des performances en termes d'accuracy. Cette tendance globale se vérifie fortement avec les réseaux C3D entraînés de zéro. Très souvent, il arrive que des réseaux améliorent légèrement leurs performances dans les classes où ils se trouvaient en difficulté avant l'ajout d'un PMC au détriment des classes où ils obtenaient de bons résultats.

Nous observons que notre meilleur réseau en termes d'accuracy, qui est le réseau ajusté 3S-I3D ayant comme entrée le triplet (RVB, OF, Hws_wohe) et bénéficiant de la surcouche PMC, éprouve le plus de difficultés sur les classes Turbulent Flow $\longrightarrow$ 40%, Merging Flows$\longrightarrow$ 38%, Diverging Flows$\longrightarrow$ 32%. Dans ce registre, il ne fait pas mieux que le réseau 2S-I3D ajusté ayant comme entrée le couple (RVB, OF) ou l'homologue, de ce dernier, bénéficiant de la surcouche PMC. L'étude comparative des matrices de confusion des modèles 3S-I3D ajusté + PMC (RVB, OF, Hws_wohe) et 2S-I3D ajusté (RVB, OF), illustrée dans la figure 2.10, montre que les accuracies des classes problématiques, qui sont en dessous de 50%, s'améliorent de 1.33% en moyenne pour le modèle 3S-I3D ajusté + PMC : +4% pour Turbulent Flow, -2% pour Merging Flow, +2% pour Diverging Flow.

Au vu des maigres gains en performances que réalisent les réseaux ThreeStream-I3D par rapport à leurs homologues TwoStream-I3D, nous trouvons que les moyens mis en oeuvre pour les mettre en place ne valent pas encore la peine de les déployer à grande échelle dans les systèmes d'analyse des scènes de foule.

Toutefois, il serait intéressant d'explorer l'apport de détecteurs plus récents (Abousamra et al., 2021 ; Wang et al., 2021) dans la classification des scènes de foule, ou mieux encore, penser à les évaluer selon leur capacité à produire des détections stables à travers plusieurs frames successives. En effet, bien que LSC-CNN réalise des détections globalement satisfaisantes sur une seule frame, les coordonnées de ces détections varient significativement sur deux frames successives ce qui nuit à la performance des méthodes de classification.

Au vu de notre étude de l'état-de-l'art dans la détection de têtes (c.f. section 2.4.1), les détecteurs de têtes sont le plus souvent évalués selon leur capacité à maximiser le nombre de détections dans une image fixe. Cependant, ils ne sont pas évalués selon leur capacité à fidéliser ces détections tout le long d'un clip vidéo d'une scène de foule, tout simplement car les jeux de données les plus utilisés pour se mesurer aux détecteurs de têtes de l'État-de-l'art sont des jeux de données d'images fixes à

l'instar de ShanghaiTech Part A et B (Zhang et al., 2016c), UCF-QNRF (Idrees et al., 2018), UCF_CC_50 (Idrees et al., 2013), etc.

Le jeu de données Crowd-11 peut être employé pour créer des détecteurs de têtes et des algorithmes de suivi de trajectoires pour des scènes de foule très denses. Dans ce registre, l'emploi de la métrique Intersection over Union (IoU) pour évaluer les détections de têtes entre tous les couples de frames successives d'une vidéo de foule peut être à envisager. Une détection de têtes correctement mise en oeuvre tout le long d'un clip d'une scène de foule peut donner lieu à de nouvelles approches de suivi de trajectoires et d'analyse des dynamiques des scènes de foule très denses.

2.5 Conclusion

Dans ce chapitre, nous avons, dans un premier temps, étudié la capacité du réseau *TwoStream Inflated 3D* à tirer profit de son pré-entraînement sur les jeux de données ImageNet et Kinetics 400 pour la classification des comportements de foule sur le jeu de données Crowd-11. Après avoir transféré les connaissances apprises des jeux de données sources vers le jeu de données cible, le modèle produit surpasse l'état-de-l'art, sur Crowd-11, avec une marge conséquente de $\approx 10\%$ d'accuracy. Ce bond en performance ne se vérifie pas sur le jeu de données de reconnaissance d'actions UCF Crime. Déjà difficile à la base, le jeu de données UCF Crime nécessite un pré-traitement assez poussé avant d'y appliquer une méthode de classification. Dans ce registre, il serait nécessaire de s'inspirer des travaux réalisés dans l'État-de-l'art lié à UCF Crime exploré dans la section 2.3.1.

Nous avons, dans un second temps, étudié l'apport des cartes de densité pour la classification des scènes de foule. Pour ce faire, nous avons comparé des réseaux à une seule branche qui ont été entraînés ou ajustés sur des clips de cartes de densité ou des clips en RVB, et avons comparé par la suite des réseaux à deux branches et avons implémenté des réseaux à trois branches de notre propre conception. Suite à nos expériences, il s'avère que des réseaux à trois branches obtiennent de bons résultats. Ce constat se confirme avec les réseaux à trois branches complétés par des Perceptrons Multicouches. Toutefois, les gains en performance, certes encourageants, ne sont pas assez convaincants pour disqualifier totalement les réseaux TwoStream-I3D vus dans la première partie de ce chapitre. Néanmoins, ces gains en performance invitent la recherche en analyse de foule à persévérer dans l'étude plus approfondie de l'apport des cartes de densité dans l'analyse des comportements de foule.

À ce stade de nos travaux, les approches de classification des scènes de foule, étudiées dans ce chapitre, ne peuvent pas être considérées comme des outils de classification efficaces pour la gestion des mouvements de foule. Sur la base des résultats que nous avons obtenus, nous avons vu dans le chapitre 3 dans quelle mesure les approches suivantes améliorent nos résultats :

- L'augmentation des données vidéo ;
- Usage de méthodes ensemblistes de classification ;
- Mélange de différents réseaux lors de la classification ensembliste des scènes de foule.

Chapitre 3

Apport de l'apprentissage ensembliste pour la classification des scènes de foule

3.1 Introduction

(Dupont, Tobias et Luvison, 2017) ont entraîné des modèles pour classer des scènes de foule issues de Crowd-11. Le modèle qui obtient les meilleurs résultats de classification dans leur article est issu de l'architecture C3D (Tran et al., 2015). Dans le chapitre précédent, nous avons montré que nous atteignons de meilleures performances en employant un modèle issu de l'architecture TwoStream Inflated 3D (2S-I3D) (Bendali-Braham et al., 2019). Des modèles issus de cette architecture dépassent déjà les performances des modèles C3D sur des jeux de données de reconnaissance d'actions UCF-101 et HMDB-51 (Carreira et Zisserman, 2017).

Dans ce chapitre, nous visons à améliorer les performances de classification sur Crowd-11, en constituant des ensembles de modèles. Dans la section 3.2.1, nous avons présenté des méthodes ensemblistes appliquées à la classification de vidéos de manière générale ou plus spécifiquement à l'analyse de foules. Dans la section 3.2.2, nous avons introduit nos approches ensemblistes. Nous avons poursuivi, dans la section 3.2.3, par une présentation de nos différentes expériences. Tout d'abord, nous avons mené des expériences pour comparer des ensembles de modèles homogènes, dans la section 3.2.3.1. Ensuite, nous avons recherché la meilleure technique de pondération des modèles individuels constituant les ensembles de modèles, dans la section 3.2.3.2. Enfin, dans la section 3.2.3.4, nous avons évalué toutes les combinaisons possibles d'ensembles homogènes donnant lieu à des ensembles globaux de modèles ayant différentes architectures afin d'élire le modèle global qui réalise les meilleures performances. Nous avons terminé par une discussion des résultats que nous avons obtenus, une conclusion globale et des perspectives.

3.2 Apport de l'apprentissage ensembliste pour la classification des mouvements de foule

3.2.1 Contexte (État-de-l'art)

Les méthodes ensemblistes réalisent de très bonnes performances dans plusieurs tâches liées à l'apprentissage automatique (Fawaz et al., 2019). (Zhou, 2009) divise les méthodes ensemblistes en trois grandes familles représentatives :

— Boosting, illustrée par son algorithme le plus connu AdaBoost (Freund et Schapire, 1995), qui consiste à apprendre T modèles en associant à chaque

fois des poids différents aux exemples d'apprentissage. Au début similaires, ces poids changent à chaque itération t de l'algorithme AdaBoost en prenant en compte l'erreur obtenue d'un modèle entraîné à l'itération $t-1$. À la fin, un vote majoritaire pondéré est utilisé pour combiner les décisions des T modèles.
— Bagging, contraction de Bootstrap Aggregating, où des méthodes statistiques sont entraînées sur des échantillons créés par des échantillonnages Bootstrap (Efron, 1992). Par la suite, ces méthodes sont combinées dans un ensemble par un vote majoritaire.
— Stacking, où différentes méthodes statistiques sont entraînées sur un jeu de données. Par la suite, une seconde méthode statistique, appelée méta-classifieur, apprend à combiner les modèles entraînés.

Par ailleurs, (Zhou, 2009) envisage que des méthodes ensemblistes ne fassent partie d'aucune de ces trois grandes catégories.

L'apparition des premières méthodes ensemblistes en apprentissage automatique supervisé remonte aux années 1970s (Sagi et Rokach, 2018). L'état-de-l'art sur les méthodes ensemblistes a déjà été étudié, auparavant (Zhou, 2009 ; Sagi et Rokach, 2018 ; Dietterich, 2000). Nous explorons ici quelques approches ensemblistes récentes liées à l'analyse d'images et de vidéos, et à l'analyse de foule.

Dans le cadre de l'analyse d'images et de vidéos, (Liu et al., 2017) appliquent une méthode ensembliste pour apporter une solution aux classes faiblement pourvues dans un jeu de données de classification des images de véhicules. Pour ce faire, ils appliquent un échantillonnage équilibré et l'augmentation de données. Leur méthode ensembliste consiste en une combinaison des décisions de multiples modèles ResNets (He et al., 2016) (ResNet-50, ResNet-101, et ResNet-152, pré-entraînés sur ImageNet (Deng et al., 2009)), en utilisant le vote majoritaire.

(Pouyanfar et Chen, 2017) proposent EDL "Ensemble Deep Learning" qu'ils utilisent pour la classification de vidéos sur les jeux de données Trecvid (Smeaton, Over et Kraaij, 2006) et Disaster (Pouyanfar et Chen, 2016). EDL est une suite de modèles deep learning extracteurs de caractéristiques sur des images, qui sont des Réseaux de Neurones Convolutifs (CNNs) (LeCun et Bengio, 1995) pré-entrainés sur ImageNet, et chaque extracteur est chapeauté par une machine à vecteurs de support (SVM) (Cortes et Vapnik, 1995) qui sert d'apprenant faible dans un ensemble de modèles. Les caractéristiques extraites sont prises de la dernière couche Fully Convolutional Network (FCN) de chaque modèle. Les architectures utilisées sont : AlexNet (Krizhevsky, Sutskever et Hinton, 2012), CaffeNet (Jia et al., 2014), R-CNN (Region based CNN) (Girshick et al., 2014), GoogleNet (Szegedy et al., 2015), ResNet, la combinaison des décisions se fait via un vote pondéré.

Inspirés par (Liu, Wu et Zhou, 2008), (Chen et al., 2014) utilisent une méthode ensembliste qu'ils ont nommée EnwMi (pour Ensemble Weighted Multi-Instance Learning). Ils commencent par échantillonner plusieurs sous-ensembles de la classe majoritaire, et en combinant à chaque fois un sous-ensemble de la classe majoritaire avec la totalité de la classe minoritaire, ils entraînent, en utilisant AdaBoost (Freund et Schapire, 1995), un modèle. Les modèles entraînés sont combinés pour la décision finale.

Dans le cadre de l'analyse de foule, (Walach et Wolf, 2016) appliquent du gradient boosting et l'échantillonnage sélectif sur une architecture basique de CNNs pour compter le nombre d'objets dans une image. Leur approche est appliquée sur des jeux de données de comptage de cellules microscopiques de bactéries, et sur des jeux de données de comptage des statistiques de foule.

(Wu et al., 2017a) empilent plusieurs modèles (stacking) dont les résultats sont considérés comme de nouvelles caractéristiques qui seront utilisées en entrée pour un

nouveau modèle.

Au vu du peu de données annotées, (Gong, Xiang et Hongeng, 2010) apprennent, dans un contexte semi-supervisé, un ensemble de pose-sensitive DPM (Deformable Part-based Model) mixtures (Felzenszwalb et al., 2009) pour la détection de personnes quelle que soit leur posture dans une scène. Les classes de postures considérées sont : front, rear, left, right. Chaque DPM mixture, sensible à une posture spécifique, est entraîné par un Latent-SVM (Yu et Joachims, 2009) à reconnaître une posture spécifique.

Contrairement aux approches précédemment citées (Liu et al., 2017 ; Liu, Wu et Zhou, 2008), nous ne visons pas à résoudre le problème des données non équilibrées. Nous ne faisons pas de Boosting, comme dans (Walach et Wolf, 2016), car dans le Boosting les entraînements sont répétés plusieurs fois en changeant la pondération des exemples d'apprentissage, or un seul entraînement sur des vidéos requiert déjà un temps de calcul de plusieurs heures. Répéter plusieurs entraînements nécessiterait, dans ces conditions, plusieurs jours. Nous optons pour un compromis entre une forme de Stacking (Wu et al., 2017a), sans méta-classifieur car nous combinons les modèles non pas à l'entraînement mais lors de la prédiction afin de ne pas alourdir la procédure d'apprentissage des modèles, et une forme de Bagging, car nous réalisons une agrégation de modèles sans recourir à l'échantillonnage Bootstrap. Ici, les échantillons sont déjà découpés pour la validation croisée. Ce découpage est stratifié, car chaque échantillon respecte la distribution des classes du jeu de données original. Nous ne faisons pas d'apprentissage semi-supervisé combiné à l'usage de méthodes ensemblistes, comme dans (Gong, Xiang et Hongeng, 2010), car nous n'avons pas de problème de manque d'annotation dans les données que nous utilisons. Tous les vidéo-clips y sont annotés.

(Zhou, Wu et Tang, 2002) soutiennent qu'il n'est pas utile de mettre un très grand nombre de modèles dans un ensemble. Un choix d'un petit nombre de modèles qui produisent déjà de bonnes performances, suffit à produire de meilleures performances lorsqu'ils sont combinés dans un ensemble. De ce fait, nous avons décidé de découper le jeu de données en 5 échantillons, tel qu'il a été déjà réalisé dans la section 2.2.3.1, ce qui permet de doter chaque ensemble de 4 modèles individuels qui extraient des connaissances différentes du jeu de données Crowd-11.

3.2.2 Classification ensembliste

Au vu des approches citées dans la section 3.2.1, et en nous basant sur la définition proposée par (Zhou, 2009), l'apprentissage ensembliste consiste à entraîner ou à évaluer un ensemble de méthodes statistiques semblables, quelles que soient leurs conditions d'entraînement, ou totalement différentes.

Dans un premier temps, nous mettons en place des approches ensemblistes constituées de modèles homogènes disposant des mêmes conditions d'entraînement. Dans un second temps, nous proposons de mettre en place des approches ensemblistes globales qui incluent des modèles hétérogènes ayant différentes architectures et disposant de conditions d'entraînement différentes.

Dans ce chapitre, quand nous parlons d'un ensemble homogène de modèles, ces modèles disposent d'une même architecture et ont été soit entraînés de zéro soit ajustés, par exemple : un ensemble de modèles C3D ajustés. Quand nous parlons d'un ensemble global de modèles hétérogènes, cet ensemble global dispose de plusieurs groupes différents d'ensembles de modèles homogènes, par exemple : un ensemble global constitué d'un ensemble de modèles C3D ajustés et un ensemble de modèles I3D entraînés de zéro.

3.2.2.1 Composition d'ensembles de modèles homogènes

Dans la première partie du chapitre 2, nous avons montré que les modèles issus du réseau TwoStream Inflated 3D (2S-I3D) obtiennent de meilleures performances que les modèles issus du réseau 3D ConvNets (C3D) et du réseau à branche unique I3D (Bendali-Braham et al., 2019). Toutefois les performances des réseaux plafonnent en moyenne à 68% d'accuracy. Ces résultats ont été confirmés par une validation croisée à 5 échantillons.

Dans ce chapitre, nous redécoupons le jeu de données en 5 échantillons, et nous entraînons, pour chaque combinaison possible des échantillons, un modèle issu d'une des architectures suivantes :

— L'architecture 2S-I3D (Carreira et Zisserman, 2017), illustrée dans la figure 2.4,
— L'architecture I3D (Carreira et Zisserman, 2017), illustrée dans la figure 2.3,
— L'architecture C3D (Tran et al., 2015), illustrée dans la figure 2.2,
— L'architecture R3D (Hara, Kataoka et Satoh, 2017).

Dans chaque combinaison, 3 échantillons sont dédiés à l'apprentissage, 1 à la validation, et 1 au test. En sélectionnant à chaque fois, un échantillon de test, nous pouvons produire 4 modèles sur les combinaisons des échantillons restants. Lors de l'évaluation sur un échantillon de test, les décisions des 4 modèles sont combinées pour former un ensemble de modèles.

Avec cette opération, nous nous retrouvons avec 20 combinaisons différentes des ensembles d'apprentissage, de validation, et de test.

Nous rappelons dans les paragraphes qui suivent la constitution de chacune des architectures 2S-I3D, I3D, C3D, R3D.

L'architecture I3D est composée d'une base de 2 couches de réseaux convolutifs 3D (3D ConvNets), chacune appuyée par une normalisation par lots et suivie d'une opération de MaxPooling 3D. Ces 2 couches sont suivies par 9 modules Inception dont la composition interne change légèrement d'un module à un autre. Le dernier module Inception est connecté à un AveragePooling 3D qui est relié à un SoftMax de classification.

L'architecture TwoStream-I3D est composée de deux branches. Chaque branche reprend l'architecture du réseau I3D. Une des deux branches extrait des caractéristiques d'un clip vidéo RVB, et une autre extrait les caractéristiques d'un clip vidéo en flux optique. Les sorties de ces deux branches sont connectées à la fonction de classification Softmax.

L'architecture C3D, proposée par (Tran et al., 2015), est constituée de 5 couches de convolutions 3D, suivies de deux couches de FCN et d'une fonction de classification Softmax.

L'architecture R3D, que proposent (Hara, Kataoka et Satoh, 2017), est constituée de blocs résiduels. Chaque bloc résiduel est formé de deux couches de convolutions 3D. Tout comme, nous l'avons fait dans le chapitre précédent, nous choisissons la version à 34 couches cachées constituée de 16 blocs résiduels, et se finissant par une couche FCN qui précède l'opération de classification softmax.

3.2.2.2 Constitution des ensembles d'apprentissage, de validation, et de test

Afin de créer des échantillons devant participer à la réalisation de la validation croisée, Section 2.2.3.1, nous avons découpé les échantillons en partant des scènes. Pour maintenir la diversité et une quantité similaire de vidéos entre les échantillons

nous avons appliqué les algorithmes 1 et 2. Ces deux algorithmes servent à constituer les échantillons qui seront utilisés pour la création des ensembles d'apprentissage, de validation, et de test, telle qu'elle est illustrée dans la figure 3.1 [1].

La validation croisée sur laquelle nous nous basons a été réalisée dans la section 2.2.3.1, afin de valider les expériences que nous y avons menées. Cette opération a permis de découper le jeu de données en 5 échantillons.

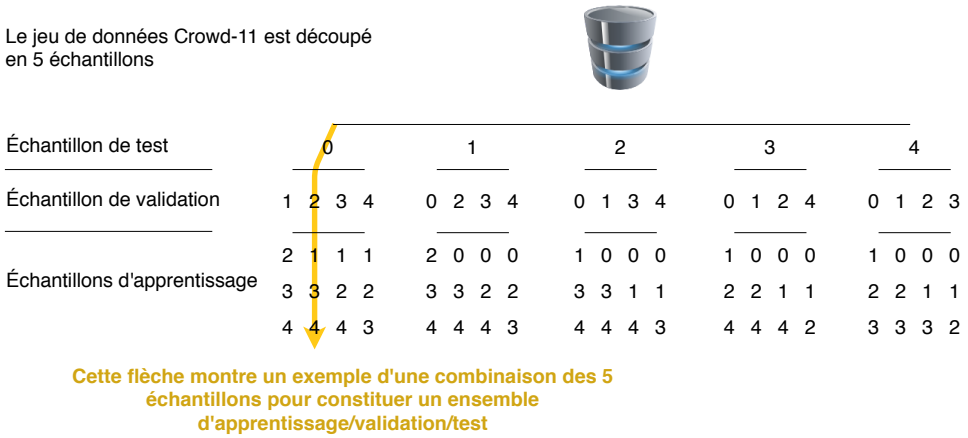


FIGURE 3.1 – Illustration de la procédure de constitution des ensembles d'apprentissage, de validation, et de test à partir de différentes combinaisons des échantillons résultant du découpage du jeu de données

Données : Pré-calculer Sc_freq et Cls_freq
Sc : liste des scènes;
Nb_echs : nombre des échantillons;
Sc_freq : nombre de vidéos par scène;
Cls_freq : nombre de vidéos par classe;
Résultat : Découpage du jeu de données en scènes $Echs\_scenes$.
$Echs\_scenes$: listes de scènes pour tous les échantillons initialisés à des listes vides;
$Echs\_distribs$: distributions des scores des échantillons en fonction des vidéos qu'ils contiennent et de la diversité de ces dernières;
tant que *Sc n'est pas vide* **faire**
    Sélectionner l'échantillon avec le score le plus faible à partir de $Echs\_distribs$;
    Sélectionner la scène qui contient le plus de vidéos Sc_freq;
    Supprimer la scène sélectionnée de Sc_freq et Sc;
    Ajouter la scène sélectionnée à l'échantillon sélectionné dans $Echs\_scenes$;
    Mettre à jour le score de l'échantillon sélectionné dans $Echs\_distribs$;
fin

Algorithme 1 : Découper le jeu de données en échantillons

1. Le code source de cette section est publiquement disponible ici : https://github.com/MounirB/Crowded-scenes-Ensemble-classification

Données : En arguments : $Ech\_distrib$, s, Nb_echs, Cls_freq
s : scène précédemment sélectionnée;
$Vids$: liste de toutes les vidéos;
$Database$: informations sur le jeu de données qui lient des scènes à leurs vidéos;
Résultat : Nouveau score de l'échantillon $Ech\_distrib$ dans $Echs\_distribs$
Sc_vids : jointure entre la scène s et $Database$, et récupération des vidéos de la scène à partir de $Vids$;
$Sc\_classes$: établir la liste des classes présentes dans la scène s à partir de Sc_vids;
pour chaque *classe* c *dans* $Sc\_classes$ **faire**
  $Ech\_distrib_c = Ech\_distrib_c + \frac{Nb\_echs}{Cls\_freq_c}$
fin

Algorithme 2 : Mettre à jour le score d'un échantillon

3.2.2.3 Ensembles globaux de modèles hétérogènes

Nous créons des ensembles globaux de modèles ayant soit différentes architectures, par exemple des ensembles 2S-I3D ajustés couplés avec des ensembles C3D ajustés, soit différentes conditions d'entraînement, par exemple des ensembles I3D ajustés et I3D entraînés de zéro, ou divers ensembles de modèles cumulant les deux différences, par exemple des ensembles C3D entraînés de zéro, des ensembles I3D ajustés, et des ensembles 2S-I3D ajustés. Nous créons des ensembles globaux à partir des ensembles de modèles homogènes comparés dans la section 3.2.3.1. Afin de trouver la combinaison appropriée des ensembles de modèles, nous allons évaluer toutes les combinaisons possibles à partir de ces ensembles de modèles homogènes. L'équation 3.1 calcule le nombre de combinaisons sans répétition pouvant donner corps à des ensembles globaux de modèles hétérogènes.

$$nb\_combinaisons = \sum_{i=2}^{K} C(K, i) \tag{3.1}$$

Où K représente la taille maximale d'une combinaison constituant un ensemble global. $C(K, i)$ représente la fonction de calcul d'une combinaison sans répétition où le nombre de choix est i. i est la longueur du tuple qui représente le nombre d'ensembles de modèles homogènes donnant lieu à une combinaison d'un ensemble global de modèles. Comme nous évaluons déjà les ensembles de modèles homogènes dans la section 3.2.3.1, i commence à partir de 2 qui est considéré comme la taille minimale d'une combinaison.

3.2.3 Expériences

Dans cette section, nous détaillons les différents types d'expériences que nous avons menées :

— Nous avons comparé les performances d'ensembles formés de modèles individuels pré-entraînés face à d'autres ensembles formés de modèles individuels qui n'ont pas bénéficié du pré-entraînement ;
— Nous avons recherché les poids optimaux sur les ensembles d'apprentissage et de validation en faisant soit appel à l'évolution différentielle, soit à la recherche exhaustive, la pondération basée sur l'inverse de l'erreur à la validation, ou le

choix qui se porte sur la classe ayant la plus forte probabilité dans l'ensemble pour chaque clip de l'échantillon de test ;
— Nous avons comparé des ensembles de modèles ayant bénéficié d'un entraînement sur le jeu de données Crowd-11 augmenté et comparé à des modèles n'ayant pas bénéficié d'un entraînement sur le jeu de données augmenté ;

Dans ce travail, les hyperparamètres choisis pour les apprentissages correspondent aux hyperparamètres présentés dans la section 2.2.3. Les clips du jeu de données Crowd-11 durent en moyenne ≈ 5 secondes. Pour les architectures I3D et 2S-I3D, 20 images sont sélectionnées d'un clip vidéo. Ces images se trouvent à intervalles réguliers tout au long du clip. La taille de chaque image est de 224×224 pixels. Pour les architectures R3D et C3D, 16 images sont sélectionnées d'un clip vidéo et la taille de chaque image est de 112×112 pixels. Dans les expériences de comparaisons de modèles ajustés avec les modèles non ajustés, la version flux optique de chaque clip est obtenue via l'algorithme TV-L1 (Zach, Pock et Bischof, 2007). Dans les expériences portant sur la comparaison de modèles ayant bénéficié ou non d'une augmentation des données, nous allons tester l'usage de l'algorithme d'extraction de flux optique Farneback du fait de sa rapidité par rapport à l'algorithme TV-L1. Nous vérifierons également si le recours à l'algorithme Farneback ne réduit pas beaucoup les performances des modèles 2S-I3D.

3.2.3.1 Comparaison d'ensembles de modèles ayant des architectures homogènes

Nous souhaitons vérifier si un ensemble de modèles ajustés est plus performant qu'un ensemble de modèles entraînés de zéro, et quelle condition de pré-entraînement favorise le mieux la création d'ensembles de modèles. Les modèles 2S-I3D et I3D ajustés sur Crowd-11, ont été pré-entraînés sur les jeux de données ImageNet (Deng et al., 2009) et Kinetics (Kay et al., 2017). Les modèles C3D ajustés sur Crowd-11, ont été pré-entraînés sur le jeu de données Sports-1m (Karpathy et al., 2014). D'un autre côté, nous entraînons de zéro sur Crowd-11 les modèles 2S-I3D, I3D, C3D, et R3D.

Dans toutes les situations, nous créons 5 grands contextes d'apprentissage, de validation et de test, où nous fixons un échantillon de test, en amont, et nous varions les échantillons de validation (différents du test), en aval. Dans ces conditions, pour chaque ensemble de modèles, nous réservons trois échantillons, différents du test et à chaque fois différents de la validation, pour l'apprentissage de 4 modèles individuels. Au total, pour les 5 échantillons de test, 20 modèles individuels sont appris. Chaque groupe de 4 modèles individuels, qu'ils soient ajustés ou entraînés de zéro, forme un ensemble de modèles.

Les résultats des prédictions de ces modèles sont visibles dans les tables 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7. À partir des tables 3.1, 3.3, 3.5 nous voyons globalement que la constitution de modèles ensemblistes à partir des modèles individuels ajustés améliore très souvent les performances globales lors de la classification. Ceci n'est pas toujours le cas pour les modèles ensemblistes regroupant des modèles individuels entraînés de zéro, tel que nous pouvons le constater dans les tables 3.2, 3.4. Ici, les écarts-types des performances sont plus prononcés. Dans cette situation l'amélioration des performances induite par la construction d'un ensemble est mince, et parfois, un des modèles individuels surpasse la performance de l'ensemble des modèles. Dans tous les cas, nous voyons que le recours à un ensemble de modèles est meilleur que la moyenne des performances des modèles individuels. Ces résultats montrent que

les ensembles de modèles ayant des réseaux qui obtiennent déjà de bonnes performances telles que les réseaux ajustés 2S-I3D, I3D, et C3D sont ceux qui obtiennent les meilleures performances. En l'occurrence, les ensembles de modèles ajustés 2S-I3D obtiennent les meilleurs scores sur Crowd-11.

Quelques ensembles de modèles homogènes sont illustrés dans la figure 3.2.

Échantillon de test	**0**	**1**	**2**	**3**	**4**	μ	σ
Éch de val : accuracy par modèle individuel associé	1 : 67.86	0 : 66.55	0 : 66.04	0 : 63.09	0 : 69.73		
	2 : 69.08	2 : 66.55	1 : 67.28	1 : 66.26	1 : 70.60		
	3 : 67.33	3 : 66.29	3 : 68.52	2 : 63.52	2 : 69.30		
	4 : 69.86	4 : 65.44	4 : 66.66	4 : 62.15	3 : 67.39		
Écart-type des accuracies par échantillon de test	0.99	0.45	0.91	1.52	1.17		
Moyenne des accuracies par échantillon de test	68.53	66.21	67.13	63.76	69.26	66.98	**1.92**
Accuracy par ensemble	**70.48**	**67.57**	**68.61**	**66.43**	**72.00**	**69.02**	1.99

TABLE 3.1 – Comparaison entre les résultats obtenus par les ensembles de modèles 2S-I3D ajustés et leurs modèles individuels

Échantillon de test	**0**	**1**	**2**	**3**	**4**	μ	σ
Éch de val : accuracy par modèle individuel associé	**1 : 54.93**	0 : 54.55	0 : 55.52	0 : 46.91	0 : 58.60		
	2 : 48.55	2 : 53.70	1 : 53.13	1 : 52.05	1 : 53.56		
	3 : 47.16	3 : 55.31	3 : 59.94	2 : 54.36	2 : 57.13		
	4 : 47.07	4 : 50.55	4 : 56.76	4 : 51.36	3 : 56.78		
Écart-type des accuracies par échantillon de test	3.23	1.81	2.45	2.69	1.83		
Moyenne des accuracies par échantillon de test	49.43	53.53	56.34	51.17	56.52	53.40	**2.79**
Accuracy par ensemble	54.41	**56.42**	**60.83**	**54.45**	**61.30**	**57.48**	3.01

TABLE 3.2 – Comparaison entre les résultats obtenus par les ensembles de modèles 2S-I3D, n'ayant pas bénéficié du pré-entraînement, et leurs modèles individuels

Échantillon de test	**0**	**1**	**2**	**3**	**4**	μ	σ
Éch de val : accuracy par modèle individuel associé	1 : 57.99	0 : 57.87	0 : 58.26	0 : 55.30	0 : 58.43		
	2 : 58.16	2 : 57.19	1 : 59.68	1 : 56.33	1 : 60.34		
	3 : 60.26	**3 : 60.68**	3 : 57.29	2 : 57.87	2 : 60.26		
	4 : 58.95	4 : 59.23	4 : 56.76	4 : 55.39	3 : 59.73		
Écart-type des accuracies par échantillon de test	0.89	1.33	1.10	1.03	0.76		
Moyenne des accuracies par échantillon de test	58.84	58.74	58.00	56.22	59.65	58.30	**1.16**
Accuracy par ensemble	**61.13**	60.59	**61.00**	**58.13**	**61.56**	**60.48**	1.21

TABLE 3.3 – Comparaison des performances, par échantillon de test, des ensembles de modèles C3D ajustés

3.2.3.2 Recherche d'une pondération optimale à partir de l'ensemble de modèles 2S-I3D ajustés

À la suite des expérimentations menées dans la section 3.2.3.1, nous avons trouvé que le recours à la somme des probabilités ou à une pondération simple $\frac{1}{n}$, n étant le nombre de modèles individuels constituant un ensemble, améliore globalement les résultats des modèles. Forts de ces résultats, nous avons décidé de rechercher une pondération qui met davantage en valeur les modèles les plus performants. Nous allons

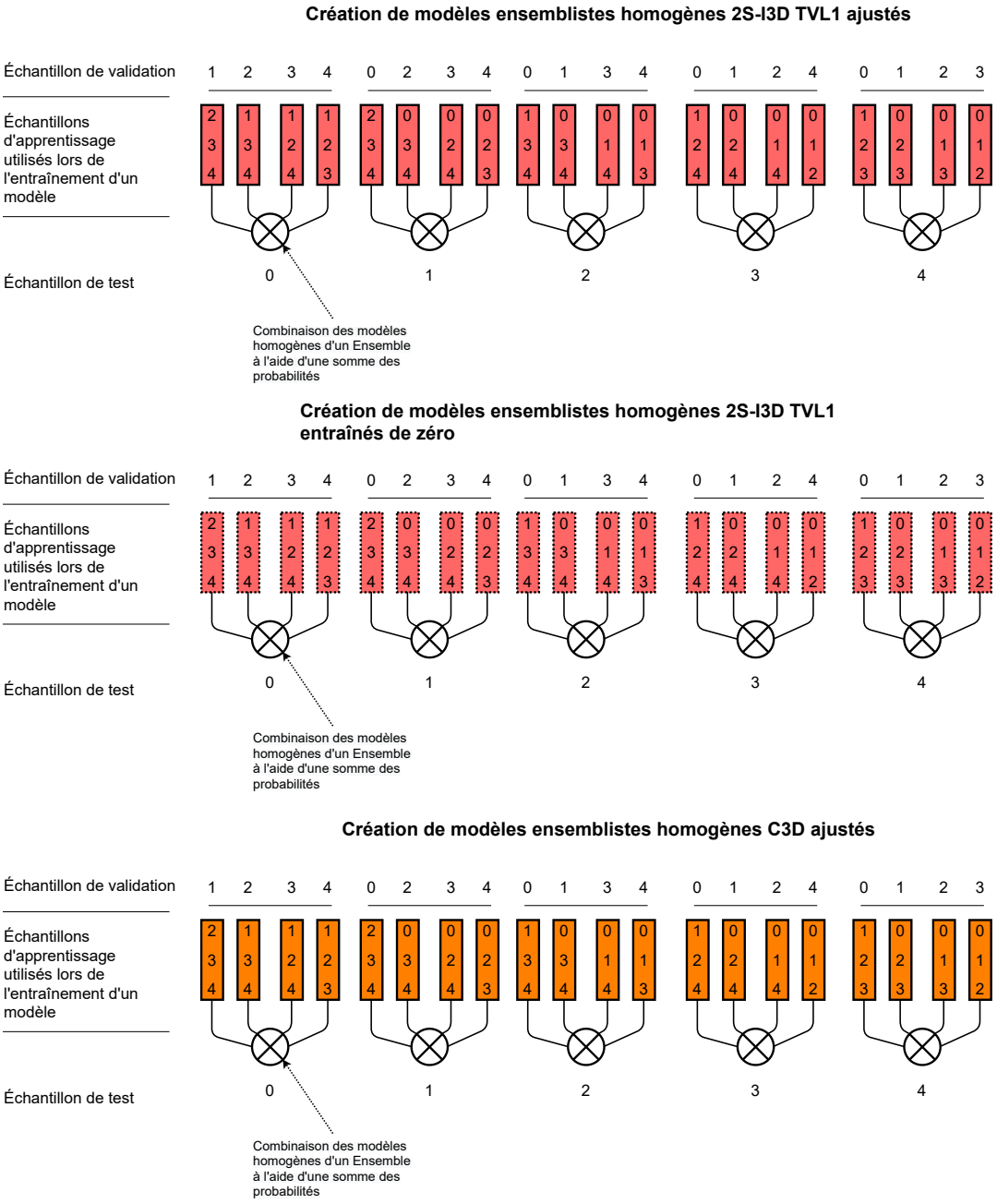


FIGURE 3.2 – Quelques exemples de modèles homogènes. Les modèles ajustés sont représentés par des rectangles en trait et les modèles entrainés de zéro sont représentés par des rectangles en pointillé

Échantillon de test	**0**	**1**	**2**	**3**	**4**	μ	σ
Éch de val : accuracy par modèle individuel associé	**1 : 31.79**	0 : 33.78	0 : 30.76	0 : 30.05	0 : 36.52		
	2 : 28.73	2 : 31.65	1 : 30.41	1 : 28.59	1 : 35.21		
	3 : 31.35	3 : 32.25	**3 : 33.86**	2 : 32.19	2 : 37.91		
	4 : 30.91	**4 : 34.38**	4 : 31.91	**4 : 33.47**	**3 : 39.39**		
Écart-type des accuracies par échantillon de test	1.17	1.10	1.34	1.88	1.55		
Moyenne des accuracies par échantillon de test	30.69	33.02	31.74	31.07	37.26	32.76	**2.38**
Accuracy par ensemble	31.26	32.76	32.27	31.76	38.08	**33.23**	2.47

TABLE 3.4 – Comparaison des performances, par échantillon de test, des ensembles de modèles C3D entraînés de zéro

Échantillon de test	**0**	**1**	**2**	**3**	**4**	μ	σ
Éch de val : accuracy par modèle individuel associé	1 : 60.87	0 : 57.95	0 : 61.27	0 : 55.05	0 : 57.56		
	2 : 61.92	2 : 57.53	1 : 60.91	1 : 57.44	1 : 59.47		
	3 : 60.87	3 : 59.48	3 : 61.53	2 : 57.10	2 : 57.65		
	4 : 63.93	4 : 58.55	4 : 59.41	4 : 54.53	3 : 60.43		
Écart-type des accuracies par échantillon de test	1.24	0.73	0.82	1.26	1.22		
Moyenne des accuracies par échantillon de test	61.89	58.38	60.78	56.03	58.78	59.17	**2.03**
Accuracy par ensemble	**64.10**	**60.25**	**62.24**	**57.70**	**60.95**	**61.05**	2.12

TABLE 3.5 – Comparaison des performances, par échantillon de test, des ensembles de modèles I3D ajustés

Échantillon de test	**0**	**1**	**2**	**3**	**4**	μ	σ
Éch de val : accuracy par modèle individuel associé	1 : 49.78	0 : 54.04	0 : 53.93	0 : 49.14	0 : 56.00		
	2 : 53.44	2 : 52.34	1 : 58.17	1 : 52.56	1 : 55.91		
	3 : 52.83	3 : 52.68	3 : 54.28	2 : 51.96	2 : 55.56		
	4 : 53.10	4 : 54.97	4 : 53.05	4 : 50.08	3 : 52.00		
Écart-type des accuracies par échantillon de test	1.46	1.06	1.96	1.38	1.66		
Moyenne des accuracies par échantillon de test	52.29	53.51	54.86	50.94	54.86	53.29	**1.51**
Accuracy par ensemble	**54.93**	**55.91**	**58.53**	**53.85**	**58.86**	**56.42**	1.97

TABLE 3.6 – Comparaison des performances, par échantillon de test, des ensembles de modèles I3D entraînés de zéro

Échantillon de test	**0**	**1**	**2**	**3**	**4**	μ	σ
Éch de val : accuracy par modèle individuel associé	1 : 44.45	0 : 48.93	0 : 49.60	0 : 44.60	0 : 49.73		
	2 : 44.54	2 : 48.34	1 : 45.97	1 : 45.89	1 : 46.52		
	3 : 47.86	3 : 49.44	3 : 46.59	2 : 48.11	2 : 45.21		
	4 : 42.18	4 : 50.89	4 : 45.80	4 : 45.89	3 : 43.91		
Écart-type des accuracies par échantillon de test	2.02	0.94	1.53	1.26	2.16		
Moyenne des accuracies par échantillon de test	44.75	49.40	46.99	46.12	46.34	46.72	**1.52**
Accuracy par ensemble	47.42	**52.00**	**50.13**	**48.63**	**50.43**	**49.72**	1.57

TABLE 3.7 – Comparaison des performances, par échantillon de test, des ensembles de modèles R3D, avec 34 couches cachées, entraînés de zéro

limiter nos recherches d'une pondération optimale aux ensembles de modèles ajustés 2S-I3D du fait des bons résultats qu'ils obtiennent comparativement aux autres architectures de modèles.

Échantillon de test	**0**	**1**	**2**	**3**	**4**	μ	σ
Somme des probabilités	70.48	67.57	68.61	**66.43**	**72.00**	**69.021**	1.99
Probabilité maximale	70.48	**67.74**	68.61	66.35	71.56	68.95	**1.87**
Inverse de l'erreur à la validation	70.74	67.65	68.61	**66.43**	71.65	69.020	1.92
Recherche des poids par Recherche exhaustive	**70.91**	66.89	**68.70**	66.01	71.65	68.83	2.19
Recherche des poids par Évolution différentielle	70.82	66.72	**68.70**	66.26	71.56	68.81	2.12

TABLE 3.8 – Performances des ensembles de modèles 2S-I3D ajustés en fonction des poids qui sont accordés à leurs modèles individuels

Pondération max Dans cette approche, nous affectons au clip de test, l'étiquette de la classe ayant obtenu la plus grande probabilité dans tout l'ensemble de modèles.

Inverse de l'erreur à la validation Dans cette approche, les poids des modèles individuels d'un ensemble sont estimés en calculant l'inverse de l'erreur de chaque modèle obtenue lors de la validation. Cette idée s'inscrit dans le même sillage qu'une pondération pratiquée par (WICHARD, 2016). Ici, les poids sont normalisés en utilisant l'opération suivante :

$$W_k^{norm} = \frac{W_k}{\sum_k^n W_k} \tag{3.2}$$

n représente le nombre de modèles d'un ensemble.

Évolution différentielle Ici, les poids des modèles individuels sont recherchés en réalisant des évaluations sur les échantillons ayant été utilisés pour l'apprentissage et la validation des modèles. L'algorithme de recherche de poids optimaux est la méta-heuristique évolution différentielle (STORN et PRICE, 1997).

Recherche exhaustive Nous recherchons de manière exhaustive les poids associés aux modèles individuels d'un ensemble, en procédant à des évaluations sur les échantillons d'apprentissage et de validation. L'exploration de ces poids se fait dans l'intervalle $[0.0, 1.0]$ avec un pas de 0.1. À chaque itération de la fonction de recherche, les poids sélectionnés dans l'intervalle sont normalisés en fonction du nombre de modèles.

Le recours à la recherche exhaustive nous a fourni des poids qui sont approximativement similaires à ceux obtenus lors de l'évolution différentielle.

Discussion des résultats des méthodes de pondération Les performances obtenues à la suite des cinq méthodes de pondération peuvent être consultées dans la table 3.8. De prime abord, nous observons que peu importe la méthode de pondération choisie, les ensembles de modèles ont des performances proches en moyenne. Nous voyons que le recours à la somme des probabilités ou à l'attribution d'un même poids à chaque modèle individuel suffit en moyenne à mettre en valeur chaque modèle individuel dans un ensemble de modèles. Par ailleurs, vu que la recherche de poids exhaustive ou l'évolution différentielle se fait sur les échantillons d'apprentissage et de validation, les poids trouvés permettront à l'ensemble de sur-apprendre les échantillons d'apprentissage et de validation, mais cela ne lui permettra pas de bien généraliser sur l'ensemble de test.

3.2.3.3 Augmentation des données

Nous avons mené des expériences en augmentant les données vidéo afin de tester l'apport de l'augmentation de données sur la classification des vidéos de mouvements de foule. L'augmentation de données n'a été appliquée que sur les données de l'ensemble d'apprentissage du jeu de données Crowd-11. Le type d'opérations d'augmentation de données effectuées sont : découpage aléatoire (Random Crop), effet poivre et sel (Salt and Pepper), effet miroir horizontal (Horizontal Flip). Dans nos expériences, nous comparons deux stratégies d'augmentation de données :

— Une augmentation de données fixe pré-calculée avant le lancement de la phase d'entraînement. Les méthodes d'augmentation choisies aléatoirement de manière exclusive ou combinée avec d'autres méthodes d'augmentation. Suite à l'augmentation de données, la taille de l'ensemble d'apprentissage a été augmenté 3 fois. À chaque époque de la phase d'entraînement, le modèle explore les données augmentées ainsi que les données non augmentées.

— Une augmentation de données à-la-volée qui se renouvelle à chaque fois lors de la phase d'entraînement. Nous accordons à cette augmentation 75% de probabilité de se produire à chaque itération pendant les époques d'entraînement. Les époques sont répétées 4 fois afin de permettre à l'augmentation de données à-la-volée de correspondre à l'augmentation de données pré-calculée en termes de quantité et de variété des données augmentées et non augmentées.

Préalablement à l'augmentation des données, nous avons décidé de changer l'algorithme d'extraction du flux optique TVL1 par l'algorithme de Farneback. Ce changement est motivé par la vitesse d'extraction de l'algorithme de Farneback qui est théoriquement connu pour être moins précis que l'algorithme TVL1.

Échantillon de test	**0**	**1**	**2**	**3**	**4**	μ	σ
Éch de val : accuracy par modèle individuel associé	1 : 68.03	0 : 65.36	0 : 69.40	0 : 61.98	0 : 68.86		
	2 : 66.46	2 : 66.72	1 : 68.70	1 : 62.75	1 : 68.95		
	3 : 66.81	**3 : 68.42**	3 : 68.16	2 : 63.69	2 : 68.34		
	4 : 69.17	4 : 64.85	4 : 68.34	4 : 60.53	3 : 68.00		
Écart-type des accuracies par échantillon de test	1.06	1.38	0.47	1.15	0.39		
Moyenne des accuracies par échantillon de test	67.62	66.34	68.65	62.24	68.54	66.68	**2.36**
Accuracy par ensemble	**70.56**	66.55	**69.93**	**64.21**	**70.78**	**68.41**	2.59

TABLE 3.9 – Performances des ensembles de modèles 2S-I3D ajustés se basant sur les flots optiques extraits à l'aide l'algorithme de Farneback et n'ayant pas bénéficié d'une augmentation de données

Échantillon de test	**0**	**1**	**2**	**3**	**4**	μ	σ
Éch de val : accuracy par modèle individuel associé	1 : 66.20	0 : 68.51	0 : 69.40	0 : 64.38	0 : 69.21		
	2 : 69.86	2 : 67.48	1 : 67.99	1 : 62.75	1 : 67.21		
	3 : 68.82	3 : 66.72	3 : 69.40	2 : 59.76	2 : 69.04		
	4 : 69.60	4 : 65.19	4 : 70.29	4 : 61.47	3 : 69.13		
Écart-type des accuracies par échantillon de test	1.45	1.21	0.82	1.69	0.83		
Moyenne des accuracies par échantillon de test	68.62	66.97	69.27	62.09	68.65	67.12	2.62
Accuracy par ensemble	**71.00**	**69.10**	**71.88**	**65.58**	**71.47**	**69.81**	**2.31**

TABLE 3.10 – Performances des ensembles de modèles 2S-I3D ajustés se basant sur les flots optiques extraits à l'aide l'algorithme de Farneback et bénéficiant de l'augmentation de données pré-calculée

Échantillon de test	**0**	**1**	**2**	**3**	**4**	μ	σ
Éch de val : accuracy par modèle individuel associé	1 : 67.59	0 : 64.42	0 : 64.98	0 : 61.64	0 : 65.73		
	2 : 64.80	2 : 65.87	1 : 68.08	1 : 64.04	**1 : 69.65**		
	3 : 64.10	3 : 66.72	3 : 66.48	2 : 60.78	2 : 66.60		
	4 : 64.27	4 : 64.42	4 : 67.19	4 : 61.21	3 : 67.47		
Écart-type des accuracies par échantillon de test	1.41	0.98	1.13	1.26	1.45		
Moyenne des accuracies par échantillon de test	65.19	65.36	66.68	61.92	67.36	65.30	**1.87**
Accuracy par ensemble	**68.47**	**67.23**	**69.23**	**64.21**	69.30	**67.69**	1.89

TABLE 3.11 – Performances des ensembles de modèles 2S-I3D ajustés se basant sur les flots optiques extraits à l'aide l'algorithme de Farneback et bénéficiant de l'augmentation de données à la volée

Échantillon de test	**0**	**1**	**2**	**3**	**4**	μ	σ
Accuracy par ensemble FarneBack Non Augmented	70.56	66.55	69.93	64.21	70.78	68.41	2.59
Accuracy par ensemble FarneBack Augmented Precomputed	**71.00**	**69.10**	**71.88**	**65.58**	**71.47**	**69.81**	2.31
Accuracy par ensemble FarneBack Augmented On The Fly	68.47	67.23	69.23	64.21	69.30	67.69	**1.89**

TABLE 3.12 – Comparaison des performances des ensembles par échantillon de test avec ou sans augmentation de données

Discussion des résultats de l'augmentation des données À la suite de nos expériences, dont les résultats sont illustrés dans les tables 3.9,3.10,3.11, nous constatons que le recours à l'algorithme d'extraction Farneback réduit légèrement les performances des modèles 2S-I3D. La moyenne des scores passe de 69.02% à 68.41%. Cette réduction est toutefois minime et peut être négligée pour la suite des expériences illustrées dans les deux tables 3.10,3.11. L'augmentation des données calculée à la volée n'améliore pas les résultats. Pire : elle réduit les performances des modèles 2S-I3D. L'augmentation des données pré-calculée améliore considérablement les résultats des modèles 2S-I3D ajustés alimentés dans leurs secondes branches par des clips en flux optique extraits à l'aide de l'algorithme Farneback. Ces ensembles progressent même, en moyenne, d'1 point d'accuracy passant de 68.41% à 69.81% battant même les modèles 2S-I3D ajustés dont les secondes branches ont été alimentées par les clips en flux optique extraits à l'aide de l'algorithme TVL1. Ces bonnes performances sont contrebalancées par le temps d'entraînement requis par les modèles bénéficiant de l'augmentation des données pré-calculée. Ce temps d'entraînement est approximativement 5 fois plus important que le temps d'entraînement des modèles dont les secondes branches sont alimentées par les clips en flux optique extraits à l'aide de TVL1.

3.2.3.4 Recherche d'un ensemble global de modèles avec des architectures hétérogènes

Nous tirons ces ensembles globaux en combinant les modèles d'ensembles homogènes illustrés dans la table 3.14. Ces 8 ensembles peuvent donner lieu à 247 combinaisons formant des ensembles globaux de modèles hétérogènes. Le classement ascendant selon le score d'accuracy de chaque combinaison est affiché, en Annexe, dans les tables C.1,C.2,C.3,C.4.

La combinaison qui obtient les meilleurs résultats associe les ensembles 2S-I3D ajustés ayant bénéficié d'un entrainement sur les données augmentées pré-calculées

Combinaison	Accuracy
I3D_SCRATCH	56.42
C3D_PRETRAINED	60.48
C3D_PRETRAINED, I3D_SCRATCH	63.09
TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, I3D_SCRATCH	68.80
TWOSTREAM_I3D_PRETRAINED_OF_TVL1	69.02
TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED	69.24
TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, C3D_PRETRAINED, I3D_SCRATCH	69.24
TWOSTREAM_I3D_PRETRAINED_OF_TVL1, I3D_SCRATCH	69.39
TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED	69.81
TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, I3D_SCRATCH	70.07
TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, C3D_PRETRAINED	70.47
TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1	71.10
TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, I3D_SCRATCH	71.25
TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED	71.30
TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, I3D_SCRATCH	**71.32**

TABLE 3.13 – Classement par ordre ascendant des ensembles de modèles constituant le meilleur ensemble de modèles hétérogènes (dernière ligne de la table)

de Crowd-11, avec les ensembles 2S-I3D ajustés n'ayant pas bénéficié d'un entraînement sur des données augmentées, les ensembles C3D ajustés, et les ensembles I3D entrainés de zéro. Les résultats de cette combinaison sont illustrés dans la table 3.15. Nous constatons que cette combinaison améliore les performances globales de 1.5% en termes d'accuracy. Cet ensemble global est illustré dans la figure 3.3.

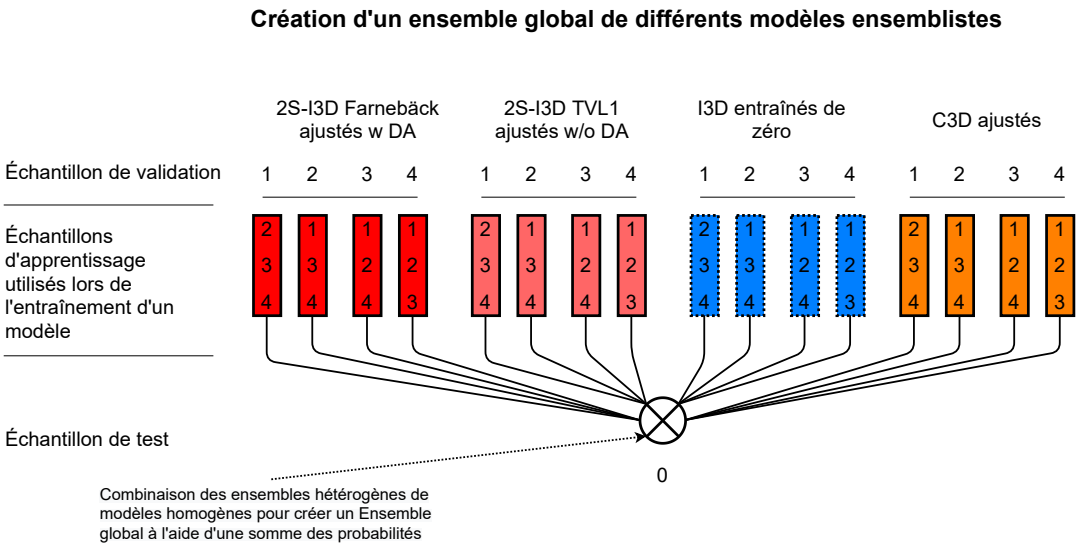


FIGURE 3.3 – Illustration du meilleur ensemble global regroupant 4 différents modèles ensemblistes. Une couleur symbolise une architecture, et le type du cadre fait référence aux conditions d'entraînement : en trait pour les modèles ajustés, et en pointillé pour les modèles entraînés de zéro

Nous allons analyser dans cette discussion la combinaison globale qui réalise le meilleur score d'accuracy et les scores obtenus par ses modèles. À travers la table 3.13, nous pouvons constater les briques de connaissances apportées par chaque ensemble de modèles. Même si l'association de ces 4 modèles, illustrée dans la dernière ligne de la table, aboutit à un score de 71.32%, nous observons que l'association de C3D et I3D pénalise le score de chaque réseau 2S-I3D pris isolément ; le score de 2S-I3D non augmenté baisse de 0.2% et celui du 2S-I3D augmenté baisse de 0.6%. Nous supposons que l'association des probabilités de C3D et I3D doit être contrebalancée par l'association des probabilités des deux réseaux proches 2S-I3D pour que l'association de tous les ensembles permette à chaque modèle d'apporter ses connaissances sans pénaliser les décisions correctes.

Dans la figure 3.4, nous représentons les diagrammes à barres associés à chaque ensemble de modèles constituant l'ensemble global le plus performant. Chaque diagramme à barres représente les performances des modèles individuels constituant chaque ensemble de modèles homogènes.

Chaque ensemble de test est représenté par une couleur différente. À travers ces diagrammes, nous répondons à la question suivante : pour chaque ensemble de modèles homogènes, quel est le nombre de modèles individuels qui réussit à détecter un clip de l'échantillon de test ? À partir de cette figure, nous voyons qu'il existe une limite pour chaque ensemble de modèles. Chaque échantillon de test est constitué en moyenne de 1150 vidéo-clips. Pour chacun de ces échantillons, approximativement 200 à 300 clips ne sont correctement classifiés par aucun modèle d'un ensemble. Ce nombre monte à presque 400 clips mal classés pour les modèles I3D entraînés de zéro.

À partir des matrices de confusion des ensembles de modèles hétérogènes, Figure 3.5, nous constatons que les classes problématiques persistent ; la performance de l'ensemble global des modèles hétérogènes pour ces classes reste en dessous de la moyenne. Ces classes de mouvements étiquetés par les nombres 3, 5, et 6, sont respectivement les classes de mouvements Turbulent Flow, Converging et Diverging Flows. Toutefois, en comparant ces résultats classe par classe affichés dans les matrices de confusion de la figure 3.5 avec les performances obtenues par le réseau 2S-I3D ajusté, illustrées dans la figure 2.5, nous remarquons que les performances de l'ensemble global de modèles hétérogènes s'améliorent légèrement pour les classes Turbulent Flow et Diverging Flow, mais presque pas du tout pour la classe Converging Flows. Comme nous avons sélectionné le modèle ensembliste hétérogène qui réalise le meilleur score d'accuracy, celui-ci doit disposer de modèles qui avantagent les classes problématiques, en évitant de trop pénaliser les classes habituellement non problématiques.

À travers l'analyse des matrices de différence, Figure 3.6, qui résultent de la soustraction entre la matrice de confusion de l'ensemble de modèles hétérogènes et les matrices de confusion de chaque ensemble homogène le constituant, nous constatons que les décisions de l'ensemble de modèles hétérogènes sont très différentes de celles des modèles I3D entraînés de zéro et C3D ajustés, mais elles sont proches des modèles 2S-I3D. Compte tenu des informations apportées par la table 3.13, ceci montre que le modèle global bénéficie davantage de l'expérience des modèles 2S-I3D mais prend en compte l'expérience des modèles I3D et C3D pour la classification de certains clips où les probabilitiés des modèles 2S-I3D sont faibles.

La discussion se conclut par deux résultats principaux :

1. En comparant les performances lors de la classification de modèles individuels et de modèles ensemblistes homogènes, nous trouvons que la classification ensembliste obtient de meilleurs résultats.
2. Toutefois, la combinaison de plusieurs modèles hétérogènes, en un ensemble global, améliore les résultats lors de la classification sous la condition que ces modèles soient complémentaires.

3.3 Conclusion

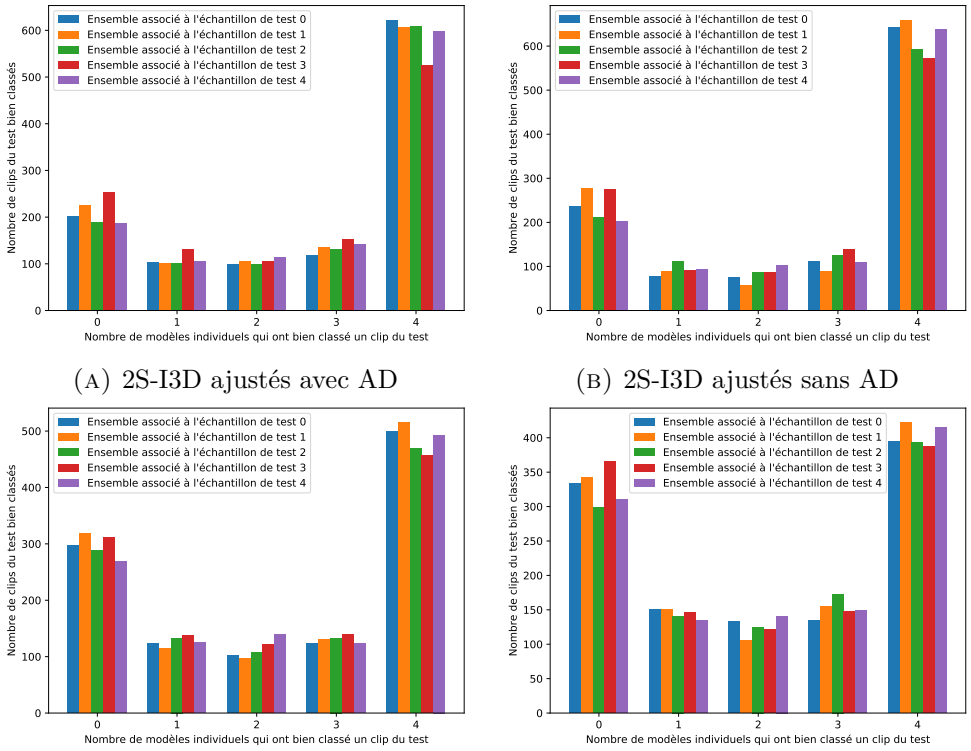
Nous avons constaté au cours de nos expériences que les modèles ensemblistes améliorent en moyenne les résultats de leurs modèles individuels qui ont été entraînés sur des échantillons d'apprentissage différents. Par exemple, pour les modèles 2S-I3D ajustés, l'accuracy moyenne passe de 66.98% à 69.02%. Dans les expériences que nous avons menées jusqu'ici, la recherche d'une pondération optimale des modèles individuels 2S-I3D ajustés en se basant sur les échantillons d'apprentissage et de

Échantillon de test	**0**	**1**	**2**	**3**	**4**	μ	σ
C3D scratch	31.26	32.76	32.27	31.76	38.08	33.23	2.47
C3D pretrained	61.13	60.59	61.00	58.13	61.56	60.48	**1.21**
I3D scratch	54.93	55.91	58.53	53.85	58.86	56.42	1.97
I3D pretrained	64.10	60.25	62.24	57.70	60.95	61.05	2.12
R3D (w 34 layers) scratch	47.42	52.00	50.13	48.63	50.43	49.72	1.57
2S I3D scratch (TVL1)	54.41	56.42	60.83	54.45	61.30	57.48	3.01
2S I3D pretrained (TVL1) w/o DA	70.48	67.57	68.61	**66.43**	**72.00**	69.02	1.99
2S I3D pretrained (Farneback) w DA	**71.00**	**69.10**	**71.88**	65.58	71.47	**69.81**	2.31

TABLE 3.14 – Comparaison entre les ensembles de modèles ayant des architectures homogènes. Quelques explications : **w/o DA** sans augmentation de données ; **w DA** avec augmentation de données.

Échantillon de test	**0**	**1**	**2**	**3**	**4**	μ	σ
(1) C3D pretrained	61.13	60.59	61.00	58.13	61.56	60.48	**1.21**
(2) I3D scratch	54.93	55.91	58.53	53.85	58.86	56.42	1.97
(3) 2S I3D pretrained (TVL1) w/o DA	70.48	67.57	68.61	**66.43**	72.00	69.02	1.99
(4) 2S I3D pretrained (Farneback) w DA	71.00	69.10	71.88	65.58	71.47	69.81	2.31
Ensemble global (1) + (2) + (3) + (4)	**72.05**	**70.04**	**73.20**	66.35	**74.95**	**71.32**	2.95

TABLE 3.15 – Comparaison entre la meilleure combinaison donnant lieu à un ensemble global et les modèles ensemblistes le constituant



(A) 2S-I3D ajustés avec AD

(B) 2S-I3D ajustés sans AD

(C) C3D ajustés

(D) I3D entraînés de zéro

FIGURE 3.4 – Pour chaque ensemble de modèles, représenté par une couleur différente, le nombre maximal (en abscisse) de modèles ayant pu reconnaître la vraie classe d'un clip dans l'échantillon de test associé à cet ensemble

validation ne met pas assez en valeur ces modèles dans un ensemble de modèles. Affecter au clip de test, l'étiquette de la classe ayant obtenu la plus grande probabilité dans tout l'ensemble de modèles, ne donne pas non plus, plus raison à l'ensemble que

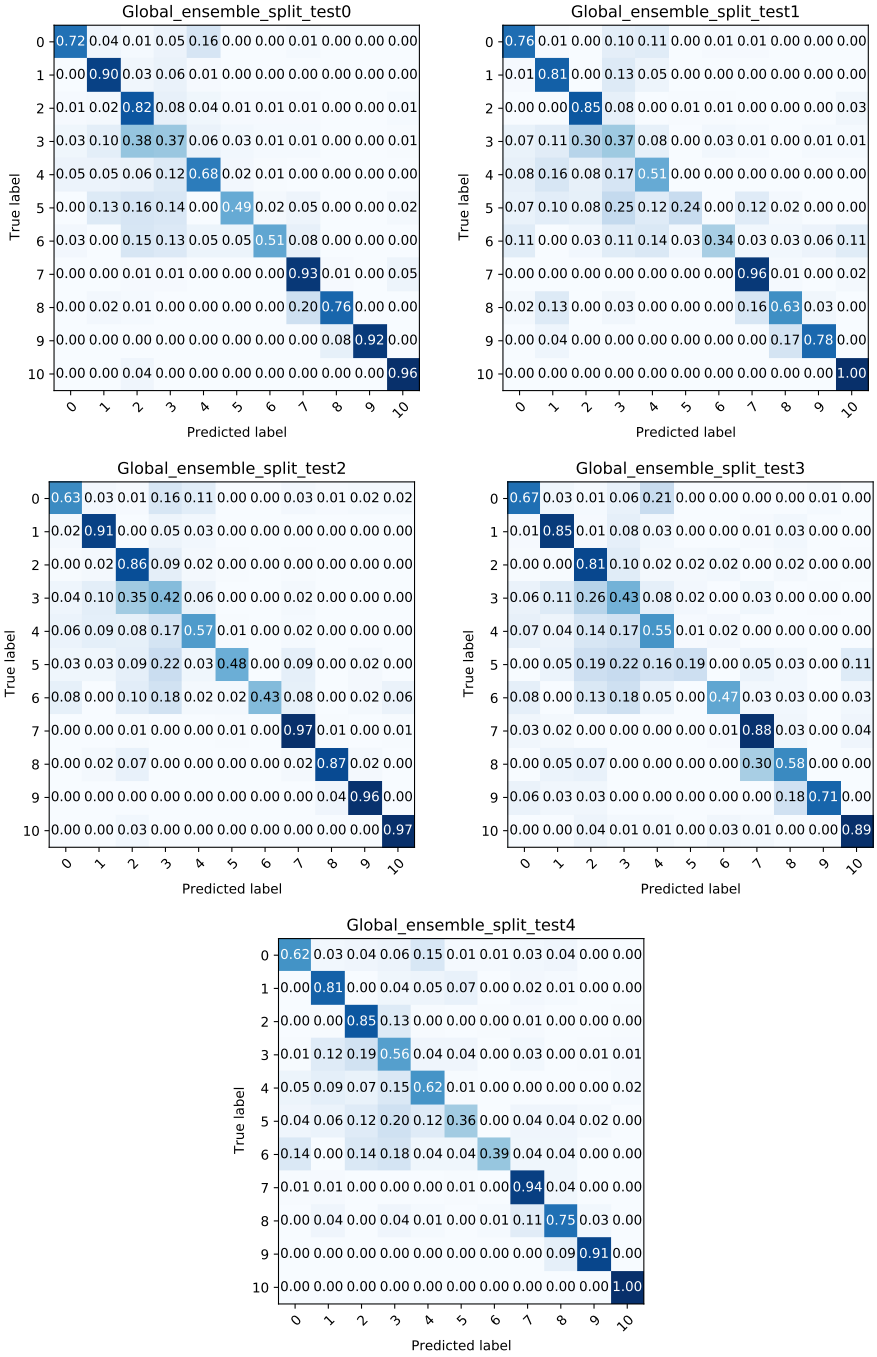


FIGURE 3.5 – Matrices de confusion de chaque ensemble global de modèles contenant les ensembles de modèles homogènes 2S-I3D ajustés avec augmentation des données, les modèles 2S-I3D ajustés sans augmentation des données, les modèles C3D ajustés, et les modèles I3D entraînés de zéro

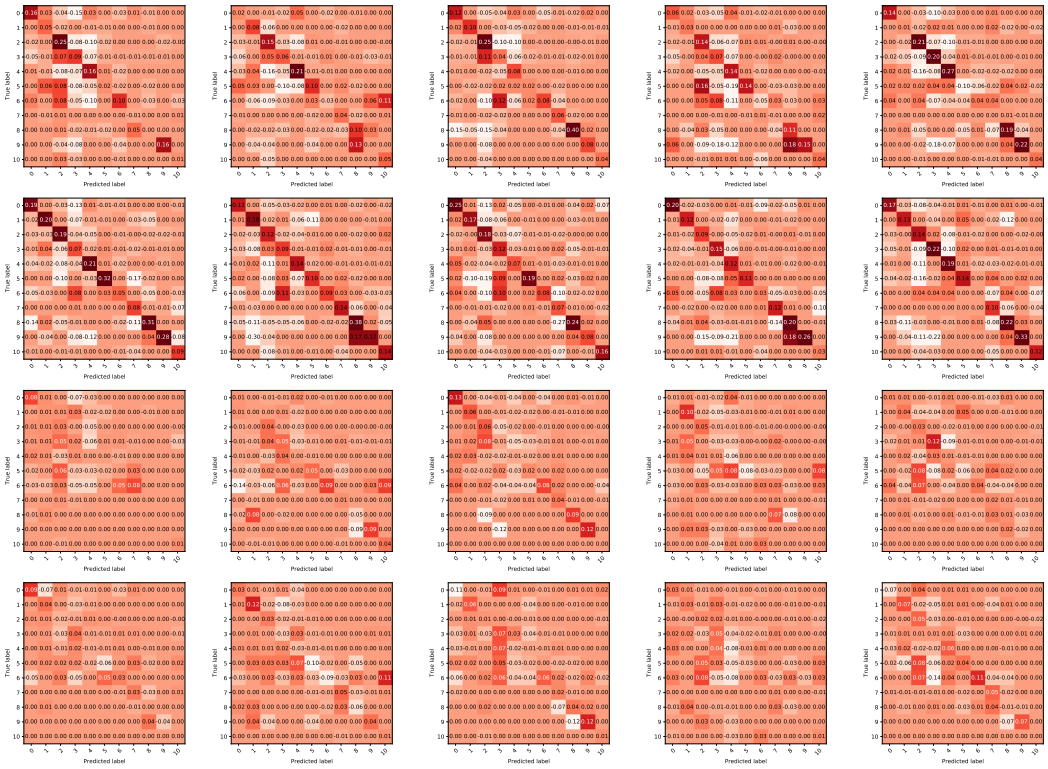


FIGURE 3.6 – Matrices de différence entre chaque ensemble global et les ensembles de modèles homogènes qui le constituent. Chaque rangée contient les matrices de différence associées à un ensemble de modèles homogènes selon cet ordre : **(1)** C3D ajustés, **(2)** I3D entraînés de zéro, **(3)** 2S-I3D, avec le flux optique TVL1, ajustés, sans augmentation de données, **(4)** 2S-I3D, avec le flux optique Farneback, ajustés, avec augmentation de données.

lorsque tous les modèles individuels, aux poids similaires, participent à la décision via la somme.

Nous avons exploré plusieurs façons de composer des ensembles de modèles. Tout d'abord, nous avons créé des ensembles de modèles homogènes, et les avons comparés. Ensuite, nous avons cherché des méthodes de pondération qui favorisent le mieux les ensembles homogènes. Nous avons trouvé qu'une somme simple suffit à constituer des ensembles homogènes. Par la suite, nous avons augmenté les données et comparé deux approches d'augmentation des données. Nous avons vu que l'augmentation des données pré-calculée était la meilleure manière de permettre à des modèles de mieux apprendre. La stratégie d'augmentation de données précalculée a aidé les ensembles 2S-I3D ajustés à améliorer leur accuracy en la faisant passer de 69.02% à 69.81%.

Enfin, nous avons recherché toutes les combinaisons possibles des modèles ensemblistes homogènes donnant lieu à un ensemble global de modèles, et nous avons trouvé la meilleure combinaison possible constituée par 4 ensembles de modèles ayant différentes architectures et différentes pré-conditions d'entraînement. Le meilleur ensemble global d'ensembles hétérogènes augmente l'accuracy moyenne de 69.81% à 71.32%.

Actuellement, nos modèles ne permettent pas des prédictions en temps réel. Pour y remédier, nous souhaitons nous focaliser dans nos futurs travaux sur le transfert des connaissances apprises par un de nos ensembles de modèles vers un réseau plus léger, qui permettrait idéalement une prédiction en temps réel, en appliquant la technique de la Distillation des Connaissances (Hinton, Vinyals et Dean, 2015).

Conclusion et Perspectives

Conclusion

La prédiction des comportements de foule permet aux forces de l'ordre d'anticiper des événements dangereux quelques secondes avant qu'ils ne surviennent. Une classification efficace de ces comportements permet aux forces de l'ordre de comprendre les mécanismes les régissant et de pouvoir les prédire, par la suite. L'objectif de cette thèse est de proposer un modèle de classification hors-ligne des comportements de foule.

Pour accomplir notre objectif, nous avons étudié les travaux existants dans le domaine de l'analyse de foule. Notre travail de recherche est détaillé dans l'état-de-l'art contenu dans le premier chapitre. À partir de notre étude des articles précédents sur l'état-de-l'art, nous avons établi une taxonomie permettant de classer tous les articles récents en analyse de foule et de comprendre pourquoi l'analyse des comportements de foule n'est pas assez développée. Nous nous sommes rendus compte, à travers notre étude, que le manque d'innovation dans ce domaine est imputable à la rareté de jeux de données vidéo illustrant une variété de comportements de foule. Cette indisponibilité peut être soit liée à une réticence à diffuser des données pouvant nuire à la confidentialité des personnes détectées et qui nécessiteraient une anonymisation (Flückiger et Auer, 2006 ; Dubois, 2016 ; Bouzit et Ghali, 2018) ; une anonymisation peut se faire via des opérations de floutage des visages détectés (Ruchaud et Dugelay, 2017), soit liée à des règles étatiques imposant une durée de conservation des données vidéo issues de ces systèmes (Lemaire, 2020).

Toutefois, nous avons des raisons de penser que ce manque de données est provisoire au vu de l'émergence récente de jeux de données semblables à Crowd-11 (Dupont, Tobias et Luvison, 2017), En attendant la mise en place de techniques d'acquisition de données vidéo de scènes de foule soucieuses de préserver l'anonymat des personnes filmées et de respecter la durée légale de conservation des données, nous avons concentré nos travaux sur Crowd-11. Dans le cadre d'un travail préliminaire, consigné dans la première partie du chapitre 2, nous avons mis à l'épreuve des modèles issus de trois architectures de réseaux de neurones convolutifs dans la classification des scènes de foule de Crowd-11. Dans la deuxième partie du chapitre 2, nous avons utilisé la détection de têtes comme étape de prétraitement permettant de focaliser l'attention de nos modèles sur le déplacement des personnes détectées.

Dans le troisième chapitre, nous avons orienté nos travaux sur la mise en place de méthodes ensemblistes de classification. À partir des modèles évalués dans le travail préliminaire du chapitre 2, nous avons constitué des modèles ensemblistes. Nous avons comparé des modèles ensemblistes homogènes avec des modèles individuels pour déterminer l'apport de la classification ensembliste. Par la suite, nous avons mis en place des modèles plus complexes, appelés modèles hétérogènes, mêlant des modèles ensemblistes issus d'architectures différentes, pour déterminer quelle est la combinaison de modèles ensemblistes qui obtient les meilleurs résultats.

Contributions

Dans le chapitre 2, nous avons d'abord proposé une solution basée sur l'apprentissage par transfert pour la classification des scènes de foule, nous avons comparé des modèles entraînés de zéro et des modèles ajustés sur le jeu de données Crowd-11. Ces modèles sont issus des réseaux 3D Convolutional Neural Nets (C3D), Inflated 3D (I3D), et TwoStream Inflated 3D (2S-I3D). (Dupont, Tobias et Luvison, 2017) avaient déjà établi un score de base assez élevé pour le modèle pré-entraîné C3D dans la classification de Crowd-11. Pour ce modèle, (Dupont, Tobias et Luvison, 2017) avaient obtenu le score de 62% d'accuracy. À la suite de notre comparaison, nous avons pu établir un nouveau score de base pour le jeu de données Crowd-11 grâce au modèle 2S-I3D. Désormais, ce score se situe aux alentours de 68% d'accuracy.

Dans la deuxième partie du chapitre 2, nous avons proposé de renforcer notre solution basée sur l'apprentissage par transfert avec une méthode de détection de têtes. Pour ce faire, nous avons utilisé l'algorithme LSC-CNN (Sam et al., 2020) pour détecter des têtes. Nous avons proposé de rassembler ces détections dans des cartes de détection de têtes de chaque vidéo de Crowd-11. Ces cartes ne gardent que les positions des centres de têtes dans l'image, et ignorent les pixels restants, ce qui incite un modèle à focaliser son attention sur le suivi des centres de têtes pour caractériser un comportement de foule. Les cartes de détection de têtes servent à entraîner des réseaux qui ont une, deux, ou trois branches. Chaque branche reçoit un type de données différent. Dans cette partie du chapitre, nous avons proposé de créer une architecture à trois branches, nommée ThreeStream-I3D (3S-I3D), dont chaque branche reprend l'architecture Inflated 3D (Carreira et Zisserman, 2017). Par la suite, nous avons comparé les réseaux issus de ces différentes architectures. À l'issue de l'entraînement à partir de zéro et de l'ajustement de ces réseaux, il s'est avéré que le réseau 3S-I3D réalise les meilleurs résultats, mais il n'est pas significativement plus efficace qu'un réseau 2S-I3D. Comme le réseau 3S-I3D est plus complexe et requiert davantage de temps de calcul pour l'apprentissage de la classification et la prédiction de la classe d'un clip vidéo, nous avons souhaité opter pour le réseau 2S-I3D pour la suite de nos travaux.

Dans le chapitre 3, nous avons proposé une solution basée sur la classification ensembliste. Nous avons entraîné des ensembles homogènes de modèles ayant la même architecture et ayant bénéficié des mêmes conditions d'entraînement. Nous avons comparé huit ensembles de modèles homogènes et avons trouvé que les ensembles de modèles 2S-I3D ajustés ayant bénéficié de l'augmentation de données obtiennent les meilleures performances lors de la classification. Par la suite, nous avons décidé d'évaluer, à la prédiction, toutes les combinaisons possibles de ces 8 ensembles de modèles homogènes. Nous avons trouvé que l'ensemble hétérogène constitué de la combinaison des ensembles issus des modèles 2S-I3D ajustés et ayant bénéficié de l'augmentation de données, des modèles 2S-I3D ajustés n'ayant pas bénéficié de l'augmentation de données, des modèles C3D ajustés, et des modèles I3D entraînés de zéro, obtient les meilleures performances en termes d'accuracy. Dans le cadre d'une distillation des connaissances (Hinton, Vinyals et Dean, 2015), cette combinaison de modèles hétérogènes pourrait servir de modèle maître pour entraîner des modèles étudiants, plus légers mais tout aussi performants, permettant de réaliser la classification temps réel.

Perspectives

Classification hors-ligne

Les modèles de classification hors-ligne des comportements de foule que nous proposons peuvent servir aux forces de l'ordre à comprendre les situations menant à certains incidents. Dans ce cadre, il serait utile de savoir interpréter les décisions d'un modèle de classification afin de mettre en perspective ses décisions avec les événements qui se déroulent réellement dans une scène de foule. Dans ce cadre, une catégorie de méthodes issue de l'Explainable AI peut être utilisée (Doran, Schulz et Besold, 2017) L'Explainable AI a pour vocation d'expliquer les décisions d'un modèle de classification en mettant en exergue les régions d'une scène qui ont influencé la décision du modèle. L'un des outils qui peut être utilisé dans ce cadre est iNNvestigate (Alber et al., 2019).

Par ailleurs, dans le cadre d'une formation des forces de l'ordre à la gestion des mouvements de foule, ces modèles de classification peuvent être couplés à des méthodes de simulation pour expliquer les mécanismes régissant les comportements de foule et préparer les forces de l'ordre à des situations réelles (Shendarkar et al., 2006 ; Crociani, Lämmel et Vizzari, 2016 ; Sharma et al., 2019).

Classification temps-réel

Les modèles que nous proposons sont perfectibles pour être utilisés dans une solution temps réel utile pour détecter des événements dangereux dans une scène à l'instar d'une foule en conflit ou d'un engorgement d'une scène de foule. Cette solution peut être mise en place soit à travers la recherche d'une architecture optimale à travers le Neural Architecture Search (Rakhshani et al., 2020), soit à travers la distillation des connaissances ; la distillation des connaissances consiste à entraîner un modèle étudiant d'un modèle maître dans le but de créer des modèles plus légers (Hinton, Vinyals et Dean, 2015).

Prédiction

À partir de nos travaux sur la classification hors-ligne des comportements de foule, il serait intéressant d'orienter la recherche vers la prédiction de ces comportements. Dans ce registre, nous avons déjà identifié des travaux portant sur la prédiction des trajectoires (Alahi et al., 2016 ; Bartoli et al., 2018). Nous pourrions appliquer cette prédiction de trajectoires sur le déplacement des têtes détectées par l'algorithme LSC-CNN (Sam et al., 2020). À partir des futures positions des têtes détectées obtenues via la prédiction de trajectoires, nous pourrions utiliser un modèle pré-entraîné sur la classification des cartes de détection pour prédire le futur comportement d'une foule avec un certain temps d'avance.

Bibliographie

Abousamra, Shahira et al. (2021). « Localization in the crowd with topological constraints ». In : *AAAI* (cf. p. 77, 81, 87).

Adam, Amit et al. (2008). « Robust real-time unusual event detection using multiple fixed-location monitors ». In : *IEEE transactions on pattern analysis and machine intelligence* 30.3, p. 555-560 (cf. p. 38).

Adelson, Edward H et James R Bergen (1985). « Spatiotemporal energy models for the perception of motion ». In : *Josa a* 2.2, p. 284-299 (cf. p. 41).

Aggarwal, Charu C (2004). « A human-computer interactive method for projected clustering ». In : *IEEE transactions on knowledge and data engineering* 16.4, p. 448-460 (cf. p. 36).

Alahi, Alexandre, Vignesh Ramanathan et Li Fei-Fei (2017). « Tracking millions of humans in crowded spaces ». In : *Group and Crowd Behavior for Computer Vision*. Elsevier, p. 115-135 (cf. p. 32).

Alahi, Alexandre et al. (2016). « Social lstm : Human trajectory prediction in crowded spaces ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 961-971 (cf. p. 35-37, 43, 111).

Alameda-Pineda, Xavier et al. (2016). « Salsa : A novel dataset for multimodal group behavior analysis ». In : *IEEE transactions on pattern analysis and machine intelligence* 38.8, p. 1707-1720 (cf. p. 51, 53, 55).

Alber, Maximilian et al. (2019). « iNNvestigate Neural Networks ! » In : *Journal of Machine Learning Research* 20.93, p. 1-8. URL : http://jmlr.org/papers/v20/18-540.html (cf. p. 111).

Ali, Saad et Mubarak Shah (2007). « A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis ». In : *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, p. 1-6 (cf. p. 31, 43).

— (2008). « Floor fields for tracking in high density crowd scenes ». In : *European conference on computer vision*, p. 1-14 (cf. p. 32, 43, 46, 55).

Allain, Pierre, Nicolas Courty et Thomas Corpetti (2012). « AGORASET : a dataset for crowd video analysis ». In : *1st ICPR International Workshop on Pattern Recognition and Crowd Analysis*, p. 1-6 (cf. p. 18, 48, 53, 55).

Andriluka, Mykhaylo, Jasper RR Uijlings et Vittorio Ferrari (2018). « Fluid Annotation : a human-machine collaboration interface for full image annotation ». In : *ACM Multimedia Conference on Multimedia Conference*, p. 1957-1966 (cf. p. 57-59).

Angelova, Anelia et al. (2015). « Real-Time Pedestrian Detection with Deep Network Cascades ». In : *BMVC*. T. 2, p. 4 (cf. p. 20, 21, 23).

Aqeel, Muhammad et al. (2020). « Detection of Anomaly in Videos Using Convolutional Autoencoder and Generative Adversarial Network model ». In : *2020 IEEE 23rd International Multitopic Conference (INMIC)*. IEEE, p. 1-6 (cf. p. 69, 73).

Aref, Hassan (1990). « Chaotic advection of fluid particles ». In : *Philosophical Transactions of the Royal Society of London. Series A : Physical and Engineering Sciences* 333.1631, p. 273-288 (cf. p. 32).

AVSS (2007). *2007 IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS 2007)* (cf. p. 55).

Azorin-Lopez, J. et al. (2015). « Self-Organizing Activity Description Map to represent and classify human behaviour ». In : *2015 International Joint Conference on Neural Networks (IJCNN)*, p. 1-7 (cf. p. 17).

Azorin-Lopez, J. et al. (2016). « Group activity description and recognition based on trajectory analysis and neural networks ». In : *2016 International Joint Conference on Neural Networks (IJCNN)*, p. 1585-1592 (cf. p. 17).

Baccouche, Moez et al. (2011). « Sequential deep learning for human action recognition ». In : *International workshop on human behavior understanding*. Springer, p. 29-39 (cf. p. 28, 43).

Bartoli, Federico et al. (2015a). « Museumvisitors : a dataset for pedestrian and group detection, gaze estimation and behavior understanding ». In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, p. 19-27 (cf. p. 51, 53, 55).

Bartoli, Federico et al. (2015b). « Watts : a web annotation tool for surveillance scenarios ». In : *Proceedings of the 23rd ACM international conference on Multimedia*, p. 701-704 (cf. p. 57, 59).

Bartoli, Federico et al. (2017). « PACE : Prediction-based Annotation for Crowded Environments ». In : *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, p. 121-124 (cf. p. 43, 56, 57, 59).

Bartoli, Federico et al. (2018). « Context-aware trajectory prediction ». In : *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, p. 1941-1946 (cf. p. 35, 36, 111).

Bazzani, Loris, Marco Cristani et Vittorio Murino (2012). « Decentralized particle filter for joint individual-group tracking ». In : *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, p. 1886-1893 (cf. p. 50, 53, 55).

Bendali-Braham, Mounir et al. (2019). « Transfer learning for the classification of video-recorded crowd movements ». In : *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)*. IEEE, p. 271-276 (cf. p. 64, 89, 92).

Benenson, Rodrigo et al. (2012). « Pedestrian detection at 100 frames per second ». In : *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, p. 2903-2910 (cf. p. 21).

Benfold, Ben et Ian Reid (2011). « Stable multi-target tracking in real-time surveillance video ». In : *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, p. 3457-3464 (cf. p. 37, 45, 55).

Bera, Aniket, Sujeong Kim et Dinesh Manocha (2018). « Modeling Trajectory-level Behaviors using Time Varying Pedestrian Movement Dynamics ». In : *Collective Dynamics* 3, p. 1-23 (cf. p. 32).

Bewley, Alex et al. (2016). « Simple online and realtime tracking ». In : *Proceedings - International Conference on Image Processing, ICIP* 2016-Augus, p. 3464-3468 (cf. p. 32, 33, 35, 43).

Bisagno, Niccoló, Bo Zhang et Nicola Conci (2018). « Group lstm : Group trajectory prediction in crowded scenarios ». In : *Proceedings of the European conference on computer vision (ECCV)*, p. 0-0 (cf. p. 37, 43).

Bloom, Victoria, Dimitrios Makris et Vasileios Argyriou (2012). « G3D : A gaming action dataset and real time action recognition evaluation framework ». In : *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, p. 7-12 (cf. p. 78).

Blunsden, Scott et RB Fisher (2010). « The BEHAVE video dataset : ground truthed video for multi-person behavior classification ». In : *Annals of the BMVA* 4.1-12, p. 4 (cf. p. 51, 53, 55).

Bolia, Nomesh B (2015). « Risk management strategies to avoid stampede at Mass gatherings ». In : *2nd World Conference on Disaster Management : Visakhapatnam, Andhra Pradesh, India* (cf. p. 1).

Borja-Borja, Luis Felipe, Marcelo Saval-Calvo et Jorge Azorin-Lopez (2017). « Machine Learning Methods from Group to Crowd Behaviour Analysis ». In : *Advances in Computational Intelligence*. Sous la dir. d'Ignacio Rojas, Gonzalo Joya et Andreu Catala, p. 294-305 (cf. p. 17, 18).

Bouzit, Mhammed et Abdelkrim Ghali (2018). « NOUVEAUX RISQUES DES NTIC : QUEL CADRE JURIDIQUE POUR LE BIG DATA AU MAROC ? » In : *International journal of advanced research* 6.4, p. 317-323 (cf. p. 109).

Branch, Home Office Scientific Development (2006). « Imagery library for intelligent detection systems (i-lids) ». In : *2006 IET Conference on Crime and Security*. IET, p. 445-448 (cf. p. 28).

Brostow, Gabriel J et Roberto Cipolla (2006). « Unsupervised bayesian detection of independent motion in crowds ». In : *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. T. 1. IEEE, p. 594-601 (cf. p. 20).

Canny, John (1986). « A computational approach to edge detection ». In : *IEEE Transactions on pattern analysis and machine intelligence* 6, p. 679-698 (cf. p. 24).

Carreira, Joao et Andrew Zisserman (2017). « Quo vadis, action recognition ? a new model and the kinetics dataset ». In : *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, p. 4724-4733 (cf. p. 2, 28, 43, 61, 63-65, 89, 92, 110).

Chaaraoui, Alexandros André, Pau Climent-Pérez et Francisco Flórez-Revuelta (2012). « A review on vision techniques applied to Human Behaviour Analysis for Ambient-Assisted Living ». In : *Expert Systems with Applications* 39.12, p. 10873 -10888 (cf. p. 17, 18).

Chan, Antoni B, Zhang-Sheng John Liang et Nuno Vasconcelos (2008). « Privacy preserving crowd monitoring : Counting people without people models or tracking ». In : *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, p. 1-7 (cf. p. 24, 43, 48).

Chan, Antoni B et Nuno Vasconcelos (2008). « Modeling, clustering, and segmenting video with mixtures of dynamic textures ». In : *IEEE transactions on pattern analysis and machine intelligence* 30.5, p. 909-926 (cf. p. 36, 38).

Chen, Chen, Roozbeh Jafari et Nasser Kehtarnavaz (2015). « UTD-MHAD : A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor ». In : *2015 IEEE International conference on image processing (ICIP)*. IEEE, p. 168-172 (cf. p. 78).

Chen, Guang et al. (2014). « Action recognition using ensemble weighted multi-instance learning ». In : *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, p. 4520-4525 (cf. p. 90).

Chen, Mulin, Qi Wang et Xuelong Li (2017a). « Anchor-based group detection in crowd scenes ». In : *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, p. 1378-1382 (cf. p. 22).

— (2017b). « Patch-based topic model for group detection ». In : *Science China Information Sciences* 60.11, p. 113101 (cf. p. 22).

Choi, Wongun, Khuram Shahid et Silvio Savarese (2009). « What are they doing ? : Collective activity classification using spatio-temporal relationship among

people ». In : *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, p. 1282-1289 (cf. p. 36, 37).

Chong, YS et YH Tay (2015). *Modeling Representation of Videos for Anomaly Detection using Deep Learning : A Review*. ArXiv :1505.00523 (cf. p. 12, 14, 27).

Cortes, Corinna et Vladimir Vapnik (1995). « Support-vector networks ». In : *Machine learning* 20.3, p. 273-297 (cf. p. 90).

Coscia, Pasquale et al. (2018). « Long-term path prediction in urban scenarios using circular distributions ». In : *Image and Vision Computing* 69, p. 81-91 (cf. p. 36).

Crociani, Luca, Gregor Lämmel et Giuseppe Vizzari (2016). « Multi-scale simulation for crowd management : a case study in an urban scenario ». In : *International Conference on Autonomous Agents and Multiagent Systems*. Springer, p. 147-162 (cf. p. 111).

Cui, Jinshi et al. (2008). « Multi-modal tracking of people using laser scanners and video camera ». In : *Image and vision Computing* 26.2, p. 240-252 (cf. p. 16).

Dalal, Navneet et Bill Triggs (2005). « Histograms of oriented gradients for human detection ». In : *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on.* T. 1, p. 886-893 (cf. p. 20, 47, 55, 58).

Davies, Anthony C, Jia Hong Yin et Sergio A Velastin (1995). « Crowd monitoring using image processing ». In : *Electronics & Communication Engineering Journal* 7.1, p. 37-47 (cf. p. 24).

De Almeida, Paulo RL et al. (2015). « PKLot–A robust dataset for parking lot classification ». In : *Expert Systems with Applications* 42.11, p. 4937-4949 (cf. p. 78).

De Smedt, Quentin, Hazem Wannous et Jean-Philippe Vandeborre (2016). « 3d hand gesture recognition by analysing set-of-joints trajectories ». In : *International Workshop on Understanding Human Activities through 3D Sensors*. Springer, p. 86-97 (cf. p. 78).

Dehghan, Afshin, Shayan Modiri Assari et Mubarak Shah (2015). « Gmmcp tracker : Globally optimal generalized maximum multi clique problem for multiple object tracking ». In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 4091-4099 (cf. p. 37).

Deng, Jia et al. (2009). « Imagenet : A large-scale hierarchical image database ». In : *IEEE conference on computer vision and pattern recognition*, p. 248-255 (cf. p. 61, 64, 76, 90, 95).

Dietterich, Thomas G (2000). « Ensemble methods in machine learning ». In : *International workshop on multiple classifier systems*. Springer, p. 1-15 (cf. p. 90).

Dollar, Piotr et al. (2012). « Pedestrian detection : An evaluation of the state of the art ». In : *IEEE transactions on pattern analysis and machine intelligence* 34.4, p. 743-761 (cf. p. 21, 47, 55).

Dollár, Piotr et al. (2014). « Fast feature pyramids for object detection ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.8, p. 1532-1545 (cf. p. 22).

Doran, Derek, Sarah Schulz et Tarek R Besold (2017). « What does explainable AI really mean ? A new conceptualization of perspectives ». In : *arXiv preprint arXiv :1710.00794* (cf. p. 111).

Doretto, Gianfranco et al. (2003). « Dynamic textures ». In : *International Journal of Computer Vision* 51.2, p. 91-109 (cf. p. 24).

Doucet, Arnaud et Adam M Johansen (2009). « A tutorial on particle filtering and smoothing : Fifteen years later ». In : *Handbook of nonlinear filtering* 12.656-704, p. 3 (cf. p. 20).

Dubois, Jean-Pierre (2016). « Nos droits face aux «big data» : quels enjeux, quels risques, quelles garanties ? » In : *Apres-demain* 1, p. 6-9 (cf. p. 109).

Dupont, C., L. Tobias et B. Luvison (2017). « Crowd-11 : A Dataset for Fine Grained Crowd Behaviour Analysis ». In : *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. T. 2017-July. Honolulu, United States, p. 2184-2191. URL : https://hal-cea.archives-ouvertes.fr/cea-01831840 (cf. p. 2, 31, 50, 55, 60-64, 66, 67, 89, 109, 110).

Dutta, Abhishek et Andrew Zisserman (2019). *The VGG Image Annotator (VIA)*. ArXiv :1904.10699 (cf. p. 57-59).

Dvornik, Nikita, Julien Mairal et Cordelia Schmid (2018). « On the Importance of Visual Context for Data Augmentation in Scene Understanding ». In : *ArXiv*. eprint : 1809.02492 (cf. p. 66).

ED (2003). *CAVIAR*. URL : http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/ (cf. p. 49, 55).

Edelsbrunner, Herbert et John Harer (2010). *Computational topology : an introduction*. American Mathematical Soc. (cf. p. 77).

Efron, Bradley (1992). « Bootstrap methods : another look at the jackknife ». In : *Breakthroughs in statistics*. Springer, p. 569-593 (cf. p. 90).

Emonet, Rémi, Jagannadan Varadarajan et Jean-Marc Odobez (2011). « Extracting and locating temporal motifs in video scenes using a hierarchical non parametric bayesian model ». In : *CVPR 2011*. IEEE, p. 3233-3240 (cf. p. 30).

Farnebäck, Gunnar (2003). « Two-frame motion estimation based on polynomial expansion ». In : *Scandinavian conference on Image analysis*. Springer, p. 363-370 (cf. p. 31).

Favaretto, Rodolfo Migon, Leandro Lorenzett Dihl et Soraia Raupp Musse (2016). « Detecting crowd features in video sequences ». In : *2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE, p. 201-208 (cf. p. 34).

Fawaz, Hassan Ismail et al. (2019). « Deep neural network ensembles for time series classification ». In : *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, p. 1-6 (cf. p. 89).

Felzenszwalb, Pedro F et al. (2009). « Object detection with discriminatively trained part-based models ». In : *IEEE transactions on pattern analysis and machine intelligence* 32.9, p. 1627-1645 (cf. p. 91).

Ferryman, James et Ali Shahrokni (2009). « Pets2009 : Dataset and challenge ». In : *2009 Twelfth IEEE international workshop on performance evaluation of tracking and surveillance*. IEEE, p. 1-6 (cf. p. 37).

FHWA (2004). *Traffic analysis tools primer,traffic analysis toolbox*. URL : http://ops.fhwa.dot.gov/trafficanalysistools/tat-vol1/index (cf. p. 7).

Flückiger, Alexandre et Andreas Auer (2006). « La vidéosurveillance dans l'oeil de la Constitution ». In : *Pratique juridique actuelle* 8, p. 924-942 (cf. p. 109).

Fothergill, Simon et al. (2012). « Instructing people for training gestural interactive systems ». In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p. 1737-1746 (cf. p. 78).

Freund, Yoav et Robert E Schapire (1995). « A desicion-theoretic generalization of on-line learning and an application to boosting ». In : *European conference on computational learning theory*. Springer, p. 23-37 (cf. p. 89, 90).

Gao, Zan et al. (2010). « Comparing evaluation protocols on the KTH dataset ». In : *International Workshop on Human Behavior Understanding*. Springer, p. 88-100 (cf. p. 28).

Ge, Weina, Robert T Collins et Barry Ruback (2009). « Automatically detecting the small group structure of a crowd ». In : *2009 Workshop on Applications of Computer Vision (WACV)*. IEEE, p. 1-8 (cf. p. 34).

Ge, Weina, Robert T Collins et R Barry Ruback (2012). « Vision-based analysis of small groups in pedestrian crowds ». In : *IEEE transactions on pattern analysis and machine intelligence* 34.5, p. 1003-1016 (cf. p. 20, 23, 34, 36).

Geiger, Andreas et al. (2013). « Vision meets robotics : The KITTI dataset ». In : *The International Journal of Robotics Research* 32.11, p. 1231-1237 (cf. p. 21, 45, 55).

Gers, Felix A, Nicol N Schraudolph et Jürgen Schmidhuber (2002). « Learning precise timing with LSTM recurrent networks ». In : *Journal of machine learning research* 3.Aug, p. 115-143 (cf. p. 28).

Girdhar, Palak, Prashant Johri et Deepali Virmani (2020). « Incept_LSTM : Accession for human activity concession in automatic surveillance ». In : *Journal of Discrete Mathematical Sciences and Cryptography*, p. 1-15 (cf. p. 69, 71, 74).

Girshick, Ross et al. (2014). « Rich feature hierarchies for accurate object detection and semantic segmentation ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 580-587 (cf. p. 90).

Godbehere, Andrew B, Akihiro Matsukawa et Ken Goldberg (2012). « Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation ». In : *American Control Conference (ACC), 2012*, p. 4305-4312 (cf. p. 58).

Gong, Shaogang, Tao Xiang et Somboon Hongeng (2010). « Learning human pose in crowd ». In : *Proceedings of the 1st ACM international workshop on Multimodal pervasive video analysis*, p. 47-52 (cf. p. 91).

Grant, Jason M et Patrick J Flynn (2017). « Crowd scene understanding from video : a survey ». In : *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 13.2, p. 19 (cf. p. 5-7, 9, 10, 23).

Guerrero-Gómez-Olmedo, Ricardo et al. (2015). « Extremely overlapping vehicle counting ». In : *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, p. 423-431 (cf. p. 76, 77).

Güler, Püren (2012). « Automated crowd behavior analysis for video surveillance applications ». Mém. de mast. Middle East Technical University (cf. p. 1).

Gupta, Agrim et al. (2018). « Social gan : Socially acceptable trajectories with generative adversarial networks ». In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 2255-2264 (cf. p. 37).

Hao, Yu et al. (2019). « Effective crowd anomaly detection through spatio-temporal texture analysis ». In : *International Journal of Automation and Computing* 16.1, p. 27-39 (cf. p. 41, 43).

Hara, Kensho, Hirokatsu Kataoka et Yutaka Satoh (2017). « Learning spatio-temporal features with 3d residual networks for action recognition ». In : *Proceedings of the IEEE International Conference on Computer Vision Workshops*, p. 3154-3160 (cf. p. 80, 92, 145).

Haralick, Robert M, Karthikeyan Shanmugam et Its' Hak Dinstein (1973). « Textural features for image classification ». In : *IEEE Transactions on systems, man, and cybernetics* 6, p. 610-621 (cf. p. 24, 41).

Hasan, Mahmudul et al. (2016). « Learning temporal regularity in video sequences ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 733-742 (cf. p. 69, 70, 72).

Hassner, Tal, Yossi Itcher et Orit Kliper-Gross (2012). « Violent flows : Real-time detection of violent crowd behavior ». In : *Computer Vision and Pattern*

Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, p. 1-6 (cf. p. 27, 38, 43, 49, 55).

He, Kaiming et al. (2015). « Spatial pyramid pooling in deep convolutional networks for visual recognition ». In : *IEEE transactions on pattern analysis and machine intelligence* 37.9, p. 1904-1916 (cf. p. 25).

— (2016). « Deep residual learning for image recognition ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 770-778 (cf. p. 24, 90).

Helbing, Dirk, Illés Farkas et Tamas Vicsek (2000). « Simulating dynamical features of escape panic ». In : *Nature* 407.6803, p. 487 (cf. p. 48).

Helbing, Dirk et Peter Molnar (1995). « Social force model for pedestrian dynamics ». In : *Physical review E* 51.5, p. 4282 (cf. p. 12, 22).

Hinton, Geoffrey, Oriol Vinyals et Jeff Dean (2015). « Distilling the knowledge in a neural network ». In : *ArXiv*. eprint : 1503.02531 (cf. p. 107, 110, 111).

Hou, Rui, Chen Chen et Mubarak Shah (2017). « Tube convolutional neural network (t-cnn) for action detection in videos ». In : *Proceedings of the IEEE international conference on computer vision*, p. 5822-5831 (cf. p. 70).

Hou, Yonghong et al. (2016). « Skeleton optical spectra-based action recognition using convolutional neural networks ». In : *IEEE Transactions on Circuits and Systems for Video Technology* 28.3, p. 807-811 (cf. p. 78).

Hsieh, Meng-Ru, Yen-Liang Lin et Winston H Hsu (2017). « Drone-based object counting by spatially regularized regional proposal network ». In : *Proceedings of the IEEE international conference on computer vision*, p. 4145-4153 (cf. p. 78).

Hu, Xing et al. (2018). « Squirrel-Cage Local Binary Pattern and Its Application in Video Anomaly Detection ». In : *IEEE Transactions on Information Forensics and Security* 14.4, p. 1007-1022 (cf. p. 40).

Huang, Chang, R. Nevatia et Yuan Li (2009). « Learning to associate : HybridBoosted multi-target tracker for crowded scene ». In : *2009 IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*. T. 00, p. 2953-2960 (cf. p. 45).

Hussein, Mohamed E et al. (2013). « Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations ». In : *Twenty-third international joint conference on artificial intelligence* (cf. p. 78).

Ibrahim, Mostafa S et al. (2016). « A hierarchical deep temporal model for group activity recognition ». In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 1971-1980 (cf. p. 37).

Idrees, Haroon et al. (2013). « Multi-source multi-scale counting in extremely dense crowd images ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 2547-2554 (cf. p. 25, 55, 76, 88).

Idrees, Haroon et al. (2018). « Composition loss for counting, density map estimation and localization in dense crowds ». In : *Proceedings of the European Conference on Computer Vision (ECCV)*, p. 532-546 (cf. p. 76, 88).

Iqbal, Haris (déc. 2018). *HarisIqbal88/PlotNeuralNet v1.0.0*. doi : 10.5281/zenodo.2526396. url : https://doi.org/10.5281/zenodo.2526396 (cf. p. 80).

Isern, Juan et al. (2020). « Reconfigurable cyber-physical system for critical infrastructure protection in smart cities via smart video-surveillance ». In : *Pattern Recognition Letters* 140, p. 303-309 (cf. p. 5).

Jia, Yangqing et al. (2014). « Caffe : Convolutional architecture for fast feature embedding ». In : *Proceedings of the 22nd ACM international conference on Multimedia*, p. 675-678 (cf. p. 90).

John, Oliver P et Sanjay Srivastava (1999). « The Big Five trait taxonomy : History, measurement, and theoretical perspectives ». In : *Handbook of personality : Theory and research* 2.1999, p. 102-138 (cf. p. 52).

Kalman, Rudolph Emil (1960). « A new approach to linear filtering and prediction problems ». In : *Journal of basic Engineering* 82.1, p. 35-45 (cf. p. 32, 58).

Kang, Di et Antoni Chan (2018). *Crowd counting by adaptively fusing predictions from an image pyramid*. ArXiv :1805.06115 (cf. p. 26).

Karamouzas, Ioannis et al. (2009). « A predictive collision avoidance model for pedestrian simulation ». In : *International Workshop on Motion in Games*, p. 41-52 (cf. p. 22).

Karpathy, Andrej et al. (2014). « Large-scale video classification with convolutional neural networks ». In : *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, p. 1725-1732 (cf. p. 48, 55, 64, 95).

Kasturi, Rangachar et al. (2009). « Framework for performance evaluation of face, text, and vehicle detection and tracking in video : Data, metrics, and protocol ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.2, p. 319-336 (cf. p. 45).

Kay, Will et al. (2017). « The kinetics human action video dataset ». In : *ArXiv*. eprint : 1705.06950 (cf. p. 47, 55, 61, 64, 95).

Ke, Qiuhong et al. (2017). « A new representation of skeleton sequences for 3d action recognition ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 3288-3297 (cf. p. 78).

Ke, Yan, Rahul Sukthankar et Martial Hebert (2007). « Event detection in crowded videos ». In : *2007 IEEE 11th International Conference on Computer Vision*. IEEE, p. 1-8 (cf. p. 28).

Kim, Jaechul et Kristen Grauman (2009). « Observe locally, infer globally : a space-time MRF for detecting abnormal activities with incremental updates ». In : *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, p. 2921-2928 (cf. p. 38).

Kiran, B Ravi, Dilip Mathew Thomas et Ranjith Parakkal (2018). « An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos ». In : *Journal of Imaging* 4.2, p. 36 (cf. p. 14, 15).

Kong, Dan, Douglas Gray et Hai Tao (2005). « Counting Pedestrians in Crowds Using Viewpoint Invariant Training. » In : *BMVC*. T. 1. Citeseer, p. 2 (cf. p. 24).

Krausz, Barbara et Christian Bauckhage (2012). « Loveparade 2010 : Automatic video analysis of a crowd disaster ». In : *Computer Vision and Image Understanding* 116.3, p. 307-319 (cf. p. 1, 5, 59).

Kritter, Julien et al. (2019). « On the optimal placement of cameras for surveillance and the underlying set cover problem ». In : *Applied Soft Computing* 74, p. 133-153 (cf. p. 1).

Krizhevsky, Alex, Ilya Sutskever et Geoffrey E Hinton (2012). « Imagenet classification with deep convolutional neural networks ». In : *Advances in neural information processing systems*, p. 1097-1105 (cf. p. 1, 6, 19, 20, 39, 40, 77, 78, 90).

Kuehne, Hildegard et al. (2011). « HMDB : a large video database for human motion recognition ». In : *Computer Vision (ICCV), 2011 IEEE International Conference on*, p. 2556-2563 (cf. p. 47, 55).

Kuhn, Harold W (1955). « The Hungarian method for the assignment problem ». In : *Naval research logistics quarterly* 2.1-2, p. 83-97 (cf. p. 20, 32).

Lamba, Sonu et Neeta Nain (2017). « Crowd monitoring and classification : a survey ». In : *Advances in Computer and Computational Sciences*. Springer, p. 21-31 (cf. p. 5, 6, 8-10, 23).

— (2019). « Segmentation of crowd flow by trajectory clustering in active contours ». In : *The Visual Computer*, p. 1-12 (cf. p. 33, 35, 43).

Landi, Federico, Cees GM Snoek et Rita Cucchiara (2019). « Anomaly locality in video surveillance ». In : *arXiv preprint arXiv :1901.10364* (cf. p. 69, 71).

Laptev, Ivan (2005). « On space-time interest points ». In : *International journal of computer vision* 64.2-3, p. 107-123 (cf. p. 39).

Laptev, Ivan et al. (2008). « Learning realistic human actions from movies ». In : *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, p. 1-8 (cf. p. 27).

Latour, Xavier et Bertrand Pauvert (2018). *Libertés publiques et droits fondamentaux*. Studyrama. 468 p. (cf. p. 5).

Leal-Taixé, L. et al. (2015). *MOTChallenge 2015 : Towards a Benchmark for Multi-Target Tracking*. ArXiv :1504.01942 (cf. p. 33, 45, 58).

LeCun, Yann, Yoshua Bengio et al. (1995). « Convolutional networks for images, speech, and time series ». In : *The handbook of brain theory and neural networks* 3361.10, p. 1995 (cf. p. 90).

Lemaire, Élodie (2020). « Sommes-nous vidéo-protégé·es ? » In : *Deliberee* 2, p. 83-87 (cf. p. 109).

Lerner, Alon, Yiorgos Chrysanthou et Dani Lischinski (2007). « Crowds by example ». In : *Computer graphics forum*. T. 26. 3. Wiley Online Library, p. 655-664 (cf. p. 35).

Leyva, R., V. Sanchez et C. Li (2017). « The LV dataset : A realistic surveillance video dataset for abnormal event detection ». In : *2017 5th International Workshop on Biometrics and Forensics (IWBF)*, p. 1-6 (cf. p. 14).

Li, Bo, Xiaojie Liang et Lianbao Jin (2016). « Video classification via spatial-temporal subspace learning ». In : *2016 6th International Conference on Digital Home (ICDH)*. IEEE, p. 38-43 (cf. p. 31).

Li, Hailong, Zhendong Wu et Jianwu Zhang (2016). « Pedestrian detection based on deep learning model ». In : *Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), International Congress on*, p. 796-800 (cf. p. 20, 23).

Li, Teng et al. (2015). « Crowded Scene Analysis : A Survey ». In : *Circuits and Systems for Video Technology, IEEE Transactions on* 25.3, p. 367-386 (cf. p. 6, 11, 13).

Li, Weixin, Vijay Mahadevan et Nuno Vasconcelos (2014). « Anomaly detection and localization in crowded scenes ». In : *IEEE transactions on pattern analysis and machine intelligence* 36.1, p. 18-32 (cf. p. 25).

Li, Xuelong et al. (2017). « A Multiview-Based Parameter Free Framework for Group Detection. » In : *AAAI*, p. 4147-4153 (cf. p. 22).

Li, Yan et al. (2019). « A top-bottom clustering algorithm based on crowd trajectories for small group classification ». In : *IEEE Access* 7, p. 29679-29698 (cf. p. 34, 35, 43).

Li, Yuhong, Xiaofan Zhang et Deming Chen (2018). « Csrnet : Dilated convolutional neural networks for understanding the highly congested scenes ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 1091-1100 (cf. p. 26, 77).

Lin, Shuheng et al. (2019). « Social MIL : Interaction-Aware for Crowd Anomaly Detection ». In : *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, p. 1-8 (cf. p. 41, 43).

Lin, Tsung-Yi et al. (2014). « Microsoft coco : Common objects in context ». In : *European conference on computer vision*, p. 740-755 (cf. p. 47, 55).

Lin, Tsung Yi et al. (2017). « Feature pyramid networks for object detection ». In : *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition* 2017-Janua, p. 936-944 (cf. p. 39).

Liu, Jun et al. (2016a). « Spatio-temporal lstm with trust gates for 3d human action recognition ». In : *European conference on computer vision*. Springer, p. 816-833 (cf. p. 78).

Liu, Wei et al. (2016b). « Ssd : Single shot multibox detector ». In : *European conference on computer vision*, p. 21-37 (cf. p. 22).

Liu, Wei et al. (2017). « An ensemble deep learning method for vehicle type classification on visual traffic surveillance sensors ». In : *IEEE Access* 5, p. 24417-24425 (cf. p. 90, 91).

Liu, Weizhe, Mathieu Salzmann et Pascal Fua (2019). « Context-aware crowd counting ». In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 5099-5108 (cf. p. 25, 43).

Liu, Xu-Ying, Jianxin Wu et Zhi-Hua Zhou (2008). « Exploratory undersampling for class-imbalance learning ». In : *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39.2, p. 539-550 (cf. p. 90, 91).

Liu, Yuting et al. (2019). « Point in, box out : Beyond counting persons in crowds ». In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 6469-6478 (cf. p. 77, 78).

Lu, Cewu, Jianping Shi et Jiaya Jia (2013). « Abnormal event detection at 150 fps in matlab ». In : *Proceedings of the IEEE international conference on computer vision*, p. 2720-2727 (cf. p. 69, 70, 72).

Lucas, Bruce D, Takeo Kanade et al. (1981). « An iterative image registration technique with an application to stereo vision ». In : (cf. p. 34).

Luo, Ping et al. (2014). « Switchable deep network for pedestrian detection ». In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 899-906 (cf. p. 21).

Lv, Hui et al. (2021). « Localizing anomalies from weakly-labeled videos ». In : *IEEE transactions on image processing* (cf. p. 69, 73).

Lyu, Siwei et al. (2017). « UA-DETRAC 2017 : Report of AVSS2017 & IWT4S Challenge on Advanced Traffic Monitoring ». In : *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*, p. 1-7 (cf. p. 42).

Mahadevan, Vijay et al. (2010). « Anomaly detection in crowded scenes ». In : *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, p. 1975-1981 (cf. p. 38, 43).

Malon, Thierry et al. (2018). « Toulouse campus surveillance dataset : scenarios, soundtracks, synchronized videos with overlapping and disjoint views ». In : *Proceedings of the 9th ACM Multimedia Systems Conference*, p. 393-398 (cf. p. 46, 55).

Manen, Santiago et al. (2017). « Pathtrack : Fast trajectory annotation with path supervision ». In : *2017 IEEE International Conference on Computer Vision (ICCV)*, p. 290-299 (cf. p. 46, 55, 56, 58, 59).

Marsden, Mark et al. (2016a). *Fully convolutional crowd counting on highly congested scenes*. ArXiv :1612.00220 (cf. p. 24).

— (2016b). « Holistic features for real-time crowd behaviour anomaly detection ». In : *Image Processing (ICIP), 2016 IEEE International Conference on*, p. 918-922 (cf. p. 25).

Marsden, Mark et al. (2017). « ResnetCrowd : A residual deep learning architecture for crowd counting, violent behaviour detection and crowd density level classification ». In : *IEEE International Conference on Advanced Video and Signal Based Surveillance* (cf. p. 24, 26, 27, 43, 44, 55).

Mehran, Ramin, Alexis Oyama et Mubarak Shah (2009). « Abnormal crowd behavior detection using social force model ». In : *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, p. 935-942 (cf. p. 38, 40, 41, 43, 55).

Mettes, Pascal, Jan C van Gemert et Cees GM Snoek (2016). « Spot on : Action localization from pointly-supervised proposals ». In : *European Conference on Computer Vision*, p. 437-453 (cf. p. 56, 58, 59).

Milan, A. et al. (2016). *MOT16 : A Benchmark for Multi-Object Tracking*. ArXiv :1603.00831 (cf. p. 33, 45, 53, 55).

Milan, Anton, Stefan Roth et Konrad Schindler (2014). « Continuous energy minimization for multitarget tracking. » In : *IEEE Trans. Pattern Anal. Mach. Intell.* 36.1, p. 58-72 (cf. p. 22).

Mousavi, Hossein et al. (2015). « Crowd motion monitoring using tracklet-based commotion measure ». In : *Image Processing (ICIP), 2015 IEEE International Conference on*, p. 2354-2358 (cf. p. 39).

Munder, Stefan et Dariu M Gavrila (2006). « An experimental study on pedestrian classification ». In : *IEEE transactions on pattern analysis and machine intelligence* 28.11, p. 1863-1868 (cf. p. 21).

Onoro-Rubio, Daniel et Roberto J López-Sastre (2016). « Towards perspective-free object counting with deep learning ». In : *European Conference on Computer Vision*, p. 615-629 (cf. p. 25).

Ordóñez, Francisco Javier et Daniel Roggen (2016). « Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition ». In : *Sensors* 16.1, p. 115 (cf. p. 71).

Pan, Sinno Jialin et Qiang Yang (2010). « A survey on transfer learning ». In : *IEEE Transactions on knowledge and data engineering* 22.10, p. 1345-1359 (cf. p. 63).

Pellegrini, Stefano et al. (2009). « You'll never walk alone : Modeling social behavior for multi-target tracking ». In : *2009 IEEE 12th International Conference on Computer Vision*. IEEE, p. 261-268 (cf. p. 21, 35).

Penmetsa, Surya et al. (2014). « Autonomous UAV for suspicious action detection using pictorial human pose estimation and classification ». In : *Electronic Letters on Computer Vision and Image Analysis* 13.1, p. 18-32 (cf. p. 39).

Perez, Mauricio, Alex C Kot et Anderson Rocha (2019). « Detection of Real-world Fights in Surveillance Videos ». In : *IEEE International Conference on Acoustics, Speech and Signal Processing*, p. 2662-2666 (cf. p. 49, 50, 55).

Peter, Trautman et al. (2013). « Robot navigation in dense human crowds : the case for cooperation ». In : *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, p. 2153-2160 (cf. p. 35).

Porikli, Fatih et al. (2013). « Video surveillance : past, present, and now the future [DSP Forum] ». In : *IEEE Signal Processing Magazine* 30.3, p. 190-198 (cf. p. 5).

Pouyanfar, Samira et Shu-Ching Chen (2016). « Semantic concept detection using weighted discretization multiple correspondence analysis for disaster information management ». In : *2016 IEEE 17th International Conference on Information Reuse and Integration (IRI)*. IEEE, p. 556-564 (cf. p. 90).

Pouyanfar, Samira et Shu-Ching Chen (2017). « Automatic video event detection for imbalance data using enhanced ensemble deep learning ». In : *International Journal of Semantic Computing* 11.01, p. 85-109 (cf. p. 90).

Qasim, Tehreem et Naeem Bhatti (2019). « A low dimensional descriptor for detection of anomalies in crowd videos ». In : *Mathematics and Computers in Simulation* 166, p. 245-252 (cf. p. 40, 43).

Qi, Yufei, Ting Liu et Yuzhuo Fu (2020). « Anomalous Action Recognition Research for Few-shot Learning ». In : *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. T. 1. IEEE, p. 1306-1310 (cf. p. 69, 71, 74).

Rabaud, Vincent et Serge Belongie (2006). « Counting crowded moving objects ». In : *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. T. 1. IEEE, p. 705-711 (cf. p. 20).

Rabiee, Hamidreza et al. (2016a). *Emotion-based crowd representation for abnormality detection*. ArXiv :1607.07646 (cf. p. 49, 55).

Rabiee, Hamidreza et al. (2016b). « Novel dataset for fine-grained abnormal behavior understanding in crowd ». In : *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance*, p. 95-101 (cf. p. 49, 55).

Rakhshani, Hojjat et al. (2020). « Neural Architecture Search for Time Series Classification ». In : *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, p. 1-8 (cf. p. 111).

Ramos, Joelmir et al. (2017). « Visual data mining for crowd anomaly detection using artificial bacteria colony ». In : *Multimedia Tools and Applications*, p. 1-23 (cf. p. 39, 40, 43).

Rand, William M (1971). « Objective criteria for the evaluation of clustering methods ». In : *Journal of the American Statistical association* 66.336, p. 846-850 (cf. p. 36).

Ranjan, Rajnish K et Anupam Agrawal (2016). « Video Summary Based on F-Sift, Tamura Textural and Middle Level Semantic Feature ». In : *Procedia Computer Science* 89, p. 870-876 (cf. p. 41).

Ravanbakhsh, Mahdyar et al. (2016). *Plug-and-play cnn for crowd motion analysis : An application in abnormal event detection*. ArXiv :1610.00307 (cf. p. 39, 43).

Ren, Shaoqing et al. (2015). « Faster r-cnn : Towards real-time object detection with region proposal networks ». In : *Advances in neural information processing systems*, p. 91-99 (cf. p. 22, 33).

Ruchaud, Natacha et Jean-Luc Dugelay (2017). « ASePPI, an adaptive scrambling enabling privacy protection and intelligibility in H. 264/AVC ». In : *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, p. 946-950 (cf. p. 109).

Russakovsky, Olga et al. (2015). « Imagenet large scale visual recognition challenge ». In : *International journal of computer vision* 115.3, p. 211-252 (cf. p. 78).

Russell, Bryan C et al. (2008). « LabelMe : a database and web-based tool for image annotation ». In : *International journal of computer vision* 77.1-3, p. 157-173 (cf. p. 56, 57, 59).

Russell, Patrick (2015). « The Emergence of Smart Cities ». In : *UT School of Architecture. Retrieved from https ://sustainability. utexas. edu/sites/sustainability. utexas. edu/files/EmergenceofSmartCiti es_PatrickRussell. pdf* (cf. p. 5).

Sadeghian, Amir, Alexandre Alahi et Silvio Savarese (2017). *Tracking the untrackable : Learning to track multiple cues with long-term dependencies*. ArXiv :1701.01909 (cf. p. 32).

Sagi, Omer et Lior Rokach (2018). « Ensemble learning : A survey ». In : *Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery* 8.4, e1249 (cf. p. 90).

Sam, Deepak Babu et al. (2019). *Locate, size and count : Accurately resolving people in dense crowds via detection*. ArXiv :1906.07538 (cf. p. 27).

Sam, Deepak Babu et al. (2020). « Locate, size and count : Accurately resolving people in dense crowds via detection ». In : *IEEE transactions on pattern analysis and machine intelligence* (cf. p. 61, 76, 110, 111).

Schröder, Gregory et al. (2018). « Optical Flow Dataset and Benchmark for Visual Crowd Analysis ». In : *IEEE International Conference on Advanced Video and Signal Based Surveillance*, p. 1-6 (cf. p. 46, 55).

Schuldt, Christian, Ivan Laptev et Barbara Caputo (2004). « Recognizing human actions : a local SVM approach ». In : *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* T. 3. IEEE, p. 32-36 (cf. p. 28).

Shahroudy, Amir et al. (2016). « Ntu rgb+ d : A large scale dataset for 3d human activity analysis ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 1010-1019 (cf. p. 78).

Shao, Jie, Nan Dong et Qian Zhao (2018). « A Real-Time Algorithm for Small Group Detection in Medium Density Crowds ». In : *Pattern Recognition and Image Analysis* 28.2, p. 282-287 (cf. p. 22, 23).

Shao, Jing, Chen Change Loy et Xiaogang Wang (2014). « Scene-independent group profiling in crowd ». In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 2219-2226 (cf. p. 22, 36, 43, 50, 55).

Shao, Jing, Chen Change Loy et Xiaogang Wang (2017). « Learning scene-independent group descriptors for crowd understanding ». In : *IEEE transactions on circuits and systems for video technology* 27.6, p. 1290-1303 (cf. p. 23, 50).

Shao, Jing et al. (2015). « Deeply learned attributes for crowded scene understanding ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 4657-4666 (cf. p. 24, 25).

Sharma, Sharad et al. (2019). « Artificial intelligence agents for crowd simulation in an immersive environment for emergency response ». In : *Electronic Imaging* 2019.2, p. 176-1 (cf. p. 111).

Shendarkar, Ameya et al. (2006). « Crowd simulation for emergency response using BDI agent based on virtual reality ». In : *Proceedings of the 2006 winter simulation conference*. IEEE, p. 545-553 (cf. p. 111).

Shi, Jianbo et Jitendra Malik (1998). « Motion segmentation and tracking using normalized cuts ». In : *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*. IEEE, p. 1154-1160 (cf. p. 24).

Shu, Tianmin, Sinisa Todorovic et Song-Chun Zhu (2017). « CERN : Confidence-Energy Recurrent Network for Group Activity Recognition ». In : *2017 IEEE Conference on Computer Vision and Pattern Recognition* (cf. p. 37, 43).

Simonyan, Karen et Andrew Zisserman (2014). *Very deep convolutional networks for large-scale image recognition*. ArXiv :1409.1556 (cf. p. 40).

Sindagi, Vishwanath A et Vishal M Patel (2017). « Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting ». In : *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*, p. 1-6 (cf. p. 25, 26, 43).

— (2018). « A survey of recent advances in cnn-based single image crowd counting and density estimation ». In : *Pattern Recognition Letters* 107, p. 3-16 (cf. p. 24).

Singh, Amarjot et Nick Kingsbury (2017). « Dual-tree wavelet scattering network with parametric log transformation for object classification ». In : *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, p. 2622-2626 (cf. p. 39).

Singh, Amarjot, Devendra Patil et SN Omkar (2018). *Eye in the Sky : Real-time Drone Surveillance System (DSS) for Violent Individuals Identification using ScatterNet Hybrid Deep Learning Network*. ArXiv :1806.00746 (cf. p. 39, 43, 48, 53, 55).

Singh, Kuldeep et al. (2020). « Crowd anomaly detection using aggregation of ensembles of fine-tuned ConvNets ». In : *Neurocomputing* 371, p. 188-198 (cf. p. 40, 43).

Siva, Parthipan et Tao Xiang (2010). « Action detection in crowd. » In : *BMVC*, p. 1-11 (cf. p. 27, 43).

Smeaton, Alan F, Paul Over et Wessel Kraaij (2006). « Evaluation campaigns and TRECVid ». In : *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, p. 321-330 (cf. p. 90).

Solera, Francesco, Simone Calderara et Rita Cucchiara (2016). « Socially constrained structural learning for groups detection in crowd ». In : *IEEE transactions on pattern analysis and machine intelligence* 38.5, p. 995-1008 (cf. p. 22).

Solmaz, Berkan, Brian E Moore et Mubarak Shah (2012). « Identifying behaviors in crowd scenes using stability analysis for dynamical systems ». In : *IEEE transactions on pattern analysis and machine intelligence* 34.10, p. 2064-2070 (cf. p. 31, 34).

Song, Xuan et al. (2013). « An online system for multiple interacting targets tracking : Fusion of laser and vision, tracking and learning ». In : *ACM Transactions on Intelligent Systems and Technology (TIST)* 4.1, p. 18 (cf. p. 16).

Soomro, Khurram, Amir Roshan Zamir et Mubarak Shah (2012). *UCF101 : A dataset of 101 human actions classes from videos in the wild*. ArXiv :1212.0402 (cf. p. 47, 53, 55).

Storn, Rainer et Kenneth Price (1997). « Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces ». In : *Journal of global optimization* 11.4, p. 341-359 (cf. p. 99).

Sugimura, Daisuke et al. (2009). « Using individuality to track individuals : Clustering individual trajectories in crowds using local appearance and frequency trait ». In : *2009 IEEE 12th International Conference on Computer Vision*. IEEE, p. 1467-1474 (cf. p. 20).

Sultani, Waqas, Chen Chen et Mubarak Shah (2018). « Real-world anomaly detection in surveillance videos ». In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 6479-6488 (cf. p. 41, 49, 55, 68-70, 72-74).

Sun, Li et al. (2020). « Discriminative Clip Mining for Video Anomaly Detection ». In : *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, p. 2121-2125 (cf. p. 69, 70, 74).

Szegedy, Christian et al. (2015). « Going deeper with convolutions ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 1-9 (cf. p. 90).

Tang, Yongyi et al. (2018). *Long-Term Human Motion Prediction by Modeling Motion Context and Enhancing Motion Dynamic*. ArXiv :1805.02513 (cf. p. 36).

Thida, Myo et al. (2013). « A literature review on video analytics of crowded scenes ». In : *Intelligent multimedia surveillance*. Springer, p. 17-36 (cf. p. 6, 10-12, 18, 23).

Tian, Yonglong et al. (2015). « Deep learning strong parts for pedestrian detection ». In : *Proceedings of the IEEE international conference on computer vision*, p. 1904-1912 (cf. p. 21, 23).

Tomasi, Carlo et Takeo Kanade (1991). *Detection and Tracking of Point Features*. Rapp. tech. International Journal of Computer Vision (cf. p. 22, 33, 36).

Tran, Du et al. (2015). « Learning spatiotemporal features with 3d convolutional networks ». In : *Proceedings of the IEEE international conference on computer vision*, p. 4489-4497 (cf. p. 61, 63-65, 82, 89, 92).

Tran, Du et al. (2018). « A Closer Look at Spatiotemporal Convolutions for Action Recognition ». In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 6450-6459 (cf. p. 28, 43).

TRB (2000). « Highway capacity manual ». In : *Transportation Research Board, National Research Council, Washington, DC* (cf. p. 7).

Tripathi, G., K. Singh et D.K. Vishwakarma (2018). « Convolutional neural networks for crowd behaviour analysis : a survey ». In : *The Visual Computer*, p. 1-24 (cf. p. 8-10, 19, 27).

UCSD (2013). *UCSD anomaly dataset*. URL : `http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm` (cf. p. 55).

Uijlings, Jasper RR et al. (2013). « Selective search for object recognition ». In : *International journal of computer vision* 104.2, p. 154-171 (cf. p. 21).

Ullah, Amin et al. (2020). « One-Shot Learning for Surveillance Anomaly Recognition using Siamese 3D CNN ». In : *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, p. 1-8 (cf. p. 69, 72).

Ullah, Habib et Nicola Conci (2012). « Crowd motion segmentation and anomaly detection via multi-label optimization ». In : *ICPR workshop on pattern recognition and crowd analysis*. T. 75 (cf. p. 31).

Ullah, Habib et al. (2019). « Two stream model for crowd video classification ». In : *2019 8th European Workshop on Visual Information Processing (EUVIP)*. IEEE, p. 93-98 (cf. p. 31, 43).

Ullah, Mohib et al. (2016). « Crowd behavior identification ». In : *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, p. 1195-1199 (cf. p. 30, 31, 43).

Vahora, SA et NC Chauhan (2018). « Deep neural network model for group activity recognition using contextual relationship ». In : *Engineering Science and Technology, an International Journal* (cf. p. 36, 43).

Vascon, Sebastiano et Loris Bazzani (2017). « Group Detection and Tracking Using Sociological Features ». In : *Group and Crowd Behavior for Computer Vision*. Elsevier, p. 29-66 (cf. p. 22).

Venugopalan, Subhashini et al. (2015). « Sequence to sequence-video to text ». In : *Proceedings of the IEEE international conference on computer vision*, p. 4534-4542 (cf. p. 30).

Vishwakarma, Sarvesh et Anupam Agrawal (2013). « A survey on activity recognition and behavior understanding in video surveillance ». In : *The Visual Computer* 29.10, p. 983-1009 (cf. p. 18).

Vondrick, Carl, Donald Patterson et Deva Ramanan (2013). « Efficiently scaling up crowdsourced video annotation ». In : *International journal of computer vision* 101.1, p. 184-204 (cf. p. 71).

Voon, Wong Pei et al. (2019). « Collective Interaction Filtering Approach for Detection of Group in Diverse Crowded Scenes. » In : *TIIS* 13.2, p. 912-928 (cf. p. 22, 23).

WALACH, Elad et Lior WOLF (2016). « Learning to count with cnn boosting ». In : *European conference on computer vision*. Springer, p. 660-676 (cf. p. 25, 90, 91).

WALIA, Gurjit Singh et Rajiv KAPOOR (2016). « Recent advances on multicue object tracking : a survey ». In : *Artificial Intelligence Review* 46.1, p. 1-39 (cf. p. 5, 15, 16).

WAN, Jia et Antoni CHAN (2019). « Adaptive density map generation for crowd counting ». In : *Proceedings of the IEEE International Conference on Computer Vision*, p. 1130-1139 (cf. p. 26, 43).

WAN, Jia, Nikil Senthil KUMAR et Antoni B CHAN (2020). *Fine-Grained Crowd Counting*. ArXiv :2007.06146 (cf. p. 27).

WANG, He et Carol O'SULLIVAN (2016). « Globally continuous and non-Markovian crowd activity analysis from videos ». In : *European Conference on Computer Vision*. Springer, p. 527-544 (cf. p. 29, 43).

WANG, Heng et al. (2011a). « Action recognition by dense trajectories ». In : *CVPR 2011*. IEEE, p. 3169-3176 (cf. p. 31).

WANG, Liming et al. (2007). « Object detection combining recognition and segmentation ». In : *Asian conference on computer vision*, p. 189-199 (cf. p. 20).

WANG, Pichao et al. (2016). « Action recognition based on joint trajectory maps using convolutional neural networks ». In : *Proceedings of the 24th ACM international conference on Multimedia*, p. 102-106 (cf. p. 78).

WANG, Qi et al. (2019). « Learning from synthetic data for crowd counting in the wild ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 8198-8207 (cf. p. 25, 26, 43, 44, 55).

WANG, Qi et al. (2020). « NWPU-crowd : A large-scale benchmark for crowd counting and localization ». In : *IEEE transactions on pattern analysis and machine intelligence* 43.6, p. 2141-2149 (cf. p. 78).

WANG, Weiyue et al. (2014). « Finding coherent motions and semantic regions in crowd scenes : A diffusion and clustering approach ». In : *European Conference on Computer Vision*. Springer, p. 756-771 (cf. p. 31).

WANG, Xiaogang, Kinh TIEU et Eric GRIMSON (2006). « Learning semantic scene models by trajectory analysis ». In : *European conference on computer vision*. Springer, p. 110-123 (cf. p. 20).

WANG, Xiaogang et al. (2011b). « Trajectory analysis and semantic region modeling using nonparametric hierarchical bayesian models ». In : *International journal of computer vision* 95.3, p. 287-312 (cf. p. 30).

WANG, Xiaoyu et al. (2013). « Regionlets for generic object detection ». In : *Proceedings of the IEEE international conference on computer vision*, p. 17-24 (cf. p. 21).

WANG, Yi et al. (2021). « A self-training approach for point-supervised object detection and counting in crowds ». In : *IEEE Transactions on Image Processing* 30, p. 2876-2887 (cf. p. 77, 87).

WANG, Zhen, Cheng CHENG et Xuzhi WANG (2018). « A Fast Crowd Segmentation Method ». In : *2018 International Conference on Audio, Language and Image Processing (ICALIP)*, p. 242-245 (cf. p. 18, 22, 23).

WEI, Xinlei et al. (2020). « A very deep two-stream network for crowd type recognition ». In : *Neurocomputing* 396, p. 522-533 (cf. p. 29, 43).

WICHARD, Joerg D (2016). « An adaptive forecasting strategy with hybrid ensemble models ». In : *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, p. 1495-1498 (cf. p. 99).

Wojke, Nicolai, Alex Bewley et Dietrich Paulus (2017). « Simple online and realtime tracking with a deep association metric ». In : *Image Processing (ICIP), 2017 IEEE International Conference on*. IEEE, p. 3645-3649 (cf. p. 33, 35, 43).

Wu, Chuting et al. (2017a). « Ensemble learning for crowd flows prediction on campus ». In : *International Conference on Smart Computing and Communication*. Springer, p. 103-113 (cf. p. 90, 91).

Wu, Mingrui et Bernhard Schölkopf (2007). « A local learning approach for clustering ». In : *Advances in neural information processing systems*, p. 1529-1536 (cf. p. 36).

Wu, Shuang et al. (2017b). « Crowd behavior analysis via curl and divergence of motion trajectories ». In : *International Journal of Computer Vision* 123.3, p. 499-519 (cf. p. 34, 35, 43).

Xie, Shaoci, Xiaohong Zhang et Jing Cai (2019). « Video crowd detection and abnormal behavior model detection based on machine learning method ». In : *Neural Computing and Applications* 31.1, p. 175-184 (cf. p. 41, 43).

Yamaguchi, Kota et al. (2011). « Who are you with and where are you going ? » In : *CVPR 2011*. IEEE, p. 1345-1352 (cf. p. 35).

Yan, Liqi, Mingjian Zhu et Changbin Yu (2019). *Crowd Video Captioning*. ArXiv :1911.05449 (cf. p. 30, 31, 43).

Yang, Shuo et al. (2016). « Wider face : A face detection benchmark ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 5525-5533 (cf. p. 76, 77).

You, Quanzeng et Hao Jiang (2018). *Action4D : Real-time Action Recognition in the Crowd and Clutter*. ArXiv :1806.02424 (cf. p. 28, 43).

Yu, Chun-Nam John et Thorsten Joachims (2009). « Learning structural svms with latent variables ». In : *Proceedings of the 26th annual international conference on machine learning*, p. 1169-1176 (cf. p. 91).

Yuan, Yuan, Yuwei Lu et Qi Wang (2017). « Tracking as a whole : Multi-target tracking by modeling group behavior with sequential detection ». In : *IEEE Transactions on Intelligent Transportation Systems* 18.12, p. 3339-3349 (cf. p. 22).

Zach, Christopher, Thomas Pock et Horst Bischof (2007). « A duality based approach for realtime TV-L 1 optical flow ». In : *Joint pattern recognition symposium*. Springer, p. 214-223 (cf. p. 65, 95).

Zaheer, Muhammad Zaigham et al. (2020). « A Self-Reasoning Framework for Anomaly Detection Using Video-Level Labels ». In : *IEEE Signal Processing Letters* 27, p. 1705-1709 (cf. p. 69, 72).

Zhan, Beibei et al. (2008). « Crowd analysis : A survey ». In : *Machine Vision and Applications* 19.5-6, p. 345-357 (cf. p. 5, 7, 8, 10, 18, 23, 38).

Zhang, Cong et al. (2015). « Cross-scene crowd counting via deep convolutional neural networks ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 833-841 (cf. p. 63, 76).

Zhang, Jiangong, Laiyun Qing et Jun Miao (2019). « Temporal convolutional network with complementary inner bag loss for weakly supervised anomaly detection ». In : *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, p. 4030-4034 (cf. p. 72).

Zhang, Jianguang, Yanbin Liu et Jianmin Jiang (2018). « Tensor learning and automated rank selection for regression-based video classification ». In : *Multimedia Tools and Applications* 77.22, p. 29213-29230 (cf. p. 31).

Zhang, Liliang et al. (2016a). « Is faster R-CNN doing well for pedestrian detection ? » In : *European Conference on Computer Vision*, p. 443-457 (cf. p. 22).

Zhang, Xuguang et al. (2018). « Energy level-based abnormal crowd behavior detection ». In : *Sensors* 18.2, p. 423 (cf. p. 31).

Zhang, Yingying et al. (2016b). « Single-image crowd counting via multi-column convolutional neural network ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 589-597 (cf. p. 24-26).

— (2016c). « Single-image crowd counting via multi-column convolutional neural network ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 589-597 (cf. p. 70, 72, 76-78, 88).

Zheng, Liang et al. (2016). « Mars : A video benchmark for large-scale person re-identification ». In : *European Conference on Computer Vision*, p. 868-884 (cf. p. 33).

Zhong, Jia-Xing et al. (2019). « Graph convolutional label noise cleaner : Train a plug-and-play action classifier for anomaly detection ». In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 1237-1246 (cf. p. 69, 70, 72-74).

Zhou, Bolei, Xiaoou Tang et Xiaogang Wang (2012). « Coherent filtering : Detecting coherent motions from crowd clutters ». In : *European Conference on Computer Vision*. Springer, p. 857-871 (cf. p. 23, 36, 37).

Zhou, Lijuan et al. (2014). « Discriminative key pose extraction using extended lc-ksvd for action recognition ». In : *2014 International Conference on Digital Image Computing : Techniques and Applications (DICTA)*. IEEE, p. 1-8 (cf. p. 78).

Zhou, Zhi-Hua (2009). « Ensemble Learning. » In : *Encyclopedia of biometrics* 1, p. 270-273 (cf. p. 89-91).

Zhou, Zhi-Hua, Jianxin Wu et Wei Tang (2002). « Ensembling neural networks : many could be better than all ». In : *Artificial intelligence* 137.1-2, p. 239-263 (cf. p. 91).

Zhu, Jun-Yan et al. (2017). « Unpaired image-to-image translation using cycle-consistent adversarial networks ». In : *Proceedings of the IEEE international conference on computer vision*, p. 2223-2232 (cf. p. 26).

Zhu, Yi et Shawn Newsam (2019). « Motion-aware feature for improved video anomaly detection ». In : *arXiv preprint arXiv :1907.10211* (cf. p. 70, 71).

Zitnick, C Lawrence et Piotr Dollár (2014). « Edge boxes : Locating object proposals from edges ». In : *European conference on computer vision*, p. 391-405 (cf. p. 20).

Zitouni, M Sami, Harish Bhaskar et Mohammed E Al-Mualla (2016). « Robust Background Modeling and Foreground Detection using Dynamic Textures. » In : *VISIGRAPP (4 : VISAPP)*, p. 403-410 (cf. p. 37).

Zitouni, M Sami, Andrzej Sluzek et Harish Bhaskar (2020). « Towards understanding socio-cognitive behaviors of crowds from visual surveillance data ». In : *Multimedia Tools and Applications* 79.3, p. 1781-1799 (cf. p. 37, 43).

Annexe A

Matrices de confusion de l'apport de la détection de têtes pour la classification des scènes de foule

Dans cette annexe, nous incluons les matrices de confusion de chaque modèle de Réseau de Neurones Convolutifs et du Perceptron Multicouche qui le complète.

A.0.1 Matrices de confusion des Réseaux de Neurones Convolutifs

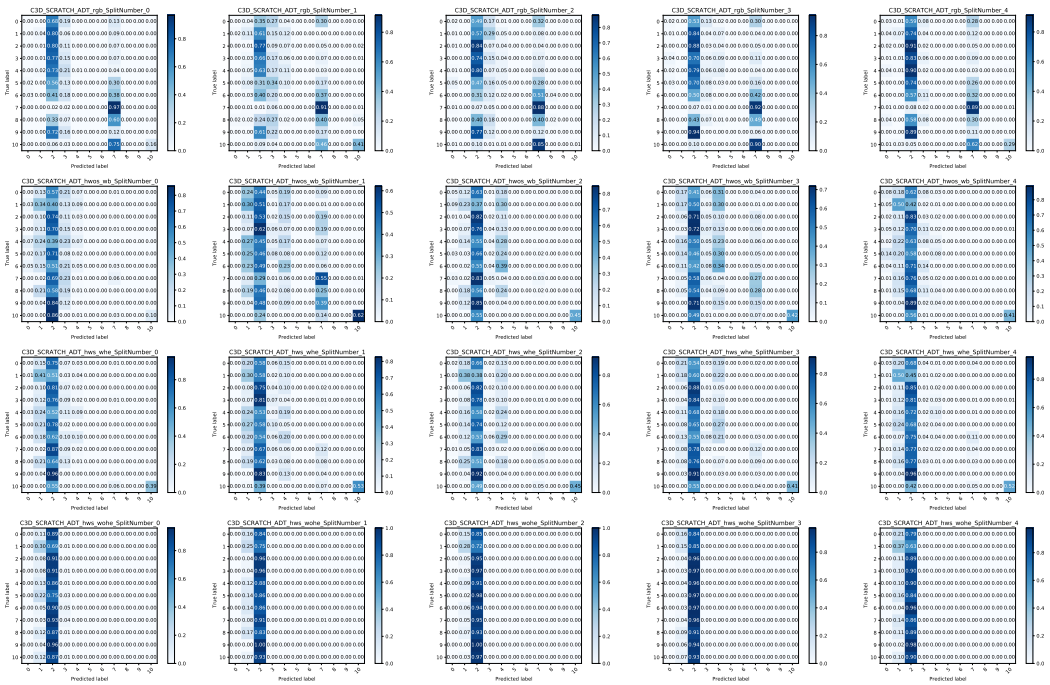


FIGURE A.1 – Matrices de confusion des modèles C3D entraînés de zéro

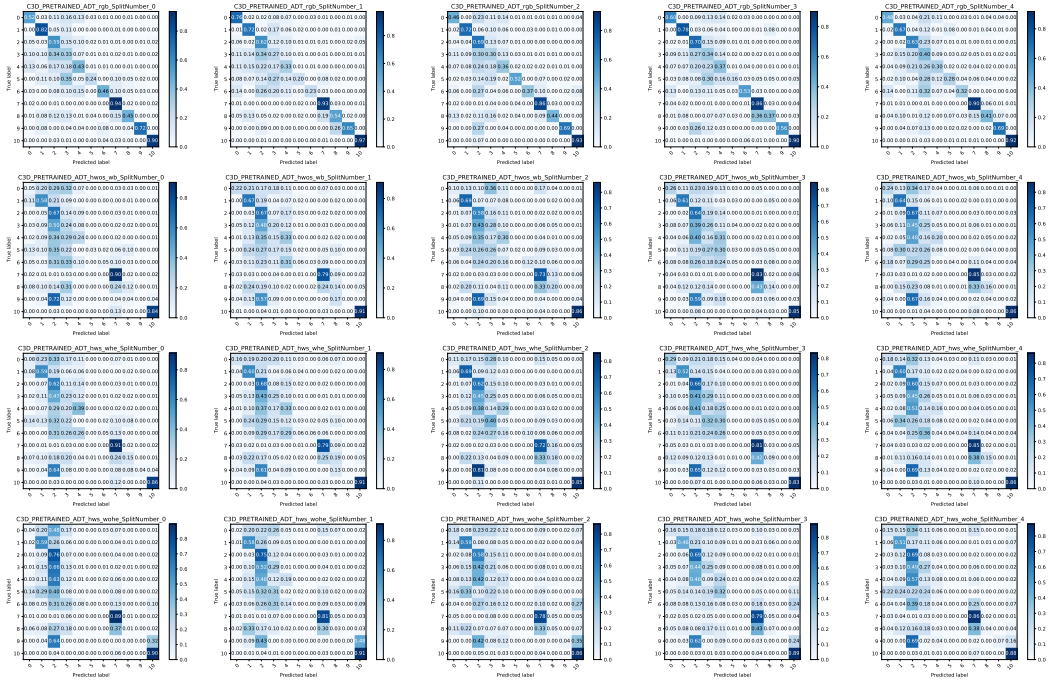


FIGURE A.2 – Matrices de confusion des modèles C3D ajustés

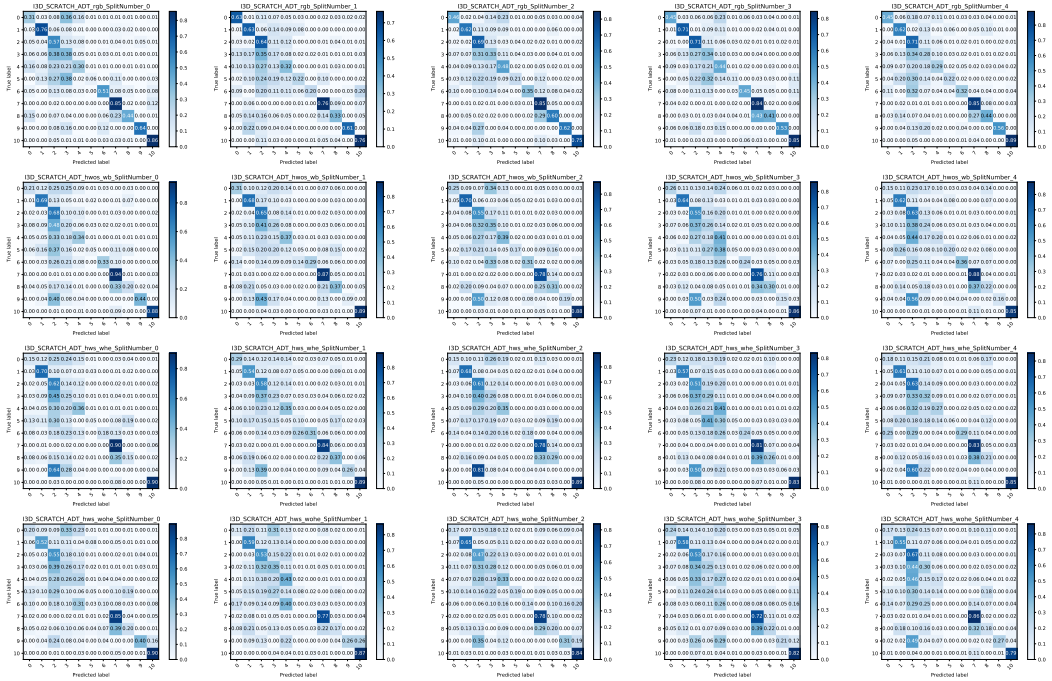


FIGURE A.3 – Matrices de confusion des modèles I3D entraînés de zéro

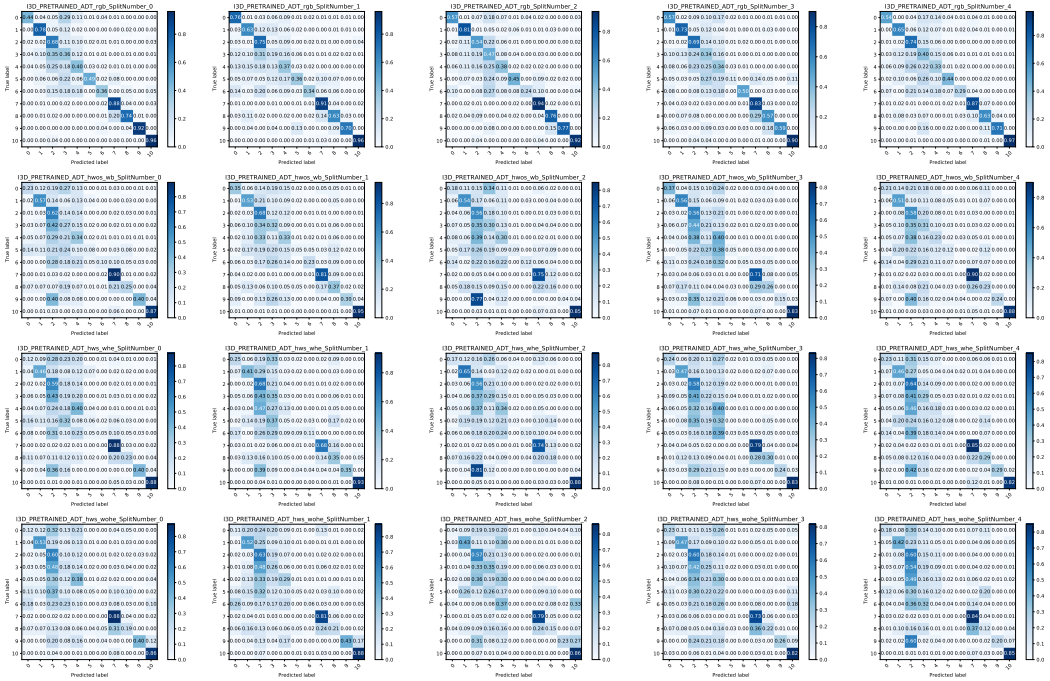


FIGURE A.4 – Matrices de confusion des modèles I3D pré-entrainés

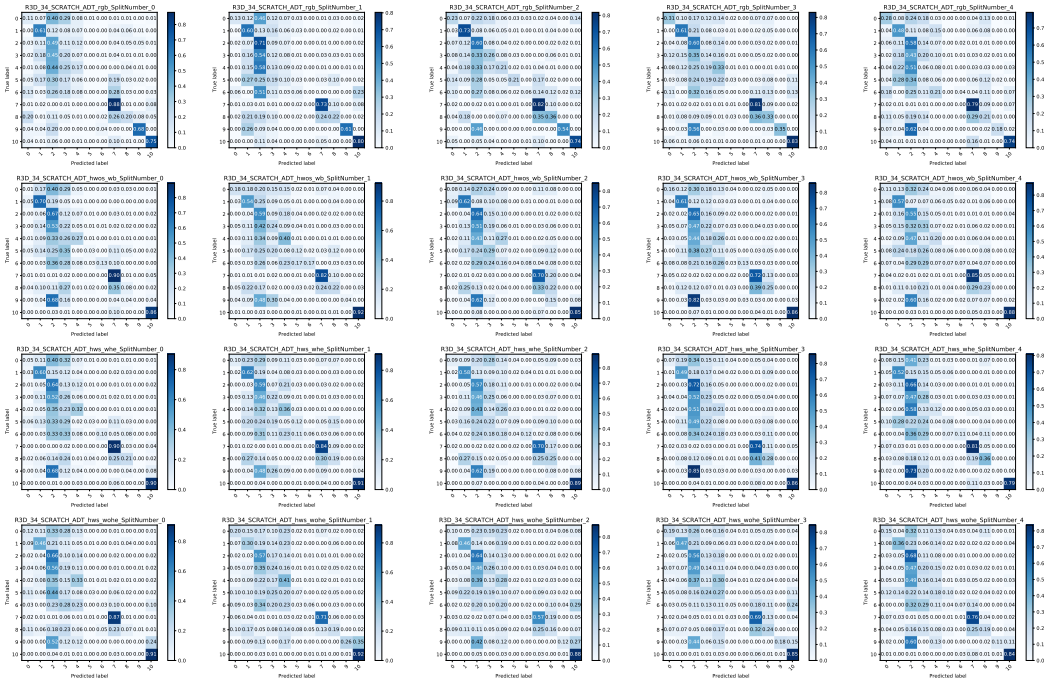


FIGURE A.5 – Matrices de confusion des modèles R3D (34 couches) entraînés de zéro

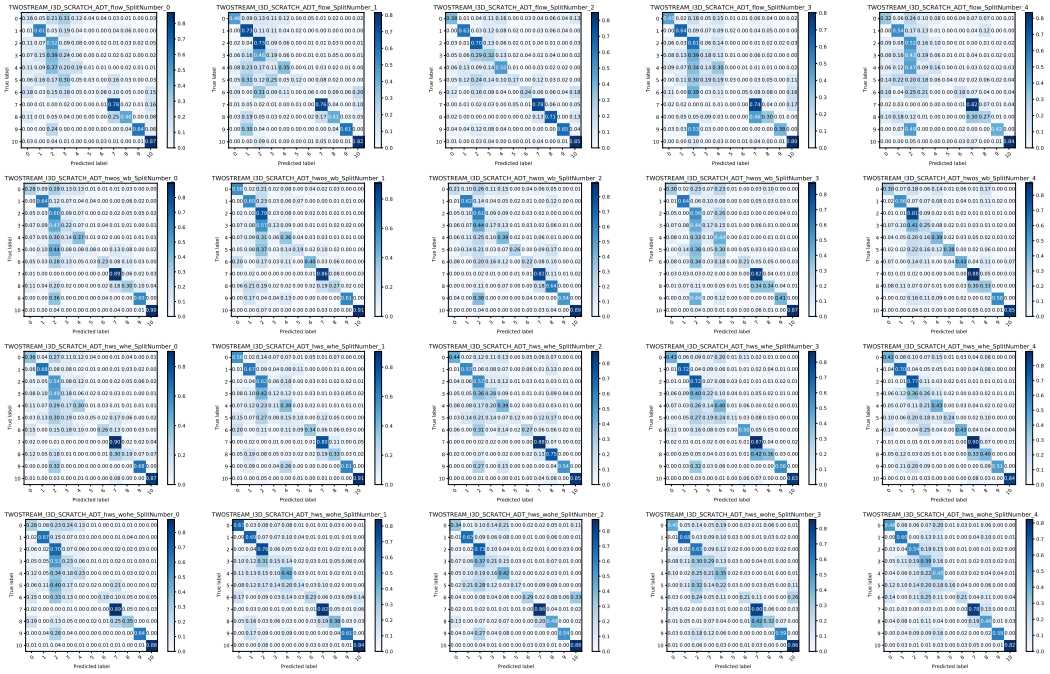


FIGURE A.6 – Matrices de confusion des modèles TwoStream I3D entraînés de zéro

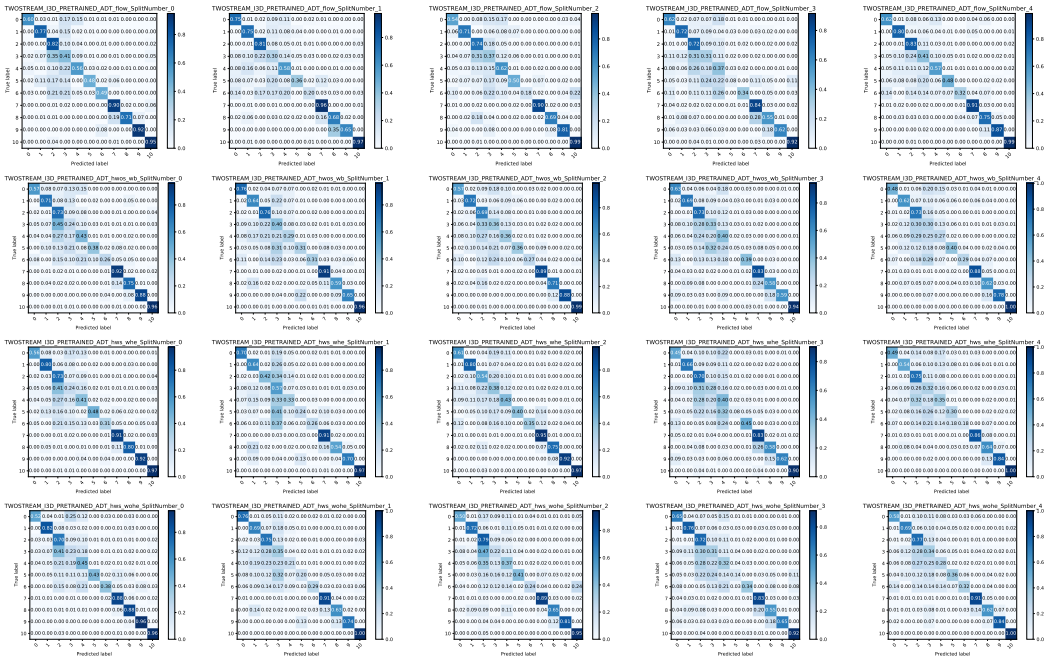


FIGURE A.7 – Matrices de confusion des modèles TwoStream I3D pré-entraînés

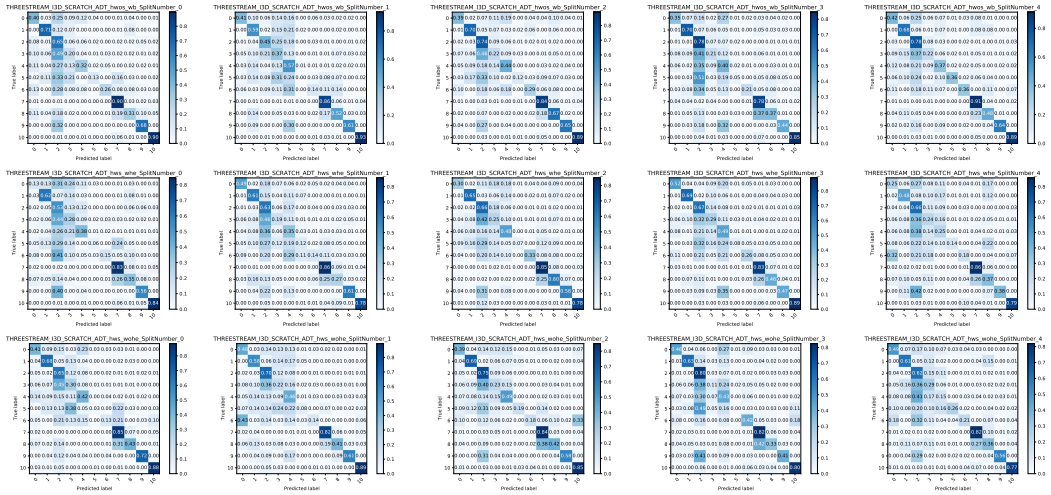


FIGURE A.8 – Matrices de confusion des modèles ThreeStream I3D entraînés de zéro

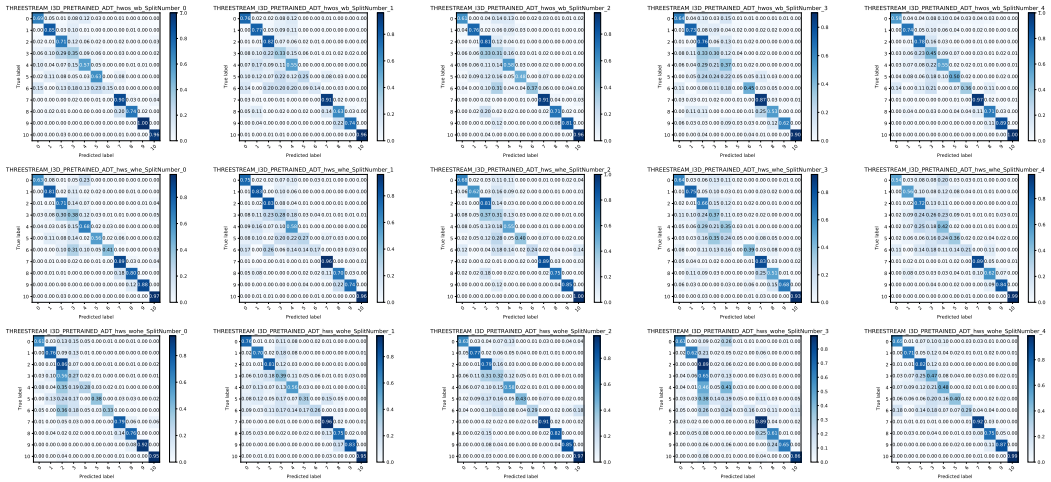


FIGURE A.9 – Matrices de confusion des modèles ThreeStream I3D pré-entrainés

A.0.2 Matrices de confusion des Perceptrons Multicouches

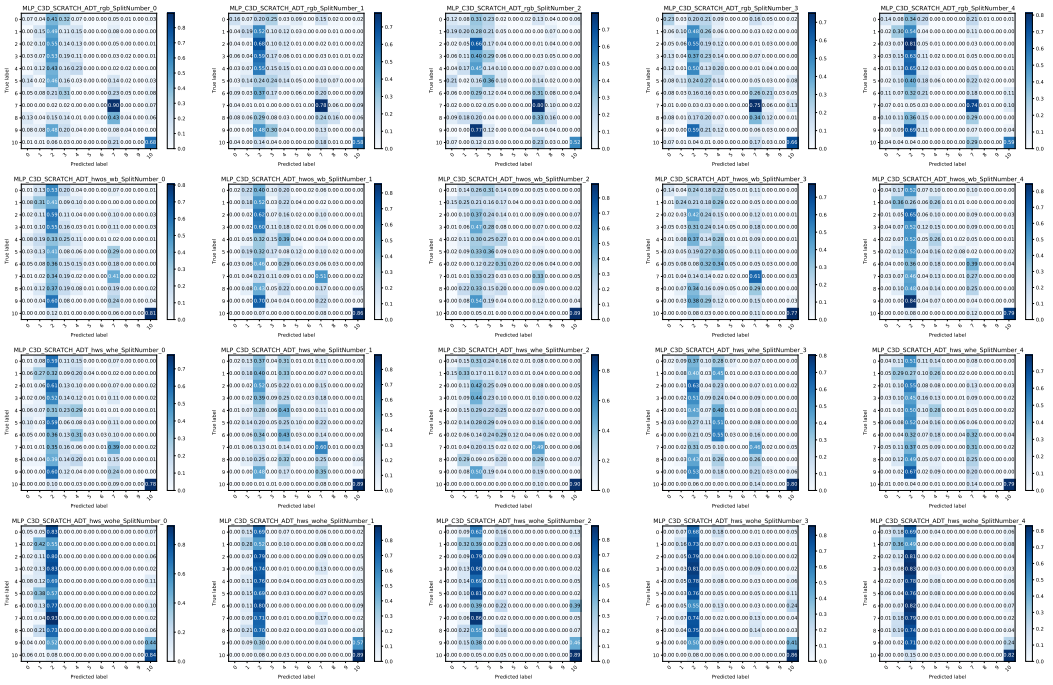


FIGURE A.10 – Matrices de confusion des perceptrons multicouches liés aux modèles C3D entraînés de zéro

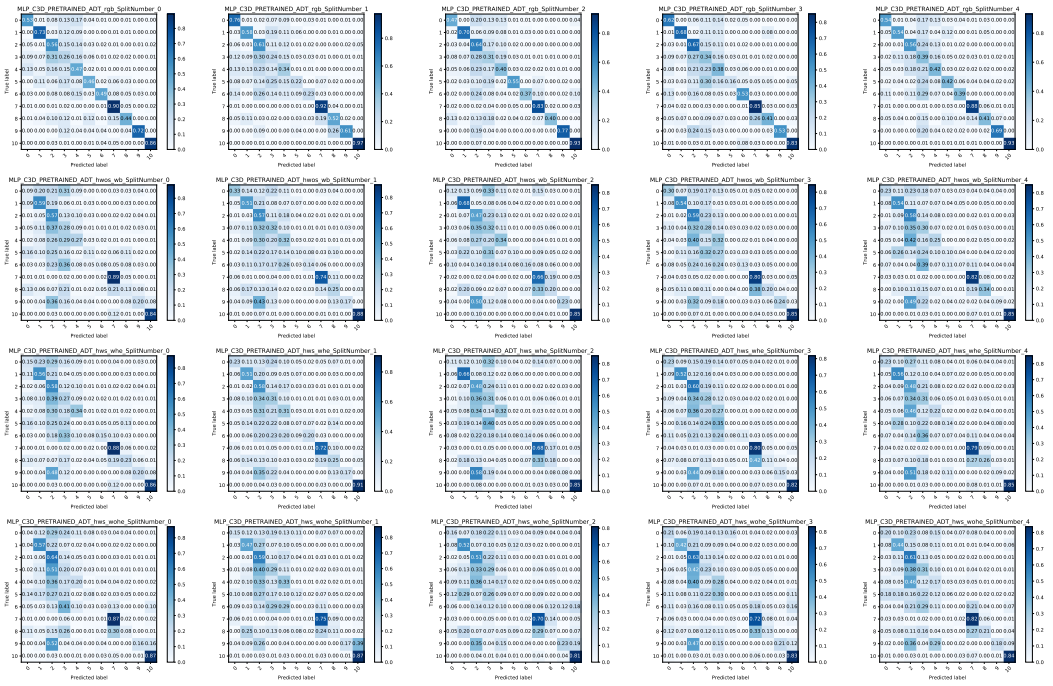


FIGURE A.11 – Matrices de confusion des perceptrons multicouches liés aux modèles C3D ajustés

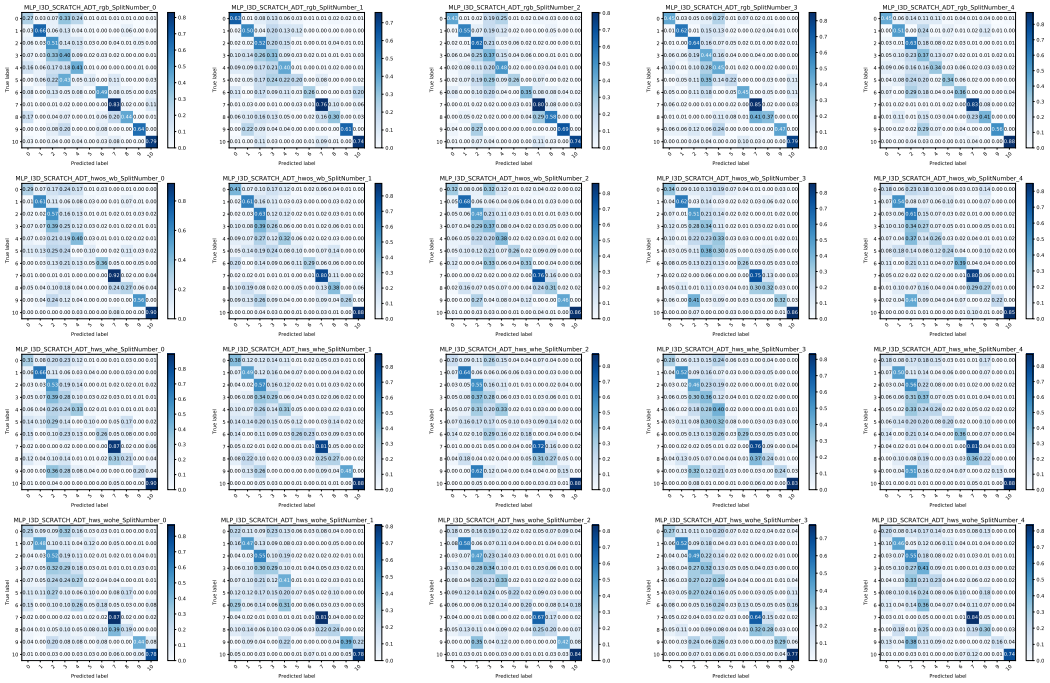


FIGURE A.12 – Matrices de confusion des perceptrons multicouches liés aux modèles I3D entraînés de zéro

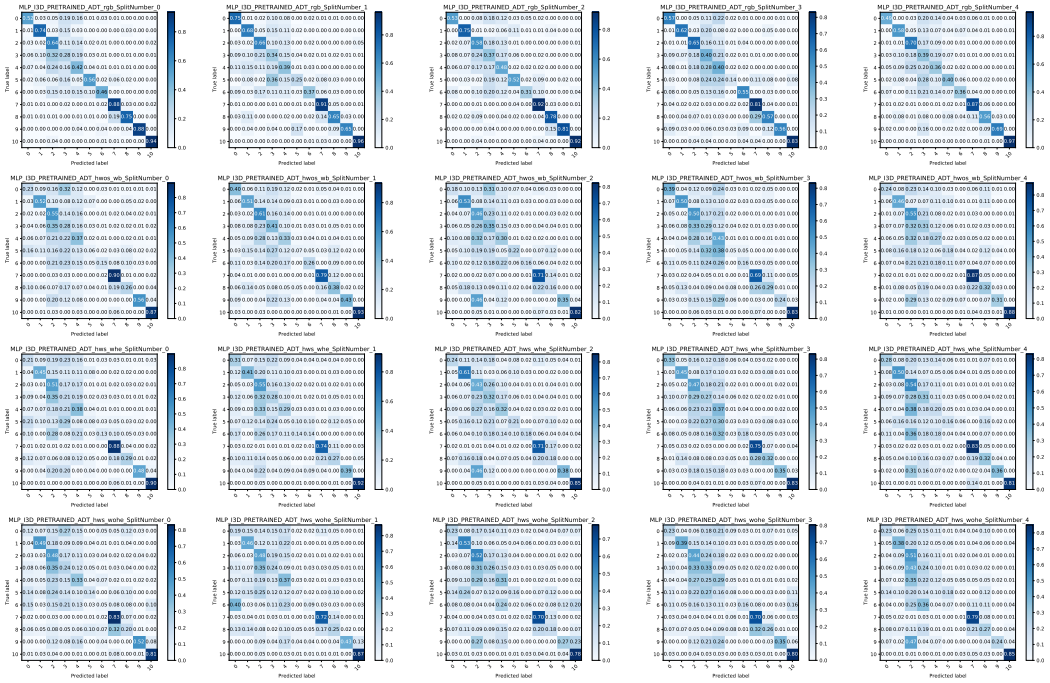


FIGURE A.13 – Matrices de confusion des perceptrons multicouches liés aux modèles I3D pré-entrainés

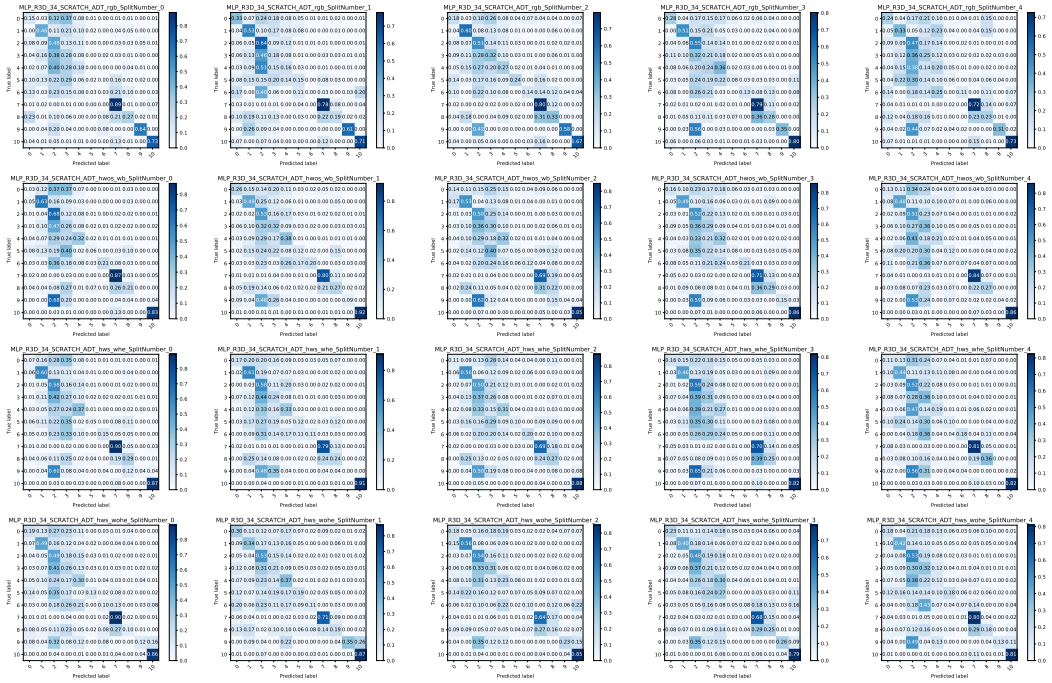


FIGURE A.14 – Matrices de confusion des perceptrons multicouches liés aux modèles R3D (34 couches) entraînés de zéro

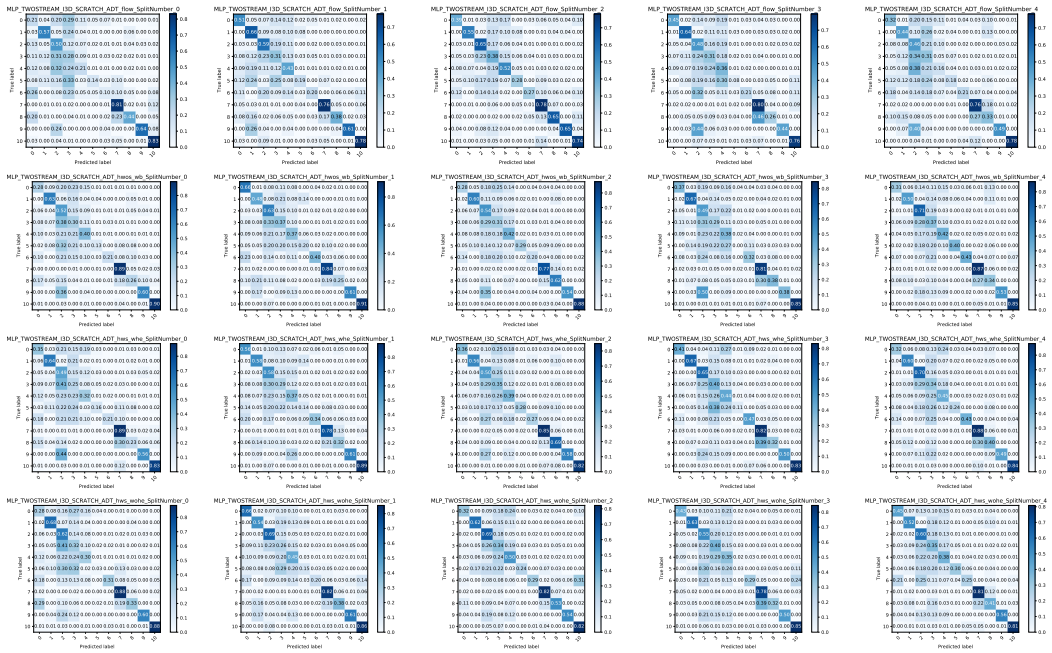


FIGURE A.15 – Matrices de confusion des perceptrons multicouches liés aux modèles TwoStream I3D entraînés de zéro

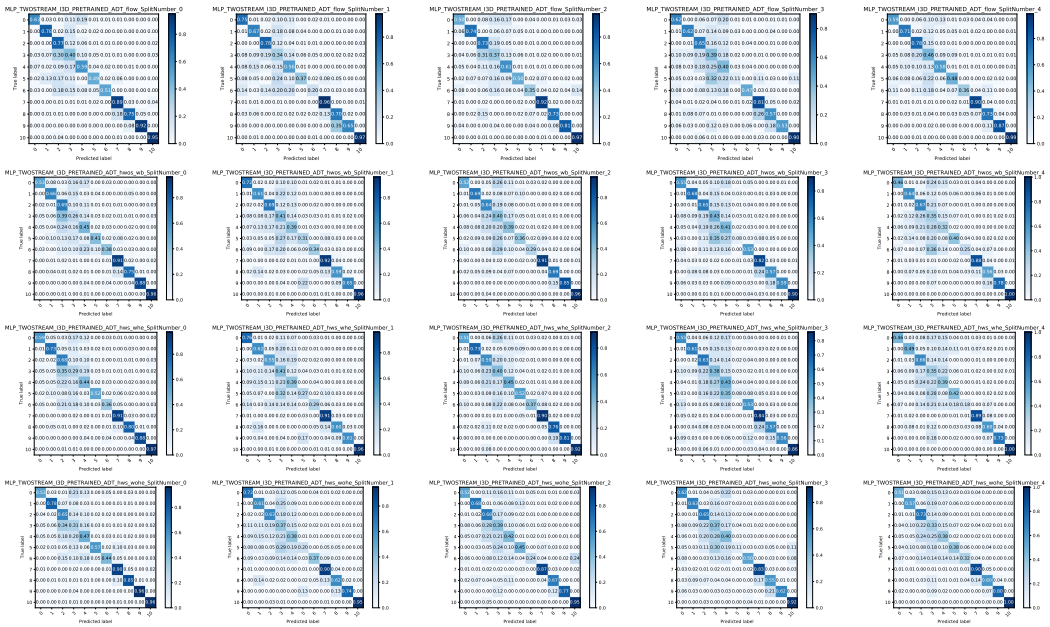


FIGURE A.16 – Matrices de confusion des perceptrons multicouches liés aux modèles TwoStream I3D pré-entraînés

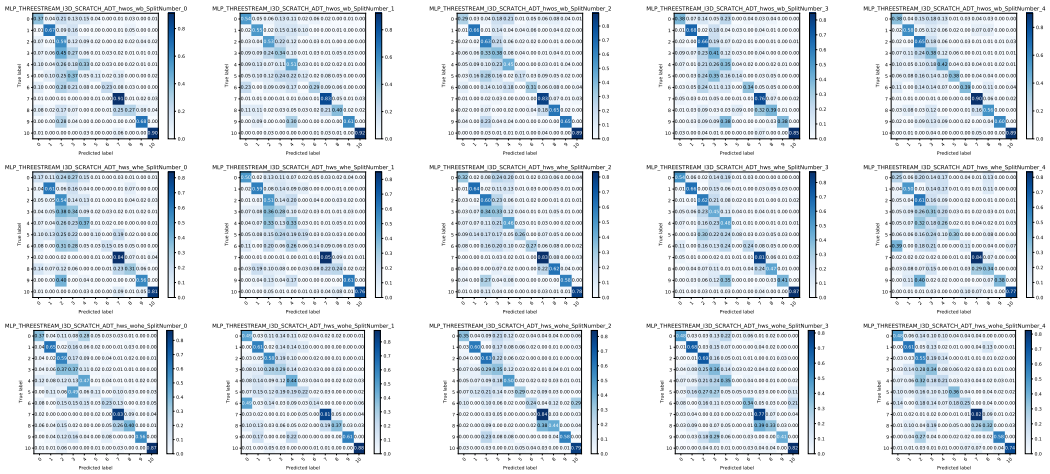


FIGURE A.17 – Matrices de confusion des perceptrons multicouches liés aux modèles ThreeStream I3D entraînés de zéro

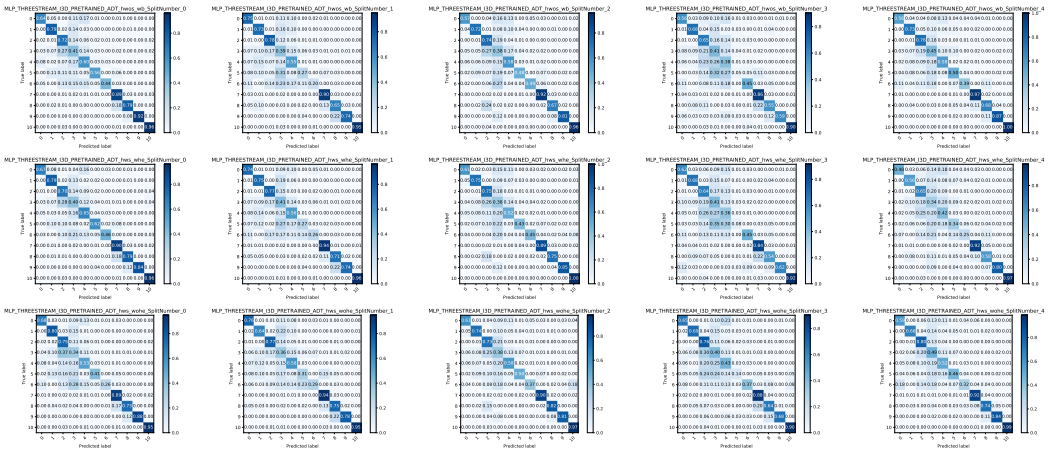
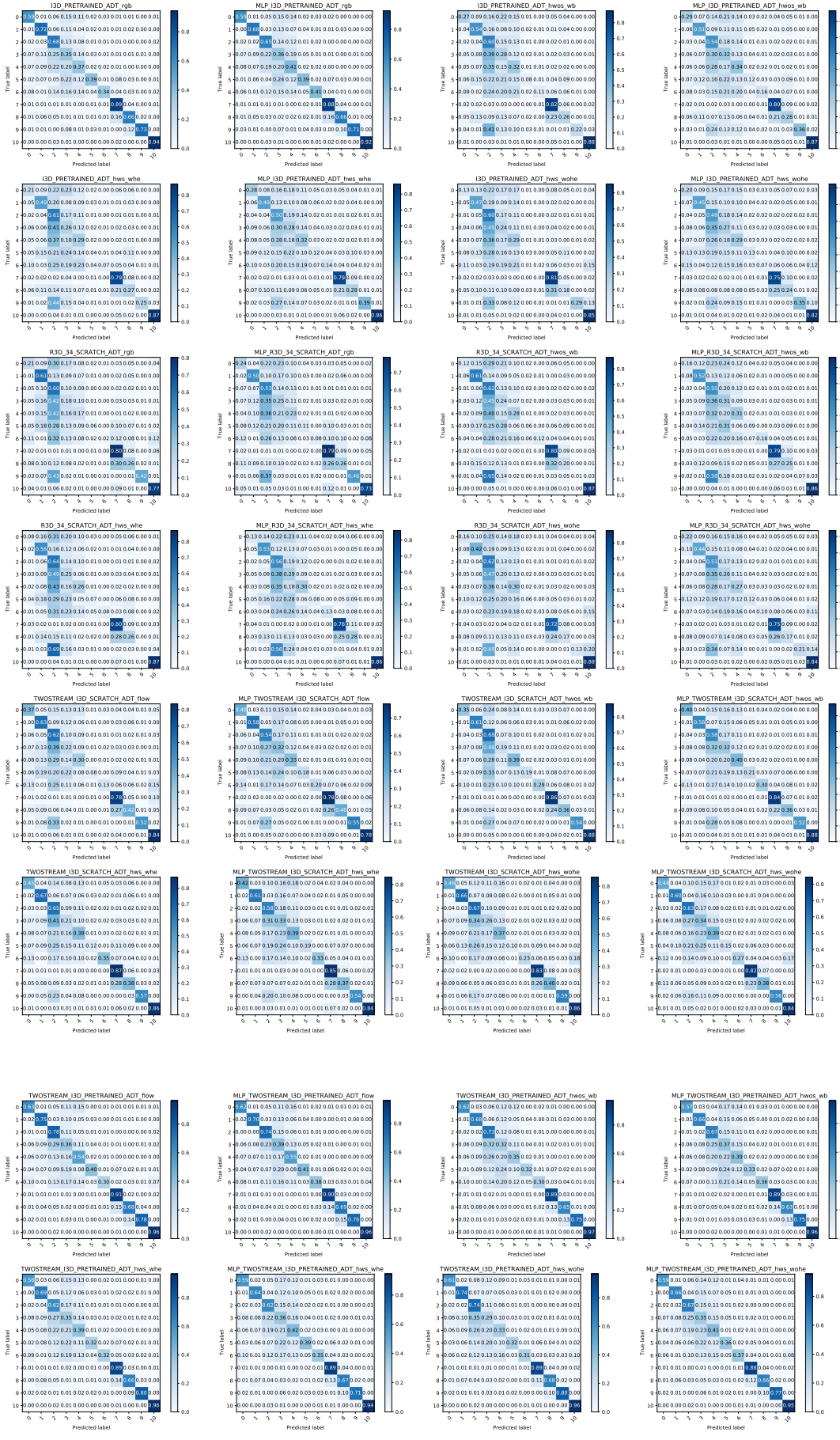
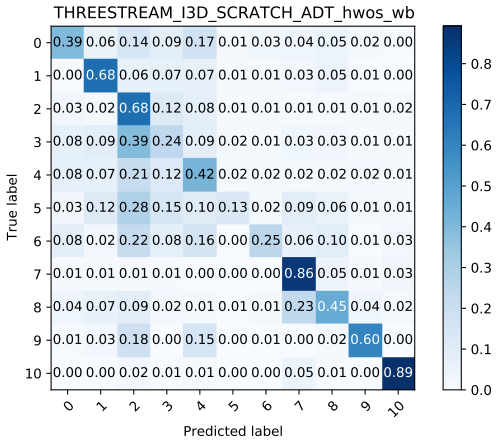
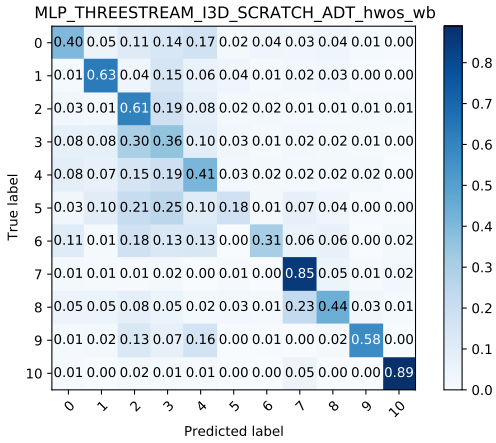


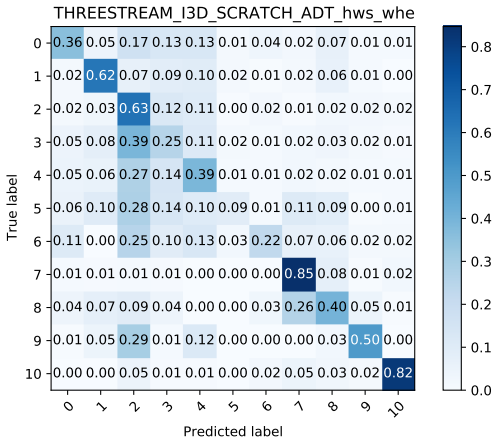
FIGURE A.18 – Matrices de confusion des perceptrons multicouches liés aux modèles ThreeStream I3D pré-entrainés

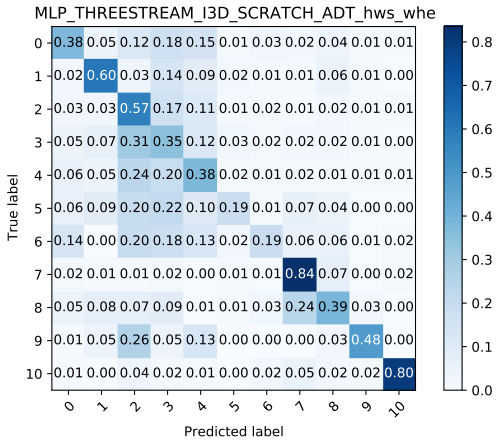
A.0.3 Matrices de confusion globales des RNC et des RNC avec la surcouche PMC

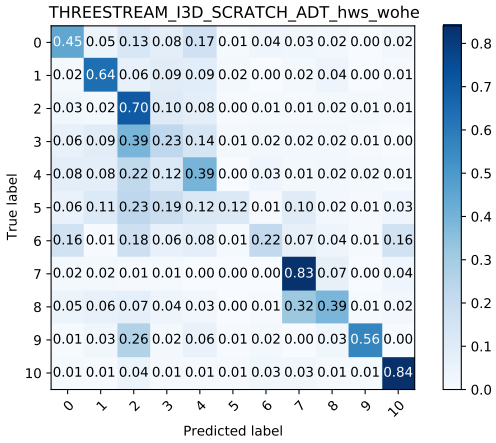

I3D_PRETRAINED_ADT_rgb
MLP_I3D_PRETRAINED_ADT_rgb
I3D_PRETRAINED_ADT_hwos_wb
MLP_I3D_PRETRAINED_ADT_hwos_wb
I3D_PRETRAINED_ADT_hws_whe
MLP_I3D_PRETRAINED_ADT_hws_whe
I3D_PRETRAINED_ADT_hws_wohe
MLP_I3D_PRETRAINED_ADT_hws_wohe
R3D_34_SCRATCH_ADT_rgb
MLP_R3D_34_SCRATCH_ADT_rgb
R3D_34_SCRATCH_ADT_hwos_wb
MLP_R3D_34_SCRATCH_ADT_hwos_wb
R3D_34_SCRATCH_ADT_hws_whe
MLP_R3D_34_SCRATCH_ADT_hws_whe
R3D_34_SCRATCH_ADT_hws_wohe
MLP_R3D_34_SCRATCH_ADT_hws_wohe
TWOSTREAM_I3D_SCRATCH_ADT_flow
MLP_TWOSTREAM_I3D_SCRATCH_ADT_flow
TWOSTREAM_I3D_SCRATCH_ADT_hwos_wb
MLP_TWOSTREAM_I3D_SCRATCH_ADT_hwos_wb
TWOSTREAM_I3D_SCRATCH_ADT_hws_whe
MLP_TWOSTREAM_I3D_SCRATCH_ADT_hws_whe
TWOSTREAM_I3D_SCRATCH_ADT_hws_wohe
MLP_TWOSTREAM_I3D_SCRATCH_ADT_hws_wohe
TWOSTREAM_I3D_PRETRAINED_ADT_flow
MLP_TWOSTREAM_I3D_PRETRAINED_ADT_flow
TWOSTREAM_I3D_PRETRAINED_ADT_hwos_wb
MLP_TWOSTREAM_I3D_PRETRAINED_ADT_hwos_wb
TWOSTREAM_I3D_PRETRAINED_ADT_hws_whe
MLP_TWOSTREAM_I3D_PRETRAINED_ADT_hws_whe
TWOSTREAM_I3D_PRETRAINED_ADT_hws_wohe
MLP_TWOSTREAM_I3D_PRETRAINED_ADT_hws_wohe
True label
Predicted label

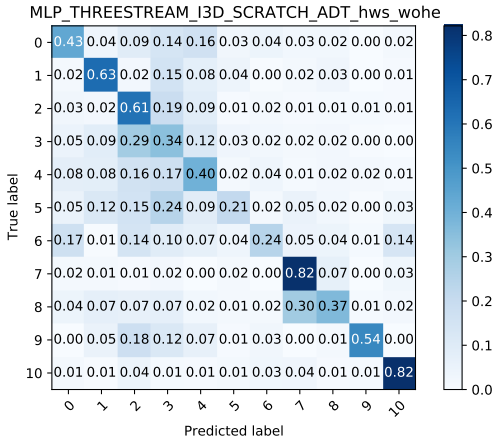

THREESTREAM_I3D_SCRATCH_ADT_hwos_wb
True label
Predicted label


MLP_THREESTREAM_I3D_SCRATCH_ADT_hwos_wb
True label
Predicted label


THREESTREAM_I3D_SCRATCH_ADT_hws_whe
True label
Predicted label


MLP_THREESTREAM_I3D_SCRATCH_ADT_hws_whe
True label
Predicted label


THREESTREAM_I3D_SCRATCH_ADT_hws_wohe
True label
Predicted label


MLP_THREESTREAM_I3D_SCRATCH_ADT_hws_wohe
True label
Predicted label

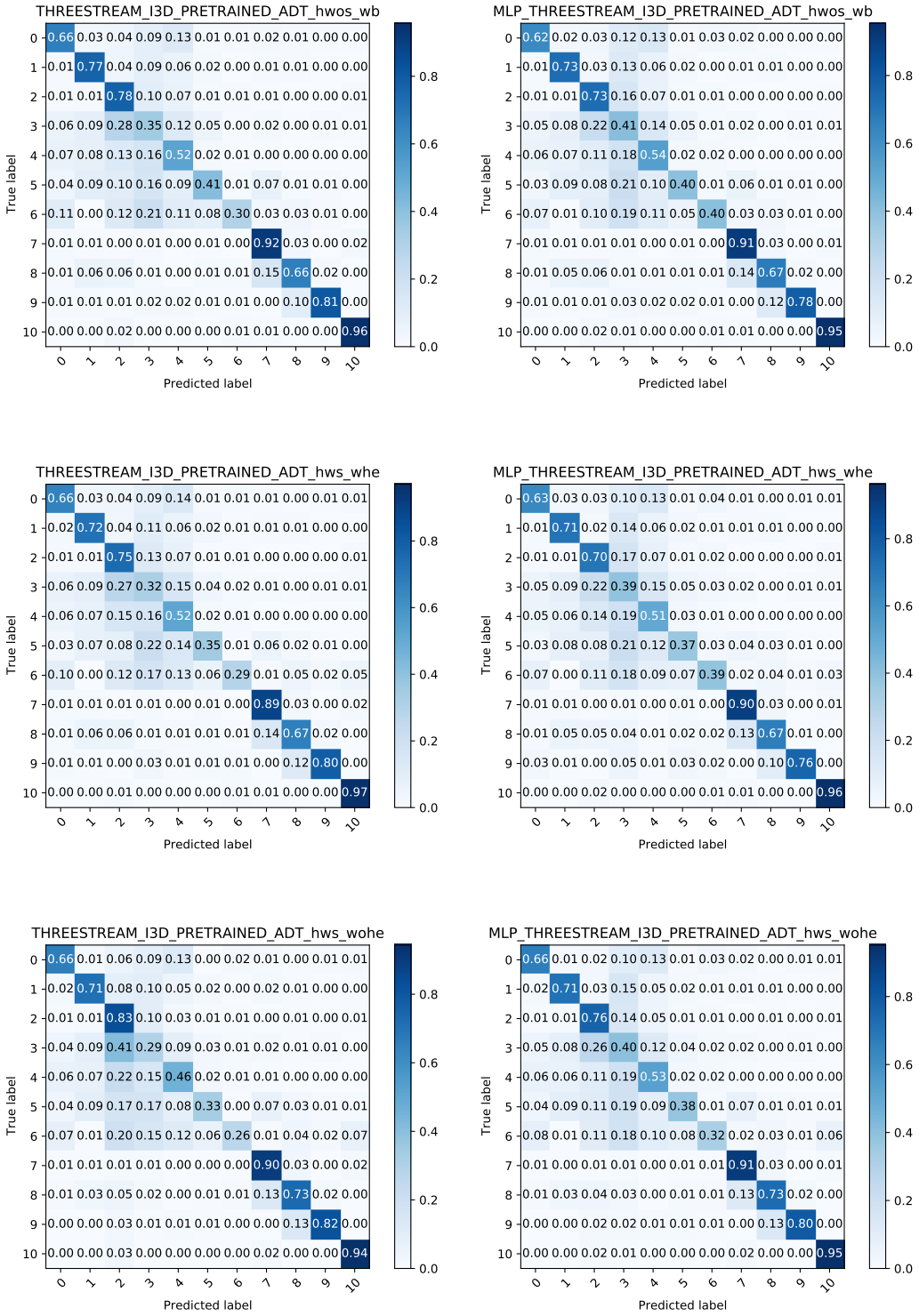


FIGURE A.19 – Comparaison entre les matrices de confusion globales, des 5 splits de Crowd-11, des Réseaux de Neurones Convolutifs et celles de leurs Perceptrons Multicouches. Pour faciliter la lecture des gains en performance apportées par l'ajout d'un PMC pour chaque RNC, nous avons mis côte à côte les matrices de confusion des réseaux RNC avec celles de leurs surcouches PMC. Sur la figure, les matrices de confusion des PMC ont le titre du réseau RNC lié préfixé avec le terme MLP (pour MultiLayer Perceptron)

Annexe B

Problème des erreurs NaN lors de l'entraînement de modèles ResNet 3D

Nous avons déjà évoqué ce problème sur stackexchange [1] et ouvert une issue sur github [2].

Nous n'arrivons pas à entraîner un modèle ResNet 3D de zéro sur la classification des clips du jeu de données Crowd-11. Ce problème se produit lorsque nous activons la régularisation L2 avec l'initialisation des poids suivant la distribution *He Normale*. Ce paramétrage attribue des valeurs *NaN* à l'erreur lors de l'apprentissage et la validation à partir d'une itération aléatoire lors de la phase d'apprentissage ou de validation. Dès l'apparition des valeurs *NaN*, elles se poursuivront jusqu'à la fin des époques d'apprentissage.

Dans le jeu de données, chaque clip vidéo destiné à alimenter le réseau ResNet3D dispose du format suivant :

16 images x 112 width x 112 height x 3 channels

Nous avons récupéré le code source de l'implémentation de l'architecture ResNet 3D de ce répertoire github : https://github.com/JihongJu/keras-resnet3d

Nous avons produit des modèles ResNet 3D en nous basant sur l'architecture à 34 couches cachées. Le choix de ce nombre de couches cachées est motivé par les bonnes performances de ce modèle illustré dans le travail de (Hara, Kataoka et Satoh, 2017) appliqué sur un jeu de données spécialisé dans la reconnaissance d'actions.

Lorsque nous activons la régularisation L2, nous attribuons la valeur suivante au facteur de régularisation : $1e-4$

À travers les graphiques affichés dans la figure B.1, nous observons l'évolution de l'erreur et de l'accuracy à la validation lorsque : (1) l'initialisation des poids suivant la distribution He Normale est employée exclusivement, (2) la régularisation L2 est employée exclusivement, et lorsque (3) les deux hyperparamètres sont activés. Nous pouvons constater qu'à partir de la 18ème époque, les erreurs lors de la validation lorsque les deux hyperparamètres sont activées (courbe verte) deviennent NaN. Nous ignorons pourquoi ce phénomène survient mais nous soupçonnons un problème de disparition du gradient.

Par ailleurs, lorsque nous avions entraîné des modèles ResNet 3D sur le jeu de données UCF-101, les erreurs NaN survenaient peu importe la configuration choisie et même si nous n'avions activé ni la régularisation L2 ni l'initialisation des poids suivant la distribution He Normale.

1. Posté sur https://datascience.stackexchange.com sous le titre **"the association of the l2 regularization and the he normal weights initialization"**
2. https://github.com/JihongJu/keras-resnet3d/issues/9

(A) L'évolution de l'erreur lors de la validation

(B) L'évolution de l'accuracy lors de la validation

FIGURE B.1 – Évolution de l'erreur et de l'accuracy lors de la validation pour les trois configurations d'hyperparamètres lors de l'apprentissage sur Crowd-11

B.0.1 Choix d'un ensemble de modèles ResNet 3D pour la classification ensembliste

Nous avons comparé les performances de deux ensembles de modèles ResNet 3D sur le jeu de données Crowd-11 afin de choisir lequel illustrer dans la classification ensembles dans la section 3.2.3.1. Les résultats de cette évaluation sont illustrées dans la table B.1. Au vu des résultats obtenus, notre choix s'est clairement porté pour l'architecture ResNet 3D ayant bénéficié exclusivement de l'initialization des poids selon la distribution He Normal.

Échantillon de test	**0**	**1**	**2**	**3**	**4**	μ	σ
R3D (w 34 layers) scratch L2 Regularization	44.54	49.87	48.89	43.92	49.39	47.32	2.55
R3D (w 34 layers) scratch He Normal Initialization	**47.42**	**52.00**	**50.13**	**48.63**	**50.43**	**49.72**	**1.57**

TABLE B.1 – Comparaison entre des ensembles de modèles ResNet 3D ayant des hyperparamètres différents

Annexe C

Classement des combinaisons d'ensembles de modèles hétérogènes

Dans cette annexe, nous classons les combinaisons d'ensembles de modèles hétérogènes par ordre ascendant selon leurs performances en termes d'accuracy lors de la classification des clips du jeu de données Crowd-11.

Rang	Combinaison	Accuracy
1	C3D_SCRATCH	33.23
2	R3D_34_SCRATCH, C3D_SCRATCH	49.41
3	R3D_34_SCRATCH	49.72
4	C3D_SCRATCH, I3D_SCRATCH	55.65
5	R3D_34_SCRATCH, C3D_SCRATCH, I3D_SCRATCH	55.89
6	TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH	56.41
7	I3D_SCRATCH	56.42
8	R3D_34_SCRATCH, I3D_SCRATCH	56.50
9	R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH	56.51
10	TWOSTREAM_I3D_SCRATCH	57.48
11	R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH	57.52
12	R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_SCRATCH	58.25
13	TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_SCRATCH	58.95
14	R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, I3D_SCRATCH	58.98
15	TWOSTREAM_I3D_SCRATCH, I3D_SCRATCH	59.48
16	R3D_34_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH	59.70
17	C3D_PRETRAINED, C3D_SCRATCH	59.92
18	R3D_34_SCRATCH, C3D_PRETRAINED	60.27
19	C3D_SCRATCH, I3D_PRETRAINED	60.32
20	C3D_PRETRAINED	60.48
21	R3D_34_SCRATCH, C3D_SCRATCH, I3D_PRETRAINED	60.60
22	R3D_34_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_SCRATCH	60.83
23	I3D_PRETRAINED	61.05
24	R3D_34_SCRATCH, C3D_PRETRAINED, I3D_SCRATCH	61.39
25	R3D_34_SCRATCH, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	61.46
26	R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH	61.51
27	R3D_34_SCRATCH, I3D_PRETRAINED	61.74
28	R3D_34_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	61.88
29	R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_SCRATCH	61.93
30	R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	62.07
31	C3D_PRETRAINED, C3D_SCRATCH, I3D_SCRATCH	62.34
32	R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED	62.47
33	C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	62.61
34	R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	62.63
35	R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_SCRATCH	62.66
36	R3D_34_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED	62.76
37	R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_PRETRAINED	62.77
38	TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	62.91
39	C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED	63.01
40	TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH	63.02
41	C3D_PRETRAINED, I3D_SCRATCH	63.09
42	I3D_PRETRAINED, I3D_SCRATCH	63.20
43	TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_SCRATCH	63.25
44	R3D_34_SCRATCH, C3D_PRETRAINED, I3D_PRETRAINED	63.27
45	R3D_34_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	63.30
46	TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_PRETRAINED	63.41
47	R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, I3D_PRETRAINED	63.41
48	C3D_PRETRAINED, I3D_PRETRAINED	63.46
49	TWOSTREAM_I3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	63.54
50	R3D_34_SCRATCH, C3D_PRETRAINED, I3D_PRETRAINED, I3D_SCRATCH	63.60
51	TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_SCRATCH	63.64
52	R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	63.77
53	C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	63.89
54	TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED	63.93
55	R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED	63.98
56	R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_PRETRAINED, I3D_SCRATCH	64.30
57	TWOSTREAM_I3D_SCRATCH, I3D_PRETRAINED	64.30
58	TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	64.35
59	C3D_PRETRAINED, I3D_PRETRAINED, I3D_SCRATCH	64.36
60	R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_PRETRAINED	64.45
61	TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED	64.77
62	TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_PRETRAINED, I3D_SCRATCH	64.87
63	TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_PRETRAINED	65.10
64	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_SCRATCH	65.62
65	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_SCRATCH, I3D_SCRATCH	66.01

TABLE C.1 – Classement par ordre ascendant des ensembles de modèles hétérogènes (partie 1)

Rang	Combinaison	Accuracy
66	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_SCRATCH	66.02
67	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, I3D_SCRATCH	66.28
68	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, I3D_SCRATCH	66.42
69	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	66.52
70	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, C3D_SCRATCH, I3D_SCRATCH	66.60
71	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, C3D_SCRATCH, I3D_SCRATCH	66.60
72	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	66.66
73	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_SCRATCH	66.79
74	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	66.80
75	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH	66.80
76	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, C3D_SCRATCH	66.83
77	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	66.89
78	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, I3D_SCRATCH	66.91
79	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_SCRATCH	66.93
80	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_SCRATCH	66.95
81	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	67.00
82	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	67.04
83	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	67.06
84	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED	67.11
85	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_SCRATCH	67.14
86	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_SCRATCH, I3D_PRETRAINED	67.18
87	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, I3D_PRETRAINED, I3D_SCRATCH	67.20
88	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH	67.24
89	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_SCRATCH	67.25
90	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	67.28
91	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_SCRATCH	67.31
92	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_PRETRAINED	67.39
93	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_SCRATCH	67.40
94	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_SCRATCH, I3D_PRETRAINED	67.43
95	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED	67.49
96	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	67.50
97	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	67.53
98	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, I3D_PRETRAINED, I3D_SCRATCH	67.53
99	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_PRETRAINED, I3D_SCRATCH	67.54
100	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH	67.55
101	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_SCRATCH	67.56
102	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED	67.56
103	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, I3D_SCRATCH	67.58
104	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, C3D_SCRATCH, I3D_PRETRAINED	67.59
105	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	67.64
106	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	67.64
107	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_PRETRAINED	67.68
108	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, C3D_PRETRAINED, I3D_SCRATCH	67.68
109	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED	67.70
110	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, I3D_PRETRAINED	67.70
111	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, I3D_PRETRAINED	67.72
112	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, I3D_PRETRAINED	67.74
113	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH	67.74
114	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, C3D_PRETRAINED, I3D_PRETRAINED, I3D_SCRATCH	67.75
115	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, I3D_SCRATCH	67.78
116	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED	67.80
117	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_PRETRAINED, I3D_SCRATCH	67.82
118	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	67.84
119	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, I3D_SCRATCH	67.85
120	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	67.85
121	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_SCRATCH, I3D_SCRATCH	67.87
122	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, C3D_SCRATCH	67.88
123	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED	67.88
124	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH	67.89
125	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	67.89

TABLE C.2 – Classement par ordre ascendant des ensembles de modèles hétérogènes (partie 2)

Rang	Combinaison	Accuracy
126	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH	67.90
127	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH	67.90
128	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, I3D_PRETRAINED	67.92
129	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, I3D_PRETRAINED	67.92
130	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_PRETRAINED	67.92
131	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_SCRATCH	67.95
132	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_SCRATCH	67.95
133	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	67.96
134	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	67.97
135	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_PRETRAINED	67.98
136	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	68.06
137	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, I3D_PRETRAINED, I3D_SCRATCH	68.07
138	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, I3D_PRETRAINED	68.09
139	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, C3D_SCRATCH, I3D_SCRATCH	68.14
140	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, C3D_SCRATCH, I3D_PRETRAINED	68.16
141	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_PRETRAINED, I3D_SCRATCH	68.17
142	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_PRETRAINED	68.17
143	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED	68.18
144	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	68.22
145	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, I3D_PRETRAINED, I3D_SCRATCH	68.22
146	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	68.27
147	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, C3D_PRETRAINED, I3D_PRETRAINED	68.27
148	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, I3D_PRETRAINED, I3D_SCRATCH	68.29
149	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH	68.30
150	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	68.30
151	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED	68.31
152	R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1	68.34
153	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_PRETRAINED, I3D_SCRATCH	68.40
154	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_SCRATCH	68.46
155	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, I3D_PRETRAINED	68.48
156	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, I3D_PRETRAINED	68.49
157	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH	68.51
158	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED	68.52
159	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_PRETRAINED	68.53
160	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED	68.54
161	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED	68.56
162	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, C3D_PRETRAINED, C3D_SCRATCH, I3D_SCRATCH	68.58
163	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_PRETRAINED	68.58
164	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, C3D_SCRATCH	68.59
165	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_SCRATCH	68.60
166	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, C3D_SCRATCH, I3D_SCRATCH	68.64
167	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH	68.64
168	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, C3D_PRETRAINED, I3D_PRETRAINED, I3D_SCRATCH	68.67
169	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_SCRATCH	68.67
170	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, I3D_PRETRAINED, I3D_SCRATCH	68.68
171	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, C3D_PRETRAINED	68.69
172	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_SCRATCH, I3D_SCRATCH	68.69
173	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_SCRATCH, I3D_PRETRAINED	68.80
174	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, I3D_SCRATCH	68.80
175	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH	68.84
176	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH	68.92
177	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	68.93
178	TWOSTREAM_I3D_PRETRAINED_OF_TVL1	69.02
179	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH	69.04
180	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_PRETRAINED	69.10
181	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, C3D_PRETRAINED, I3D_PRETRAINED	69.13
182	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	69.21
183	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	69.22
184	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED	69.24
185	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, C3D_PRETRAINED, I3D_SCRATCH	69.24
186	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED	69.30
187	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_SCRATCH	69.30
188	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_PRETRAINED, I3D_SCRATCH	69.36
189	TWOSTREAM_I3D_PRETRAINED_OF_TVL1, I3D_SCRATCH	69.39
190	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, C3D_SCRATCH	69.42

TABLE C.3 – Classement par ordre ascendant des ensembles de modèles hétérogènes (partie 3)

Rang	Combinaison	Accuracy
191	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	69.43
192	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_SCRATCH	69.47
193	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	69.53
194	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED	69.55
195	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_SCRATCH	69.57
196	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED	69.57
197	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED	69.59
198	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, C3D_PRETRAINED, C3D_SCRATCH	69.60
199	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, I3D_SCRATCH	69.60
200	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	69.62
201	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_PRETRAINED	69.62
202	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_SCRATCH, I3D_PRETRAINED	69.63
203	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	69.65
204	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	69.71
205	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED	69.81
206	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_PRETRAINED, I3D_SCRATCH	69.81
207	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, I3D_PRETRAINED	69.83
208	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH	69.84
209	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, I3D_PRETRAINED	69.86
210	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_PRETRAINED	69.87
211	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_PRETRAINED	69.87
212	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH	69.90
213	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_SCRATCH	69.90
214	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, I3D_PRETRAINED, I3D_SCRATCH	69.90
215	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, I3D_PRETRAINED	69.91
216	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_SCRATCH, I3D_PRETRAINED	69.93
217	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED	69.95
218	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_SCRATCH	69.95
219	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH, I3D_SCRATCH	69.96
220	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED	69.98
221	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, I3D_PRETRAINED	69.98
222	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, C3D_SCRATCH, I3D_SCRATCH	69.99
223	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, I3D_PRETRAINED, I3D_SCRATCH	70.02
224	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_SCRATCH, I3D_SCRATCH	70.07
225	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, I3D_SCRATCH	70.07
226	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, I3D_PRETRAINED	70.12
227	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	70.13
228	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_PRETRAINED	70.18
229	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED	70.18
230	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, C3D_SCRATCH	70.21
231	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, C3D_SCRATCH, I3D_PRETRAINED, I3D_SCRATCH	70.21
232	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_SCRATCH	70.24
233	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH	70.29
234	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, I3D_PRETRAINED	70.29
235	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, C3D_SCRATCH	70.38
236	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED, I3D_SCRATCH	70.44
237	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, I3D_SCRATCH	70.45
238	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, I3D_PRETRAINED, I3D_SCRATCH	70.45
239	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, C3D_PRETRAINED	70.47
240	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, I3D_SCRATCH	70.49
241	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, I3D_PRETRAINED, I3D_SCRATCH	70.52
242	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_PRETRAINED	70.62
243	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1	70.66
244	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, I3D_SCRATCH	70.67
245	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_SCRATCH	70.69
246	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH, C3D_SCRATCH	70.69
247	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, C3D_SCRATCH, I3D_SCRATCH	70.80
248	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, TWOSTREAM_I3D_SCRATCH	70.81
249	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, R3D_34_SCRATCH, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED	70.81
250	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, C3D_SCRATCH	70.92
251	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_SCRATCH, I3D_SCRATCH	71.02
252	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1	71.10
253	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, I3D_SCRATCH	71.25
254	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED	71.30
255	TWOSTREAM_I3D_PRETRAINED_OF_FARNEBACK_AUGMENTED, TWOSTREAM_I3D_PRETRAINED_OF_TVL1, C3D_PRETRAINED, I3D_SCRATCH	71.32

TABLE C.4 – Classement par ordre ascendant des ensembles de modèles hétérogènes (partie 4)