Réidentification de personnes dans un réseau épars de caméras instrumentant un bâtiment

par Rémi Genoux-Lubain

Tuteur : Frédéric LERASLE
Encadrants : Cyrille Equoy, Romane Scherrer Tuteur enseignant: Luc JAULIN

Remerciements

Je tiens à remercier en premier lieu Frédéric Lerasle, mon tuteur de stage, pour son encadrement, sa disponibilité, ses conseils ainsi que sa bienveillance tout au long de ce stage.

Je tiens à remercier également Cyrille Equoy pour sa confiance et ses explications. J'espère que ces travaux pourront l'aider dans le cadre de sa thèse.

Un grand merci à Romane Scherrer pour son implication et son aide pertinente au cours des problèmes rencontrés.

Enfin, je remercie toute l'équipe RAP pour leur accueil joyeux et chaleureux durant ces six mois de stages.

1 Résumé

Résumé (english version below)

La ré-identification de personnes (Person Re-Identification, ou ReID) est une tâche en développement de la vision pour ordinateur qui trouve ses applications dans les systèmes de vidéosurveillance. Les avancées récentes dans ce domaine ont été rendues possibles grâce aux progrès des réseaux de neurones profonds ainsi qu'à la mise à disposition de jeux de données de référence permettant l'entraînement et l'évaluation des modèles.

Ce rapport de stage, réalisé au LAAS-CNRS au sein de l'équipe RAP en contribution à la thèse de Cyrille Equoy, présente les méthodes étudiées et le travail réalisé afin d'adapter des approches de l'état de l'art à deux jeux de données spécifiques et horodatés : RPIfield, ainsi qu'Adream, un jeu de données développé au LAAS et de taille volontairement réduite. L'objectif final de ce travail est de fournir à l'algorithme de Recherche Opérationnelle (RO) développé dans le cadre de cette thèse, le matériel de qualité suffisante pour construire un graphe de ré-identification cohérent à l'échelle d'un réseau de caméras épars.

Le stage s'est structuré en deux phases. La première a consisté à mettre en place un cadre d'expérimentation rigoureux : ré-annotation manuelle du jeu de données Adream V2, développement d'une pipeline d'évaluation indépendante des architectures, réentraînement d'OSNet sur RPIfield et Adream, puis exploration de deux méthodes originales visant à mieux exploiter la temporalité des séquences vidéo : un clustering k-means multi-représentations des signatures, et une adaptation dynamique du seuil de similarité en fonction du profil de chaque séquence.

La seconde phase a porté sur l'évaluation et l'exploitation de X-TFCLIP, un modèle récent basé sur CLIP, qui obtient des performances très supérieures aux approches précédentes. Plusieurs méthodes de sélection d'identités ont ensuite été explorées afin de tenter de spécifier le modèle autour de l'espace de représentation de jeu de donnée Adream en utilisant un minimum de données provenant de ce jeu de données. Ces méthodes n'ont malheureusement pas eu de succès décisif face au modèle pré-entraîné.

Les expérimentations menées mettent en évidence l'apport réel mais limité des contraintes temporelles et des méthodes de traitement de tracklets sur des architectures classiques comme OSNet, ainsi que la difficulté à surpasser un modèle de fondation généraliste tel que X-TFCLIP sur des jeux de données de très petite taille, ouvrant des perspectives sur une meilleure exploitation des données disponibles plutôt que sur un réentraînement massif des modèles.

Abstract

Person Re-Identification (ReID) is a growing computer vision task with applications in video surveillance systems. Recent advances in this field have been made possible by progress in deep neural networks, as well as by the availability of reference datasets enabling model training and evaluation.

This internship report, carried out at LAAS-CNRS within the RAP team as a contribution to Cyrille Equoy's PhD thesis, presents the methods studied and the work carried out in order to adapt state-of-the-art approaches to two specific, timestamped datasets: RPIfield, as well as Adream, a dataset developed at LAAS and deliberately kept small in size. The final objective of this work is to provide the Operations Research (OR) algorithm developed as part of this thesis with signatures of sufficient quality to build a consistent re-identification graph at the scale of a sparse camera network.

The internship was structured into two phases. The first consisted of setting up a rigorous experimental framework: manual re-annotation of the Adream V2 dataset, development of an architecture-independent evaluation pipeline, retraining of OSNet on RPIfield and Adream, followed by the exploration of two original methods aimed at better exploiting the temporality of tracklets: a multi-representation k-means clustering of signatures, and a dynamic adaptation of the similarity threshold based on the profile of each tracklet.

The second phase focused on the evaluation and exploitation of X-TFCLIP, a recent CLIP-based model, which achieves performances far superior to previous approaches. Several identity selection

methods were then explored in an attempt to specialize the model towards the representation space of the Adream dataset while using a minimal amount of data from this dataset. Unfortunately, these methods did not achieve decisive success against the pre-trained model.

The experiments carried out highlight the real but limited contribution of temporal constraints and tracklet processing methods on classical architectures such as OSNet, as well as the difficulty of surpassing a generalist foundation model such as X-TFCLIP on very small datasets, opening up perspectives on making better use of available data rather than on massive retraining of models.

Table des matières

Introduction

La réidentification de personnes (person re-identification, ReID) consiste à retrouver un même individu à travers plusieurs vues capturées par des caméras disjointes, sans recouvrement de champ de vision. Ce problème a connu un intérêt croissant ces dernières années, porté par la multiplication des réseaux de vidéosurveillance et par les progrès de l'apprentissage profond. Dans un réseau de caméras fixes, les capteurs sont généralement épars et couvrent des zones disjointes du bâtiment ou de l'espace surveillé : il n'existe donc pas de continuité spatiale ou temporelle directe entre deux observations d'une même personne. Le suivi ne peut ainsi pas s'appuyer sur un simple algorithme de reconnaissance intra-caméra. Cette contrainte forte requiert des modèles de ReID une représentation visuelle discriminante et cohérente de l'individu dans un jeu de donnée de détections (dataset) afin de rassembler les vecteurs représentatifs (appelés signatures) dans l'espace des signature associé (feature space). Ce mécanisme est présenté en Figure 1 [1].

Cette tâche se heurte cependant à plusieurs difficultés. Les points de vue, les conditions d'exposition et d'éclairage, ainsi que l'orientation du sujet varient fortement d'une caméra à une autre, ce qui modifie l'apparence visuelle d'une même personne lors de sa traversée du réseau. La similarité vestimentaire entre différents individus constitue une source d'ambiguïté supplémentaire, la tenue étant souvent l'un des indices visuels les plus exploités par les modèles. C'est pourquoi, malgré des avancées impressionantes ces dernières années, la ReID reste un problème difficile sur lequel la communauté scientifique continue de travailler.

Au-delà des applications liées à l'exploitation des caméras de vidéosurveillance publiques ou privées, ce problème trouve également des débouchés dans des contextes tels que l'estimation des temps d'attente, ou encore au service des Établissements d'Hébergement pour Personnes Âgées Dépendantes (EHPAD), où il permettrait de suivre l'activité des pensionnaires et de prévenir les risques liés à une sédentarité excessive. Ce sont d'ailleurs des établissements de ce type qui ont contacté le CNRS, et qui ont en partie motivé le laboratoire à encourager des études dans ce domaine.

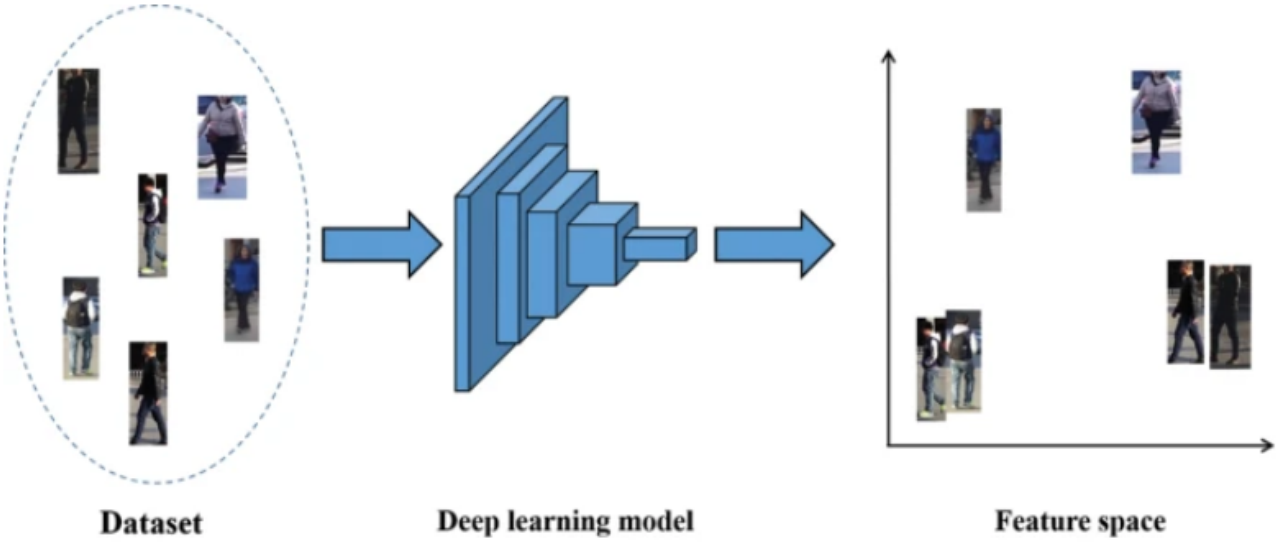


FIGURE 1 – Exemple de procédé de ReID avec un modèle de deep learning

2 Contexte

2.1 Présentation de l'environnement de travail

FIGURE 2 – LAAS-CNRS, Toulouse

Le Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS-CNRS, illustré en Figure 2) est un institut de recherche situé à Toulouse, en France, fonctionnant sous la tutelle du Centre National de la Recherche Scientifique (CNRS). Le LAAS est une unité propre du CNRS rattachée à l'Institut des sciences de l'ingénierie et des systèmes (INSIS) et à l'Institut des sciences de l'information et de leurs interactions (INS2I). Par ailleurs, il est institut Carnot, un label soulignant une politique volontariste en matière de recherche partenariale au profit du monde socio-économique. Fondé en 1968 sous le nom de « Laboratoire d'automatique et de ses applications spatiales », le laboratoire a depuis élargi son champ d'action pour traiter des systèmes complexes dans des domaines plus étendus. Avec plus de 550 chercheurs, hors personnels techniques, administratifs et logistiques, le LAAS s'organise autour de six départements scientifiques : RISC (Réseaux, Informatique, Systèmes de Confiance), ROB (Robotique), DO (Décision et Optimisation), HOPES (Systèmes Hyperfréquences et Photoniques), MNBT (Micro Nano Bio Technologies) et GE (Gestion de l'Énergie). Ces six départements scientifiques animent l'activité de 24 équipes de recherche, unités de base du laboratoire, qui se divisent elles-mêmes en groupes de projet plus restreints permettant un travail ciblé et innovant.

La thèse de Cyrille Equoy, à laquelle ce stage contribue, est mené au sein du département ROB, conjointement avec les équipes RAP et ROC, présentées auparavant. Ce stage a été encadré par des chercheurs de l'équipe RAP, spécialisée en perception au sein du département ROB. Les recherches de l'équipe RAP portent sur la conception, le prototypage, l'implémentation et l'évaluation d'algorithmes de perception visuelle, de commande et navigation référencées capteur, de manipulation mobile interactive, ainsi que de modélisation et de fusion de données multicapteurs. Mon travail a été encadré par le professeur Frédéric Lerasle, la post-doctorante Romane Scherrer et le doctorant Cyrille Equoy.

2.2 Définition du problème

On distingue classiquement deux formulations différente du problème de ReID. La ReID image (ou single-shot) associe à chaque identité une seule image de référence, tandis que la ReID vidéo (ou multi-shot) exploite une séquence d'images consécutives extraites d'un suivi, appelée tracklet (présenté en Figure 3).

Dans les deux cas, le problème est généralement formulé comme une tâche de recherche entre deux ensembles : la gallery, une base de référence contenantun ensemble d'images dont les identités sont

connue, et la query, l'image requête dont on cherche à retrouver l'identité correspondante dans la gallery. Chaque image est d'abord encodée sous la forme d'une signature, puis l'association entre la query et les éléments de la gallery est ensuite réalisée en calculant une distance cosinus entre leurs signatures respectives. Un classement des signatures en fonction de cette distance est levé, et le plus proche sont les signatures des tracklets d'identité similaire, meilleure et réalisée la tâche de réidentification. Cela revient à formuler le problème de ReID comme une tâche de mesure de similarité dans l'espace des signatures.

Il est clair que malgré son développement plus tardif, la ReID vidéo se trouve être plus efficace que la ReID image. En effet, la ReID vidéo peut utiliser la variation de perception d'une même personne au court de la tracklet pour mieux discriminer ses attributs et plus efficacemnet apparier une deux tracklets appartenant à la même identité.

FIGURE 3 – Prise de vue de différentes personnes sur deux caméras à champs disjoints

Le problème dans cette manière de procéder surgit du fait qu'on ne considère que la proximité des signatures dans l'espace des représentations pour juger de la possible appartenance à la même identité. Cependant, il existe de nombreux cas où de simples informations comme la disposition des caméras dans le réseau, ou le temps de trajet pour passer d'une caméra à l'autre rend certains appariements incompatibles. Pour améliorer ce processus, l'usage d'horodatages associés aux différentes vues permet d'imposer des contraintes de cohérence temporelle supplémentaires, en excluant les appariements physiquement impossibles (une personne ne pouvant apparaître simultanément à deux endroits distants). Cette méthode est celle utilisée par Cyrille Equoy dans le cadre de sa thèse, à l'initiative de ce stage.

Cette thèse est menée au sein des deux groupes de recherche RAP (Robotics, Action, Perception) et ROC (Operations Research, Combinatorial Optimization and Constraints). Son objectif est de construire, à partir des horodatages, de la topologie du réseau et du temps minimal de déplacement entre caméras, un graphe où les nœuds correspondent aux détections et les arcs aux associations envisageables entre elles. L'algorithme construit alors des chemins cohérents à travers ce graphe, chaque chemin représentant le parcours d'une identité dans le réseau. Cela permet d'écarter les appariements incompatibles avec les déplacements réels d'un individu et de garantir une cohérence globale entre les observations à l'échelle du réseau de caméra.

L'objectif est de fournir à ces travaux de thèse un ensemble de vecteurs représentatifs (appelé signature) par détection pour construire le graphe, dans le but de procéder à la réidentification de ces personnes. Ces signatures pouront permettre de tester et d'évaluer ces algorithmes de traitement

de graphes, c'est à dire de recherche opérationnelle , développés dans le cadre de cette thèse. Elles seront volontairement de qualité variables afin de déterminer les limites ses algorithmes ainsi que leur domaine d'application.

2.3 Sujet de stage

Le stage se structure en deux temps distinct et complémentaires. Dans un premier temps, il s'agit de lister et mobiliser des algorithmes de ReID image existants afin d'améliorer leur performances sur des jeux de données vidéo. Pour cela, il est nécessaire de développer des moyens d'exploiter la temporalité des différentes tracklet afin d'améliorer leur représentation, c'est à dire, la qualité des signatures. Cela encourage donc une bonne compréhension des architectures des modèles et des progrès possibles.

Dans un second temps, l'objectif est tout autre puisqu'il a pour objet d'utiliser les meilleurs modèles de ReID vidéo afin d'essayer d'améliorer leurs performances sur un jeu de données jugé difficile, de taille réduite, tout en utilisant un minimum de donnée pour l'entraînement. Le bon usage d'autres jeux de données annotés pour améliorer les résultats est donc un des enjeux de cette deuxième partie de stage.

2.4 Calendrier prévisionnel du stage

Pour réaliser ces objectifs, chaque partie a été découpée en plusieurs sous objectifs, détaillés dans le Tableau 4.

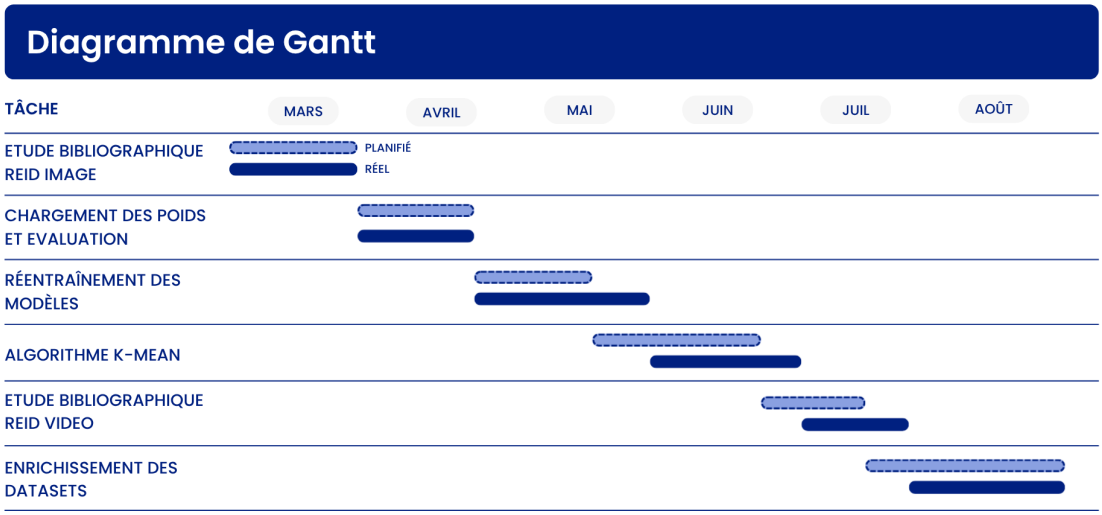


FIGURE 4 – Diagramme de Gantt prévoyant le déroulement du stage

Un léger retard a été pris lors de la phase de réentraînement sans affecter de manière importante la fin du stage.

3 État de l'art

3.1 Évolution des approches

Les premières méthodes de ReID étaient des méthodes de ReID image, elles reposaient sur des descripteurs visuels construits manuellement (hand-crafted), tels que des histogrammes de couleur et de texture ou des descripteurs comme LOMO [2]. Ces approches, limitées par leur capacité de représentation, ont rapidement été supplantées par les réseaux de neurones convolutifs (CNN).

L'ère du deep learning a d'abord été dominée par des architectures convolutives profondes comme Resnet [3], généralement entraînées avec une combinaison de perte de classification, et structurées autour de blocs résiduels permettant d'entraîner des réseaux beaucoup plus profonds sans dégradation des performances. Par la suite, des architectures légères spécifiquement conçues pour la ReID ont vu le jour, détectant chacune des caractéristiques à une échelle spatiale différente. Ces mesures permettent d'apprendre des représentations, à la fois discriminantes et généralisables, tout en restant nettement plus compact que les architectures profondes classiques dont elles s'inspirent.

À partir de 2021, les architectures à base de Transformers ont brutalement fait leur apparition dans le haut des classements. Les Transformers sont issu des méthodes de traitement automatique des langages (comme BERT [4] par exemple) utilisés dans les grands modèles de langage. C'est une nouvelle manière de traiter les images est donc de mettre à profit ces blocs de traitements de l'information en les découpant en subsections appelées patchs. Chaque patch est transformé en une représentation vectorielle qui peut être traité comme celles des différents mots dans les grands modèles de langage. Cet ajout permet de donner plus d'importance à une partie d'une image qu'à une autre (par exemple, une personne en délaissant le fond de l'image). Ces architectures s'appellent des transformers visuels (Vision Transformers, ViT [5]) et s'appuie sur les progrès récents des modèles de langages.

TransReID [6] (ReID image) est notamment le premier framework à reposer entièrement sur un Transformer pur pour la tâche de ReID. Ce modèle ajoute une perte triplet à un ViT et introduit un module d'encodage d'informations annexes (SIE) qui encode des informations non visuelles telles que la caméra ou le point de vue afin de réduire les biais liés à ces facteurs, on peut y apercevoir les prémisces des travaux actuels. Ces architectures ont permis de mieux capturer les dépendances à longue portée entre régions de l'image et de donner plus d'importance à ce qui est discriminant.

Plus récemment, l'essor des modèles de pré-entraînement vision-langage, en particulier CLIP, a ouvert une nouvelle direction de recherche pour la ReID. Ces modèles, d'abord spécialisé pour faire la correspondance entre les informations textuelles et visuelles ont, avec le développement massif des jeux de données texte/image, progressé à tel point qu'ils ont fini par développer une représentation assez fine des images et des vidéos. C'est pourquoi leur efficacité, soutenue par des millions d'exemples, a été mise en avant ces dernières années, et donc utilisés dans les modèles les plus récents, dont des modèles de ReID vidéo.

3.2 Jeux de données et benchmarks de référence

Les jeux de données les plus utilisés dans la littérature ReID pour l'entraînement et l'évaluation des modèles utilisés dans ce stage sont présentés dans le Tableau 1. On peut noter la particularité de AG-VPREID qui contient des images provenant de drones ainsi que de caméras de surveillances.

Table 1 – Caractéristiques des principaux benchmarks ReID de la littérature.

Jeu de donnée	Nb. identités	Nb. caméras	Nb. images	Nb. images (total)
Market-1501	1 501	6	12 936	32 668
DukeMTMC-reID	1 404	8	16 522	36 411
MSMT17	4 101	15 (12 ext./3 int.)	30 248	124 068
AG-VPREID	6 632	6	32 321	9.6M

À ces benchmarks standards non horodatés donc s'ajoutent des jeux de données moins présents

dans la littérature mais disposant d'informations temporelles utilisés spécifiquement dans ce stage, RPIfield et Adream (Cf Partie 4.1.1). Ces jeux de données dont les caractéristiques et les difficultés propres (taille réduite, conditions de capture spécifiques) seront détaillées dans la partie méthodologie.

3.3 Métriques d'évaluation

L'évaluation des modèles de ReID repose principalement sur deux métriques complémentaires : la mean Average Precision (mAP), qui reflète la qualité du classement de l'ensemble des images pertinentes dans la galerie pour chaque requête, et le Cumulative Matching Characteristic (CMC), généralement rapporté à Rank-1, qui mesure la probabilité que la bonne identité apparaisse en première position du classement.

Pour une requête q, l'Average Precision (AP) se définit à partir de la précision cumulée à chaque position pertinente du classement :

$$AP_q = \frac{1}{N_{gt}} \sum_{k=1}^{n} P(k) \cdot rel(k) \tag{1}$$

où N_{gt} est le nombre total d'images pertinentes (même identité) présentes dans la galerie pour la requête q, n le nombre total d'images de la galerie, $P(k)$ la précision calculée sur les k premiers résultats retournés, et $rel(k)$ une fonction indicatrice valant 1 si l'image classée en position k est pertinente et 0 sinon. La mAP est alors la moyenne des AP calculées sur l'ensemble des Q requêtes :

$$mAP = \frac{1}{Q} \sum_{q=1}^{Q} AP_q \tag{2}$$

Le CMC à Rank-k, quant à lui, mesure la proportion de requêtes pour lesquelles au moins une image pertinente figure parmi les k premiers résultats retournés :

$$CMC(k) = \frac{1}{Q} \sum_{q=1}^{Q} \mathbf{1}\left[rank_q \leq k\right] \tag{3}$$

où $rank_q$ désigne le rang de la première image correcte retournée pour la requête q, et $\mathbf{1}[\cdot]$ la fonction indicatrice.

Ces deux métriques sont calculées à partir d'une matrice de distances (distmat) entre les signatures de requête et les signatures de galerie, obtenue par une mesure de similarité telle que la distance euclidienne ou cosinus.

Une explication plus détaillée des méthodes de calcul de ces indices afin de mieux les comprendre est disponible en annexe .1.

3.4 Modèles utilisés

ResNet-50, proposé par He et al. [3] est un réseau de neurones convolutif profond fondé sur le principe des connexions résiduelles (Cf Figure 5). Ces connexions permettent de faciliter l'entraînement de réseaux de grande profondeur en limitant le problème de disparition du gradient. Le réseau dispose de poids préentraînés sur ImageNet qui génèrent effectivement des signatures à partir d'image, cependant, du fait qu'il ai été entraîné pour reconnaître d'autre classes que des personnes, les signatures du modèles des différentes personnes sont très proches dans l'espace latent, et par conséquent, les identités indissociables.

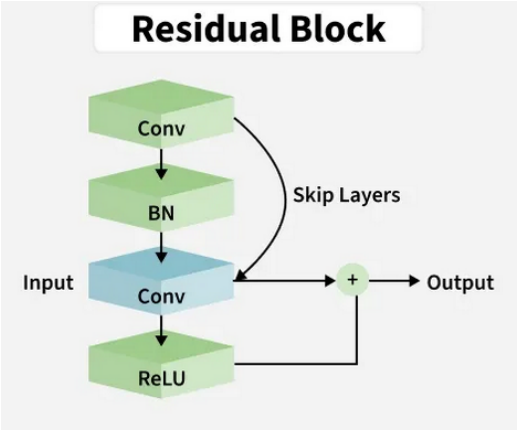


FIGURE 5 – Bloc de base de Resnet

L'architecture est composée de 50 couches organisées en plusieurs blocs résiduels qui extraient progressivement des caractéristiques visuelles de plus en plus abstraites. S'il est peu efficace dans le domaine de la ré-identification de personnes, il est en revanche fréquemment utilisé comme réseau d'extraction de caractéristiques, c'est à dire de la représentation des différentes images en vecteur, sur lequel est basé les systèmes plus complets de ré-identification.

OSNet, proposé par Zhou et al. [7] est une architecture spécialement conçue pour la ré-identification de personnes et la première à faire référence dans le domaine. Son objectif est de capturer simultanément des informations visuelles à différentes échelles spatiales afin de mieux représenter les variations d'apparence d'un individu.

Pour cela, OSNet introduit des blocs Omni-scale qui combinent plusieurs chemins convolutifs de profondeurs différentes au sein d'un même bloc. Cette conception permet d'extraire à la fois des détails locaux, tels que les textures ou les accessoires vestimentaires, et des informations plus globales liées à la silhouette ou à la posture. Les différentes caractéristiques sont ensuite fusionnées grâce à un mécanisme de pondération par canal, permettant au réseau de sélectionner les informations les plus pertinentes. Une image représentant le bloc de base de OSNET est présenté en Figure 6.

Ce réseau a servi de base à de nombreuses autres méthodes de ré-identification avant le développement des transformeurs.

ABD-Net, étudié par Chen et al. [8] (*Attentive But Diverse Network*) est une architecture dédiée à la ré-identification de personnes qui combine une représentation globale de l'individu avec une représentation attentive des régions les plus discriminantes de l'image.Basé sur un backbone ResNet-50, le réseau se divise, après les premières couches convolutives, en deux branches parallèles. La première branche produit une représentation globale de la personne, tandis que la seconde intègre des mécanismes d'attention spatiale et par canal comme OSNET. Le but étant aussi de traiter l'image au regard de différentes échelles.

Les caractéristiques produites par les deux branches sont ensuite concaténées afin de former un vecteur de représentation plus riche. L'entraînement du modèle repose sur une combinaison de plusieurs fonctions de loss, dont la triplet loss qui favorise l'écartement des différentes identités dans l'espace latent.

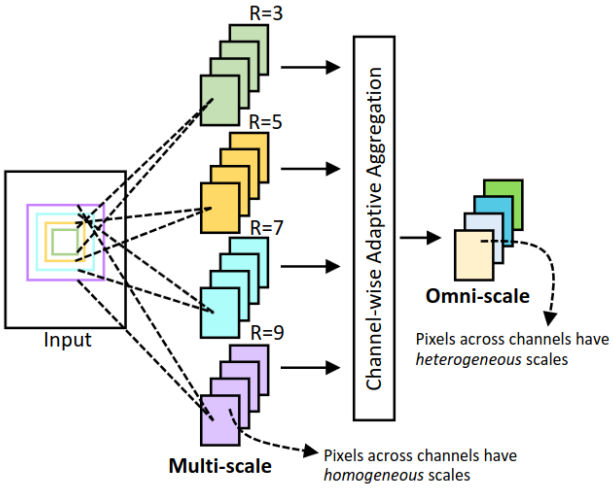


FIGURE 6 – Principe de fonctionnement de OSNET en traitant l'image sà plusieurs échelles

TF-CLIP, soumis par Yu et al. [9] (Text-Free CLIP for Person Re-Identification) adapte CLIP (Contrastive Language-Image Pre-training), un modèle pré-entraîné qui apprend un espace de représentation commun entre les images et le texte, à la ré-identification de personnes sans recourir à des descriptions textuelles. Le fonctionnement de TF-CLIP sera repris dans la section suivante avec son adaptation à un problème de réidentification de personne entre des vues drones et des vues de caméras de surveillance (Air-Ground ReID, AG ReID)

3.5 Problématiques connues et champs de recherche

Deux difficultés supplémentaires, moins centrales dans la littérature ReID classique mais essentielles dans le cadre de ce stage, méritent d'être soulignées.

La première concerne la généralisation inter jeu de donnée et le décalage de domaine (domain shift). Un modèle entraîné et évalué sur un même jeu de données obtient généralement de bonnes performances, mais celles-ci se dégradent fortement lorsqu'il est évalué sur un jeu de données différent de celui utilisé à l'entraînement, en raison des différences de caméras, de contextes et de populations d'identités entre les jeux de données. Plusieurs architectures (ReNorm 1 [10], BAU [11]) ont essayés de résoudre ce problème en utilisant des pipeline de normalisation qui ont pour objectif de rendre la représentation indiférente de l'environnemnet et de la caméra utilisée .

La seconde concerne le cas des petits jeux de données difficiles. Ces petits jeux de données tendent à produire des performances plus faibles, en raison du nombre restreint d'images d'entraînement disponibles après séparation train/test. De plus, l'entraînement sur de petits jeux de données abouti à un modèle peu général, et dont les performances seront moins reproductible dans une optique de déploiement.

Enfin, la ReID a vu son champ de recherche s'étendre avec l'amélioration de ses modèles et des performances associées. En effet, de nouvelles problématiques ont émergées comme la ReID cross-modale (infrarouge/visible), la ReID appliquée non pas sur des personnes mais sur des véhicules, ou encore l'Extra Far ReID (XF-ReID), qui utilise des caméras positionnées très loin des objectifs filmés.

3.6 Synthèse et positionnement

Cette revue de la littérature met en évidence une évolution nette des approches de ReID, des descripteurs hand-crafted vers les architectures CNN puis Transformer, jusqu'aux modèles de pré-entraînement vision-langage de type CLIP, qui constituent aujourd'hui l'état de l'art. Elle révèle cependant trois angles morts relativement moins traités : la caractérisation de signatures de bonne qualité pour la Recherche Opérationnelle, la fusion cohérente de plusieurs jeux de données hétérogènes

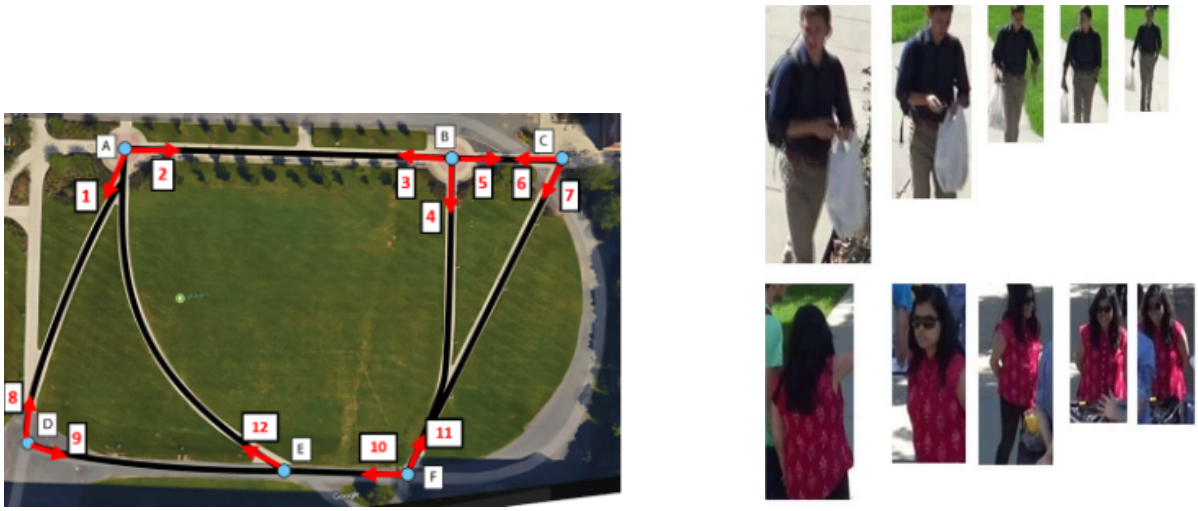
pour limiter le décalage de domaine ainsi que l'extraction de signatures performantes sur des jeux de données de très petite taille. Ce sont précisément ces trois axes qui structurent le travail présenté dans ce rapport.

4 Extractions de signatures et évaluations

4.1 Cadre d'expérimentation et contraintes associées

4.1.1 Choix des jeux de données horodatés

Le jeu de donnés disposant d'information temporelles qui a été sélectionné pour l'entraînement et les évaluations est RPIfield [12]. Il contient 112 identités, réparties sur 12 caméras et 4,000 tracklets. La figure 8a présente RPIfield avec la carte de configuration des caméras. La figure 10b montre des exemples des différentes identités présente dans le jeu de données, la différence de taille s'explique par la taille apparente des identités sur les images avant la détection. On peut remarquer que deux des caméras (5&6) sont l'une en face de l'autre et possèdent un champ de vision partiellement commun. Cette configuration est incompatible avec l'hypothèse de caméras en champs partiellement chevauchantes sur laquelle repose l'algorithme de recherche de chemins. Par conséquent, une de ces deux caméras a été exclue du traitement afin de respecter les hypothèses du problème étudié.

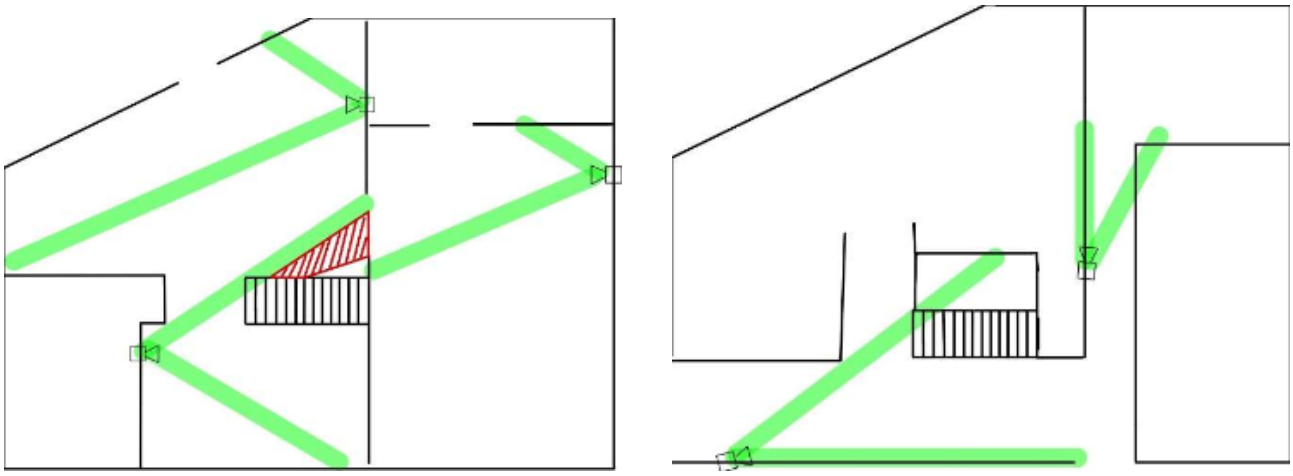


(a) Positionnement des caméras de RPIfield

(b) Extrait des bounding boxes de RPIfield

FIGURE 7 – Présentation de RPIfield.

Afin d'augmenter le volume de données disponibles et d'évaluer la robustesse des différentes méthodes d'extraction de signatures sur plusieurs environnements, un second jeu de données a été conçu au sein du LAAS préalablement à ce stage. Ce jeu de donnée, nommé Adream, a été acquis dans l'un des bâtiments du laboratoire à l'aide d'un réseau de cinq caméras synchronisées et horodatées. La représentation des lieux et des caméras est représenté sur la figure 8. Deux sessions d'acquisition, d'une durée d'environ vingt minutes chacune, ont été réalisées au cours desquelles plusieurs personnes se déplaçaient librement dans les couloirs du bâtiment. Chaque session constitue un sous-ensemble indépendant du jeu de données, notés Adream V1 et Adream V2, comportant chacun une vingtaine d'identités. L'ensemble représente 254 tracklets et fournit un complément intéressant à RPIfield pour l'évaluation des méthodes d'extraction de caractéristiques.



(a) Rez-de-chaussée (b) Premier étage

FIGURE 8 – Plan du batiment Adream et des caméras utilisées.

TABLE 2 – Caractéristiques des jeux de données utilisés.

Jeu de données	Nb. identités	Nb. caméras	Nb. tracklets	Nb. images
RPIfield	112	12 (11 utilisées)	597	115 601
Adream_V1	20	5	110	4292
Adream_V2	20	5	144	66693

Le jeu de données Adream a été conçu de telle sort qu'il soit de taille réduite, destiné à représenter un environnement complexe dans lequel les approches classiques d'extraction de caractéristiques rencontrent des difficultés.

Le jeu de données Adream est parmi les jeux de données les plus difficiles si comparé avec ceux de la littérature. Il y a par exemple les conditions de capture qui sont par endroits dégradées : l'une des caméras pointe vers un couloir obscur, produisant des images difficilement exploitables aussi bien pour l'annotation que pour l'apprentissage. À cela s'ajoute une ambiguïté visuelle marquée entre certains individus, notamment liée au port de tenues similaires : quatre personnes portent ainsi des gilets jaunes identiques, obligeant les modèles à apprendre à ne pas accorder une importance excessive à ce type d'attribut vestimentaire. Enfin, certaines identités n'apparaissent que peu de fois dans les détections, ce qui rend plus difficile une correspondance entre occurrences et constitue une difficulté supplémentaire.

En raison du faible volume de données horodatées disponibles, il constitue cependant un ajout intéressant pour étudier l'apport des contraintes temporelles, qui se trouvent déterminantes. Avoir des données supplémentaires, réalisées dans des conditions différentes permet en outre de tester la robustesse des méthodes utilisées.

Par ailleurs, les algorithmes de ré-identification de personnes (ReID) atteignent aujourd'hui des performances très élevées sur les jeux de données de référence (mAP supérieure à 90 %). Dans ces conditions, les représentations latentes produites regroupent déjà efficacement les signatures appartenant à une même identité. Dans ce cas, l'utilisation d'algorithme couteux en puissance de calcul deviendrait superfétatoire.

À l'inverse, le jeu de données Adream, dont les détections sont moins évidentes à transformer en signatures, génère des représentations latentes pour lesquelles les tracklets d'une même identité ne sont plus difficilement identifiées par les modèles. Ce contexte permet ainsi d'évaluer l'intérêt réel de l'intégration des contraintes temporelles. Il convient toutefois de trouver un équilibre : si les signature sont trop chaotiques, celles associées à une même identité deviennent trop éloignées dans l'espace latent, auquel cas l'ajout des informations temporelles ne suffit plus à compenser les erreurs de représentation et n'apporte qu'un gain limité en termes de performances de classification.

4.1.2 Contraintes à appliquer aux signatures

L'une des premières problématiques abordées au cours de ce stage consistait à quantifier la qualité des signatures. Plus précisément, il s'agissait d'identifier les propriétés à surveiller afin que l'intégration des contraintes temporelles puisse améliorer les performances de ré-identification tout en restant pertinent. Le constat est simple, comme un algorithme de recherche dans un graphe est assez coûteux en puissance de calcul, si un simple seuil sur la similarité suffit à déterminer ces identités, il perd de son intérêt. Mesurer et observer l'impact de la qualité des signatures permet donc de déterminer la plage de fonctionnement optimale de ces algorithmes. C'est pourquoi évaluer les algorithmes de recherche opérationnelle à partir de signatures de qualité graduelle est nécessaire.

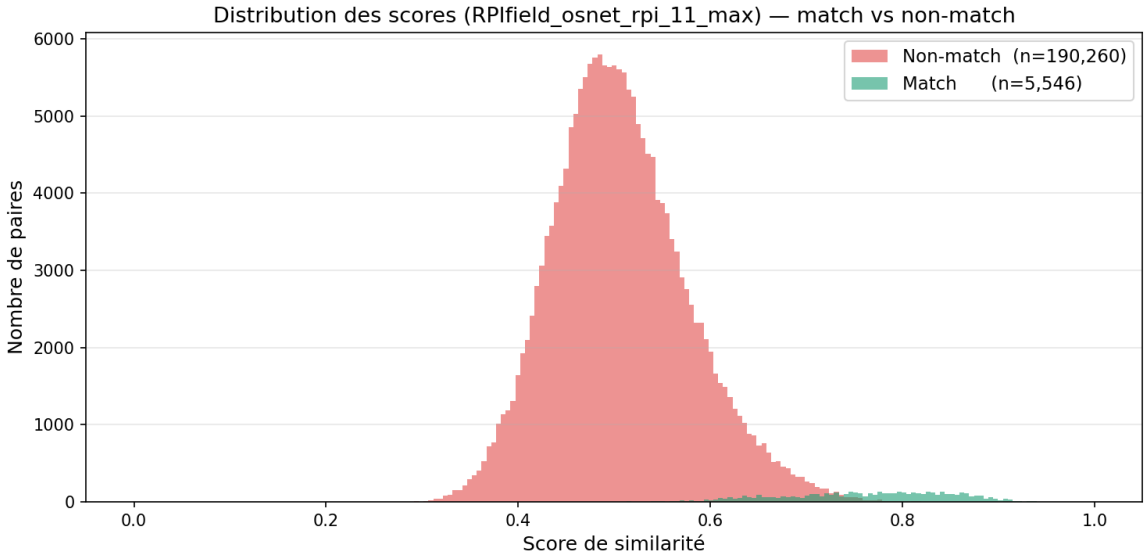


FIGURE 9 – Répartition des scores de similarités entre les tracklets

La figure 9 présente la distribution des scores de similarité (similarité cosinus) calculés entre les différentes caractéristiques extraites. Deux distributions sont distinguées : la première correspond à l'ensemble des distances lorsque les identités mesurées sont différentes, tandis que la seconde concerne les distances entre les tracklets quand les deux identités sont identiques. On peut noter un déséquilibre entre les deux courbes dus à toutes les paires d'identités différentes qui sont bien plus nombreuses que les paires de même identités. Dans une situation idéale, les caractéristiques d'une même personne présenteraient un score de similarité élevé, traduisant leur proximité dans l'espace latent, alors que les caractéristiques appartenant à des individus différents présenteraient un score de similarité faible.

On observe bien que l'algorithme de recherche dans un graphe trouve principalement son intérêt dans la zone de recouvrement entre ces deux distributions (ici, pour des scores dans $[0.6, 0.75]$). Cette région correspond aux situations ambiguës où la mesure de similarité n'est plus suffisante pour déterminer précisément si le couple requête/gallerie appartient à la même identité. Par exemple si l'on prend un seuil de 0.6 pour attribuer les identités afin de ne rater aucune paire de même identité, on se retrouverais avec une quantité de faux positifs importante, voir majoritaire. L'objectif de l'algorithme est donc d'exclure ces associations incohérentes, afin d'améliorer la précision de la classification finale.

C'est pourquoi le mAP peut être associé à une autre mesure afin de compléter l'évalution de la qualité des signatures. En effet, pour observer des améliorations significatives de la classification grâce à l'algorithme de Recherche Opérationnelle, on préfera des signatures dont les distances entre les identités similaires sont très petites, contrairement aux distances entre les identités différentes qui elles seraient plus élevées. Cette information a été mesurée à l'aide d'un calcul de séparation :

$$Séparation = \frac{|\mu_{match} - \mu_{non_match}|}{\sigma_{match} + \sigma_{non_match}} \tag{4}$$

Avec $\mu_{match} et \mu_{non_match} la moyenne des similarités entre les paires de même identité et les paires d'identités diférer$

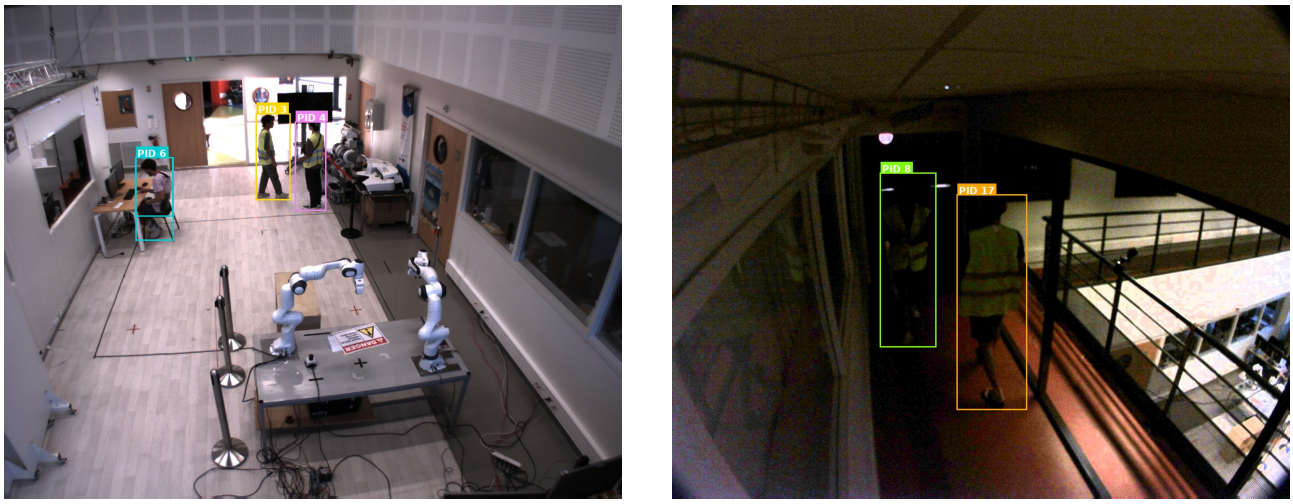
On peut à présent quantifier efficacement la qualité des signatures demandées dans le cadre de la thèse que ce soit dans la qualité de l'entourage de chaque signature (mAP, CMC) mais aussi dans l'écartement global des regroupements de signatures les uns avec les autres (similarité). C'est au regard de ces méthodes de mesures qu'il a été possible de créer un gradient de jeux de signatures de difficulté variable, afin de mieux évaluer la plage de fonctionnement des méthodes de la thèse.

4.2 Travaux réalisés et résultats

4.2.1 Annotation du jeu de données Adream V2

Afin de limiter les erreurs de ré-identification imputables non pas au modèle mais à des erreurs d'annotation, une ré-annotation manuelle du jeu de données Adream V2 s'imposait. Cette étape s'avère d'autant plus importante que le nombre de tracklet est faible : toute erreur d'étiquetage y prend un poids proportionnellement plus important et se répercute directement sur les métriques d'évaluation (mAP, CMC) qui avaient tendance à chuter énormément. La plupart de ces mauvaises détections étaient dues aux détections avec très peu d'informations (parties de corps seulement, contraste insuffisant, surexposition) ou à des erreurs d'annotation déjà présente dans le jeu de données.

L'annotation a été réalisée en deux parties, une reprise des flux vidéos pour extraire les détections des personnes présente au moment de l'expérience dans le bâtiment (toutes volontaires) en utilisant un modèle Yolo (Yolo26 [13]), et une annotation mannuelle postérieure pour attribuer les identités à chaque détection de personne. Une exemple du résultat du jeu de données annoté est présentée en Figure 10. Les images annotées sont ensuites regroupées en tracklets ou seules les détection (qui prennent la forme de bounding boxs) de chaque personne est conservée.



(a) Caméra 1 (b) Caméra 5

FIGURE 10 – Images annotées du jeu de données Adream V2

4.2.2 Pipeline d'évaluation indépendante des modèles

Afin de pouvoir comparer équitablement différents modèles de ré-identification, une pipeline d'évaluation a été développée indépendamment de l'architecture de ces modèles. L'objectif est de disposer d'un socle commun de chargement, d'extraction et d'évaluation, de sorte que seule la partie extraction de signatures (et donc les poids du modèle) varie d'une expérimentation à l'autre, garantissant ainsi une comparaison sur des bases strictement identiques.

Cette pipeline se décompose en trois étapes : d'abord, il y a l'instanciation de l'architecture correspondant à chaque modèle et chargement des poids associés, chaque modèle nécessitant sa propre

définition d'architecture pour être correctement initialisé.

Ensuite, on éxecute le passage des images dans le modèle pour extraire les signatures des tracklets, chacune étant associé à sa caméra, son numéro d'identité ainsi qu'au nombre de réapparitions déjà recensé au sein de la même caméra (`pid`, `camid`, `reap`), ce triplet définissant entièrement chaque échantillon pour les étapes de matching et d'évaluation qui suivent.

Enfin, à partir des signatures extraites, le mAP et les indices CMC sont calculées, permettent une comparaison directe entre modèles.

Les premières expérimentations menées avec cette pipeline se concentrent sur OSNet. Ce choix s'explique par le fait qu'il s'agit d'un modèle de référence dans le domaine de la ré-identification, antérieur à l'introduction des modèles à mécanismes d'attention, et qu'il avait déjà été utilisé dans le cadre de travaux de thèse précédents, ce qui en fait une base de comparaison pertinente pour expérimenter de nouvelles méthodes de traitement des tracklets.

4.3 Paramètre d'évaluation et comparaison des modèles

4.3.1 Réentraînement de OSNet sur les jeux de données RPIfield et Adream

OSNet est disponible sous plusieurs architectures et avec plusieurs jeux de poids pré-entraînés sur les données de référence (Cf Tableau 1). Cependant, aucun de ces poids n'est pré-entraîné sur RPIfield, et *a fortiori* aucun ne l'est sur les jeux de données Adream. L'utilisation directe de ces poids expose donc à une importante baisse de performance due à la nouveauté du domaine évalué comme expliqué précédemment. C'est pourquoi un réentraînement d'OSNet directement sur ces jeux de données a été réalisé.

Plusieurs paramètres ont été mis à disposition pour cette phase de réentraînement :

OSNet repose sur le principe Omni-Scale, décliné en plusieurs architectures. On distingue notamment deux familles, OSNet et OSNet-AIN, chacune disponible en plusieurs largeurs de réseau (x1, x0.75, x0.5, x0.25), les versions les plus légères étant moins coûteuses en temps de calcul mais également moins performantes.

Deux fonctions de perte (loss) ont également été testées, la cross-entropy, qui optimise la classification directe des identités, et la triplet loss, qui vise directement à structurer l'espace des signatures en rapprochant les représentations d'une même identité et en éloignant celles d'identités différentes.

Enfin, deux configurations de partionnement entraînement/test du jeu de données ont été testées, une répartition 60 % / 40 % et une répartition 70 % / 30 % entre les ensembles d'entraînement et de test.

Le tableau 3 présente les résultats obtenus pour l'ensemble des combinaisons de ces paramètres.

Base du modèle	Split	Loss	mAP	R1
osnet_x1_0	0,6	crossentropy	66	94
osnet_x1_0	0,6	triplet	41	75
osnet_x1_0	0,7	crossentropy	74	95
osnet_x1_0	0,7	triplet	50	73
osnet_ain_x1_0	0,6	crossentropy	68	92
osnet_ain_x1_0	0,6	triplet	51	82
osnet_ain_x1_0	0,7	crossentropy	74	93
osnet_ain_x1_0	0,7	triplet	53	75

TABLE 3 – Résultats de réentraînement d'OSNet sur RPIfield selon l'architecture, la fonction de perte et le partitionnement entraînement/test.

On observe un net avantage de la cross-entropy loss par rapport à la triplet loss, avec un écart de mAP compris entre 21 et 25 points selon la configuration. On note également un léger avantage de l'architecture OSNet-AIN par rapport à l'architecture OSNet classique, particulièrement visible sur

la fonction de perte triplet (par exemple +10 points de mAP pour le split 0,6), tandis que l'écart se resserre voire s'annule avec la cross-entropy loss (74 % de mAP pour les deux architectures au split 0,7). Enfin, le split 0,7/0,3 surpasse systématiquement le split 0,6/0,4, quelle que soit l'architecture ou la fonction de perte utilisée. Ces résultats sont cohérents avec les conclusions des travaux ayant introduit OSNet-AIN [14].

4.3.2 Clustering k-means sur les signatures extraites

OSNet est un modèle conçu pour la ré-identification à l'échelle de l'image, et non de la tracklet : il est entraîné à reconnaître une identité à partir d'une unique image. Or, dans le cas d'une tracklet de plusieurs images, cette limitation pose problème pour construire une signature représentative. On peut dans ce cas soit choisir une image au hasard dans la tracklet et n'utiliser que sa signature, ce qui est peu représentatif de l'ensemble de la séquence et expose à une variabilité importante (notamment sur des jeux de données difficiles comme Adream).

Une autre manière de procéder est de calculer la moyenne des signatures de toutes les images d'une tracklet pour obtenir sa représentation ce qui a l'effet inverse : les informations spécifiques à chaque image (pose, angle de vue, etc.) sont diluées dans une représentation unique et lissée.

Aucune de ces deux approches n'exploite correctement la variabilité temporelle propre à une tracklet.

L'idée qui a été expérimentée consiste donc à appliquer un clustering k-means sur les signatures extraites d'une tracklet, afin de la représenter non plus par une seule signature globale, mais par k signatures, une par cluster. Chaque cluster est ensuite résumé par la moyenne des signatures qui le composent. Cette moyenne locale, calculée sur un sous-ensemble d'images par défintion plus homogène, préserve mieux les spécificités de chaque moment de la tracklet, moments qu'on peut exploiter individuellement.

Ce découpage fait émerger naturellement des regroupements liés à des variations d'apparence caractéristiques (par exemple : cluster 1 – personne de face, cluster 2 – personne de profil, cluster 3 – personne de dos), afin d'augmenter les chances de reconnaître une même personne dans une autre tracklet lorsqu'elle y apparaît sous une posture similaire. Ainsi, deux caméras avec des angles de vues différents peuvent avoir, à un moment donné dans la tracklets, des prises de vues similaires. Ce sont ces prises de vues qui sont ciblées avec cette méthodes.

Ce changement de représentation a une conséquence directe sur le calcul du score de similarité entre deux tracklets : on ne compare plus une signature unique à une autre, mais l'ensemble des k signatures de la première tracklet à l'ensemble des k signatures de la seconde. Pour $k = 5$, cela donne une matrice de similarité 5×5, où chaque case représente la similarité entre un cluster de la tracklet A et un cluster de la tracklet B (table 4).

	Cluster B_1	Cluster B_2	Cluster B_3	Cluster B_4	Cluster B_5
Cluster A_1	0.64	0.03	0.28	0.22	0.74
Cluster A_2	0.68	0.89	0.09	0.42	0.03
Cluster A_3	0.22	0.51	0.03	0.20	0.65
Cluster A_4	0.54	0.22	0.59	0.81	0.01
Cluster A_5	0.81	0.70	0.34	0.16	0.96

TABLE 4 – Exemple de matrice de similarité entre les clusters d'une tracklet A (lignes) et d'une tracklet B (colonnes), pour $k = 5$.

Il reste alors à définir comment réduire cette matrice à une unique valeur représentative de la similarité globale entre les deux tracklets. C'est pourquoi il a été mis en place plusieurs méthode d'extraction pour essayer de trouver la plus représentative :

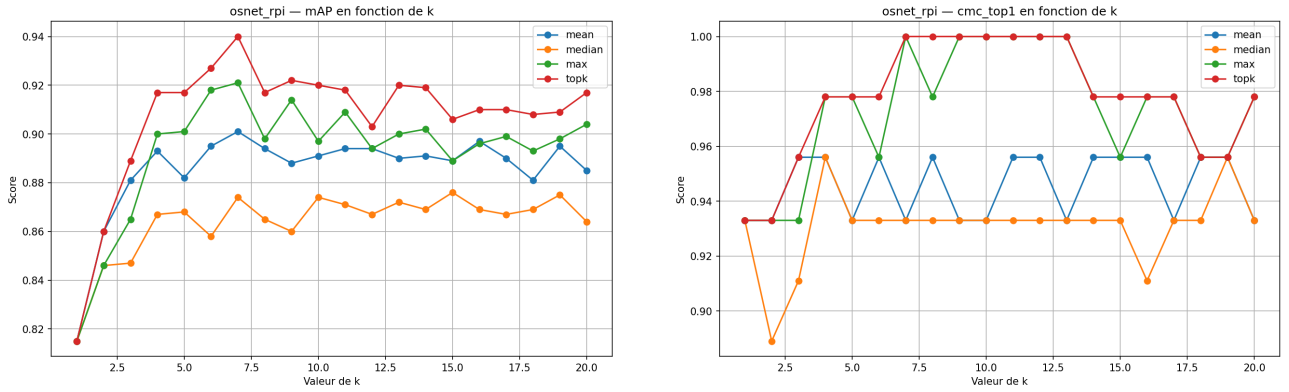
— **Le maximum** : privilégier la paire de clusters la plus similaire, dans l'idée qu'une correspondance forte entre deux clusters suffit à indiquer une même identité. Ce choix comporte toutefois un

risque : une similarité élevée peut aussi provenir de deux mauvaises détections partageant un fond commun plutôt que d'une réelle correspondance de pose.

— **La moyenne** : prendre la moyenne de l'ensemble des valeurs de la matrice, pour obtenir une mesure globale plus stable, au prix d'une possible dilution de l'information si peu de clusters se correspondent réellement.

— **La médiane** : une alternative à la moyenne, plus robuste au bruit introduit par des clusters mal formés ou des détections erronées.

— **La moyenne des 5 plus grandes valeurs** : cette méthode cherche à limiter la sensibilité du maximum au bruit tout en conservant l'essentiel du signal porté par les correspondances les plus fortes.



(a) mAP (b) CMC Rank 1

FIGURE 11 – Evolution des métriques en fonction du nombre de cluster. OSNET_ain sur le dataset RPIfield.

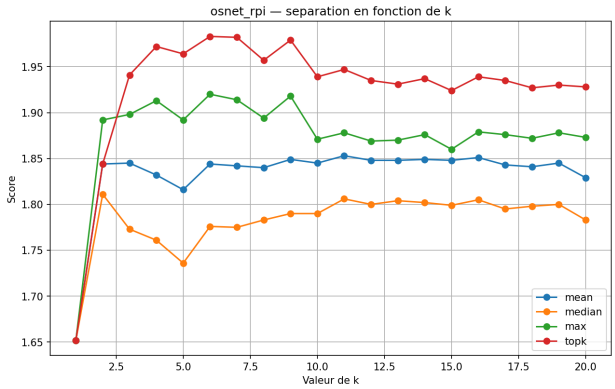


FIGURE 12 – Evolution de la séparation des courbes en fonction du nombre de cluster. OSNET_ain sur le dataset RPIfield.

Les courbes de la figure 16 et 12 montrent l'évolution du mAP, de l'indice CMC Rank 1 ainsi que de la séparation en fonction du nombre de clusters k utilisés pour représenter chaque tracklet. Chaque méthode est représentée par une couleur différente. Le cas $k = 1$ correspond au cas dégénéré où la tracklet est représentée par un unique cluster, c'est-à-dire par la moyenne de l'ensemble de ses images. Dès que k augmente, on observe une amélioration nette et rapide des trois métriques, quelle que soit la stratégie considérée. Cette progression confirme l'intérêt du découpage en clusters : décomposer la tracklet en plusieurs sous-représentations permet effectivement de mieux exploiter sa variabilité temporelle qu'une architecture pensée pour l'image seule. L'aspect étagé de la courbe CMC Rank 1 est symptomatique du nombre d'erreur sur le nombre de requête (1 erreur donne 97.875 Au-delà de ce

gain initial, les courbes présentent une nette tendance à l'asymptote : passé $k \approx 7$–10, les performances cessent de progresser significativement et oscillent dans une plage stable (mAP autour de 0,90–0,92 pour *topk* et *max*, autour de 0,88–0,90 pour *mean*), avant même de très légèrement décroître pour les valeurs de k les plus élevées, notamment sur le CMC Rank 1 et la similarité au niveau des stratégies *max* et *topk* au-delà de $k = 13$. Cette saturation suggère qu'au-delà d'un certain nombre de clusters, le découpage n'apporte plus d'information nouvelle et commence même à fragmenter excessivement la tracklet, chaque cluster devenant moins robuste statistiquement (moins d'images par cluster) sans gain de discrimination supplémentaire.

On note également une hiérarchie stable entre les stratégies d'agrégation, cohérente avec l'analyse qualitative proposée précédemment : *topk* et *max* dominent systématiquement *mean*, elle-même au-dessus de *median*, qui reste la stratégie la moins performante et la plus instable (visible notamment sur le CMC Rank 1, avec des chutes marquées à $k = 2$ et $k = 16$). Ce résultat va dans le sens attendu : privilégier les correspondances de clusters les plus fortes (*topk*, *max*) capture mieux les moments où deux tracklets partagent une pose ou un angle de vue très similaire, tandis que la médiane, en lissant fortement le signal, tend à masquer ces correspondances ponctuelles pourtant discriminantes pour la ré-identification.

4.3.3 Maximisation du seuil et renormalisation en fonction du profil de la tracklet

L'objectif de cette étape est de réaliser l'attribution d'une identité à chaque tracklet pour tenter d'améliorer le processus de génération de signatures. L'approche la plus simple consiste à définir un seuil minimal de similarité au-delà duquel deux tracklets sont considérées comme appartenant à la même identité, ce seuil étant choisi de façon à minimiser l'erreur de classification selon les indices de mesure classiques (F1-score, rappel, précision, etc.). L'observation de ces métriques différentes a permis de déceler une nouvelle manière d'approcher l'extraction de signatures.

En effet, au-delà d'un seuil global unique définit par avance, il est possible d'améliorer encore ces scores en renormalisant chaque tracklet individuellement, en fonction de son propre profil de similarité plutôt que d'un seuil fixe appliqué uniformément à l'ensemble des données.

Cette adaptation du seuil se déroule en quatre étapes :

1. **Extraction des similarités** : pour chaque tracklet requête, on calcule sa similarité avec l'ensemble des tracklets de la galerie.
2. **Définition d'un score moyen** : on établit un seuil global au-dessus duquel deux tracklets peuvent être considérées comme provenant de la même identité.
3. **Adaptation du seuil par tracklet** : ce seuil global est ensuite ajusté spécifiquement pour chaque tracklet, en fonction de la dispersion des scores de similarité autour de cette valeur. L'objectif est de déterminer, lorsqu'elle existe, une frontière nette entre les tracklets réellement très similaires et les autres.
4. **Évaluation statistique** : les scores statistiques (F1-score, rappel, précision, etc... voir en annexe .2) sont mesurés pour chaque tracklet, seuil adapté à l'appui.

Cette méthode présente l'avantage d'être reproductible et non biaisée : elle s'applique de la même manière à toutes les tracklets, sans connaissance préalable du résultat attendu, contrairement à un seuil fixé arbitrairement a priori. A noter que le choix de ce seuil influe, comme décrit ci-dessous, sur la performance de la méthode.

Les graphiques suivants comparent, pour l'ensemble des seuils moyens envisageables à l'étape 2, les performances obtenues avec et sans adaptation du seuil par tracklet.

On observe que, lorsque le seuil global initial est bien choisi, l'adaptation par tracklet permet une petite modification des résultats (courbes en pointillé). Il se trouve que le f1-score augmente sensiblement ($0.65 => 0.70$) autour du seuil optimal. Cette amélioration n'est toutefois pas observée sur Adream, trop restreint en nombre de valeurs dont les scores de similarités se trouvent être trop dispersés.

Il est donc possible avec cette procédure d'améliorer les performances de classification du modèle, ce qui est une manière d'améliore la qualité des ensembles de signatures générées. Ainsi, c'est une nouvelle manière d'élargir le gradient de difficulté des signatures générées.

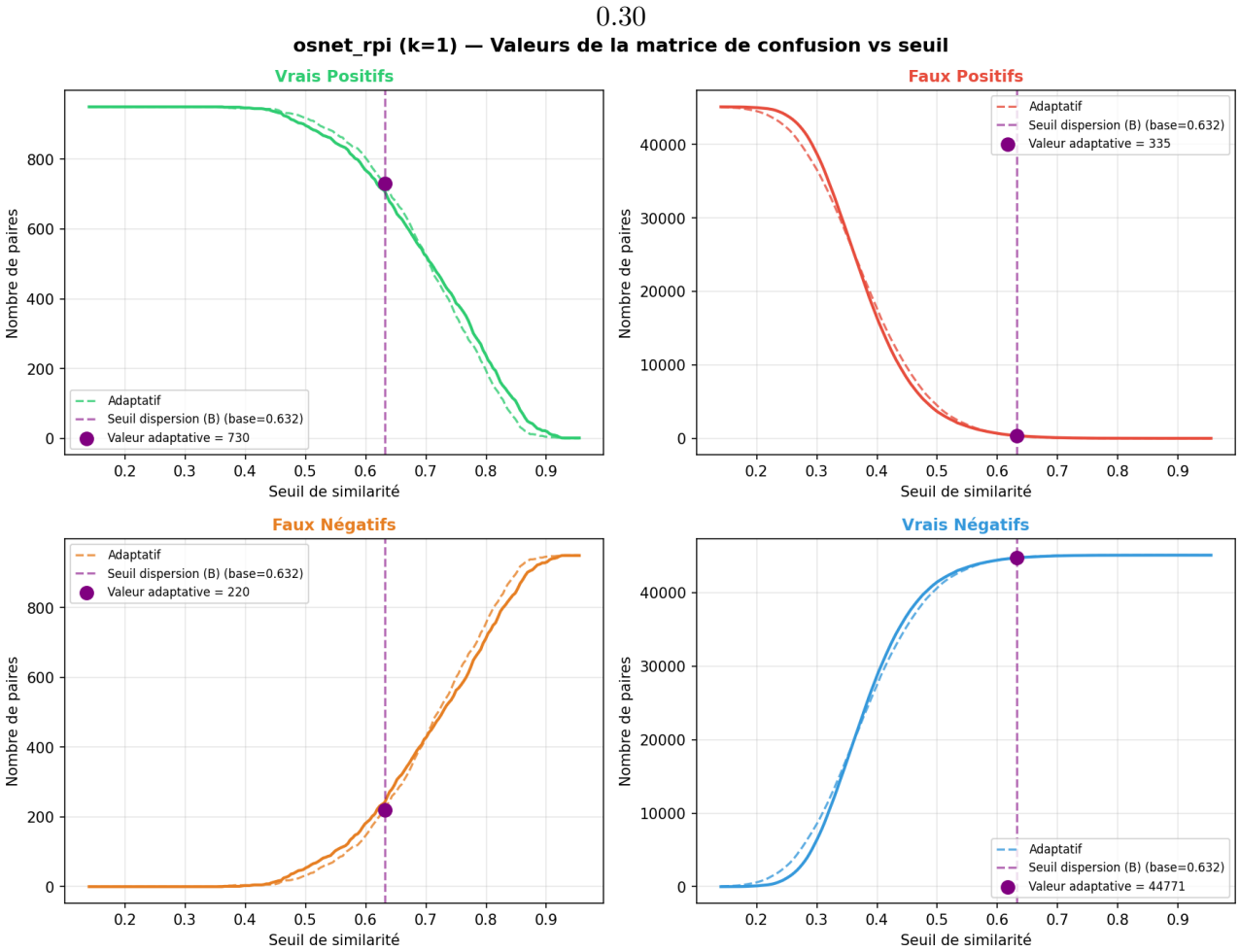


FIGURE 13 – Performances en fonction du seuil

4.4 Modèle basé sur CLIP

En utilisant un modèle basé sur CLIP, modèle beaucoup plus général puisqu'il a bénéficié de toutes les avancées récentes en matière de représentation visuelle et textuelle, X-TFCLIP obtient d'excellents résultats sur les jeux de données RPIfield et Adream, alors même qu'il a été entraîné sur des données différentes (AG-VPReID). Le tableau 5 synthétise ces résultats, on peut noter un score de mAP de 83% sur RPIfield, score qui n'a jamais été atteint sans réentraînement sur OSNET$_ain$.

Jeu de données	mAP	R1
RPIfield	83	91
Adream V1	83	84
Adream V2	73	77

TABLE 5 – Résultats de X-TFCLIP (mAP et R1, en %) sur RPIfield, Adream V1 et Adream V2 réannoté.

FIGURE 14 – Représentation visuelle du classement des tracklets par score de similarité

Les résultats restent globalement élevés sur l'ensemble des configurations, malgré l'absence de tout entraînement spécifique sur RPIfield ou Adream. On peut aussi noter une séparation bien plus élevée (de 1.30 à 2.35, +65 %) On note toutefois une performance sensiblement plus faible et plus instable sur Adream V2 à 90 % (mAP de 73 %), cohérente avec la difficulté intrinsèque de ce jeu de données déjà évoquée précédemment (faible nombre d'identités, ambiguïtés visuelles, identités à apparition unique).

4.5 Discussion

Ces résultats posent une difficulté méthodologique de fond : il est difficile de surpasser un modèle aussi général que X-TFCLIP, dont les performances reposent sur un pré-entraînement effectué sur d'énormes quantités de données. Un fine-tuning sur nos propres jeux de données pourrait certes améliorer ses performances de manière ciblée, mais une telle opération nécessite elle-même une quantité de données conséquente pour être efficace — quantité de données qui fait justement défaut sur des jeux de données comme Adream, et qui est par ailleurs nécessaire pour alimenter les algorithmes de Recherche Opérationnelle en aval.

C'est pourquoi la suite de ce travail se concentre non pas sur un réentraînement massif du modèle, mais sur une meilleure exploitation des données disponibles, dans le but d'améliorer, ou à défaut de conserver, les performances de X-TFCLIP tout en minimisant la quantité de données prélevée sur les jeux de données utilisés.

5 Amélioration par sélection d'identité

5.1 Zoom sur X-TFCLIP

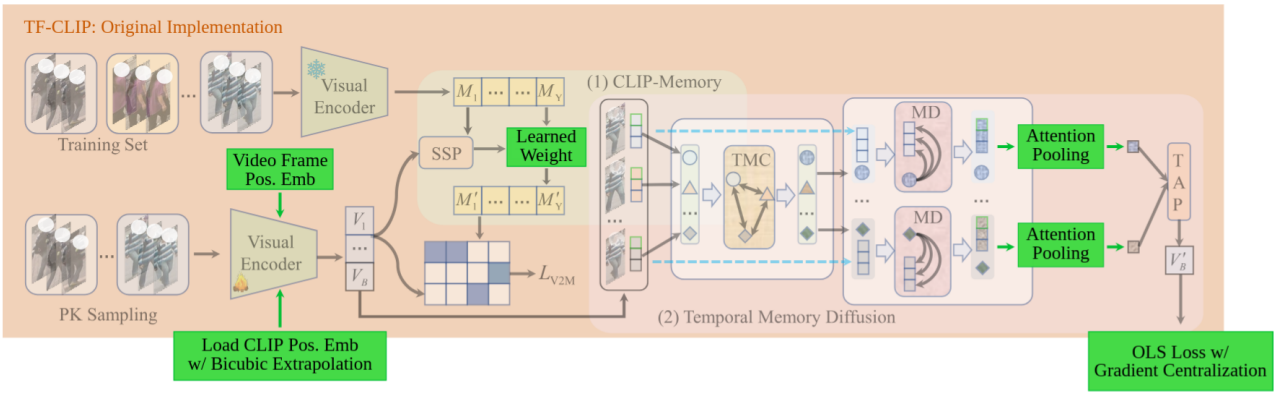


FIGURE 15 – Architecture du modèle X-TFCLIP

X-TFCLIP [15] est un modèle créé dans le cadre d'une compétition de ré-identification de personnes fondée sur des séquences vidéo aériennes, la AG-VPReID 2025 Challenge. Cette compétition a pour sujet la ré-identification à partir de vues aériennes prises à haute altitude, comprise entre 80 et 120 mètres. Elle s'appuie sur le jeu de données AG-VPReID capturées à l'aide de drones (UAV), de caméras de vidéosurveillance (CCTV) et de caméras portées (wearable cameras). La compétition a réuni quatre équipes internationales et vu l'équipe qui a créé X-TFCLIP l'emporter. Le modèle créé est un modèle de ReID vidéo, c'est à dire qui prends en paramètre l'entiereté d'une tracklet pour établir une signature, et son principe de fonctionnement est basé sur le modèle de constrastive learning CLIP.

5.1.1 Principe de CLIP et de TF-CLIP

Appliqué à la ReID, le principe de CLIP pose un problème important : l'absence de description textuelle associée aux identités ; or le module est fait pour aligner ces représentations visuelles et textuelles et les faire correspondre.

TF-CLIP répond à cette contrainte en proposant une variante "text-free" de CLIP pour la ReID vidéo : l'encodeur de texte n'est plus utilisé pour générer des descriptions linguistiques, mais remplacé par une mémoire visuelle qui joue un rôle fonctionnellement analogue.

Le modèle construit ainsi des mémoires des tracklets avec un modèle de CLIP préentraîné. Autrement dit, comme les modules visuels et textuels de CLIP renvoient des représentations qui partagent le même espace, on utilise un module préentraîné de CLIP visuel que l'on gèle pour remplacer le mudule de CLIP textuel, dans le but de conserver une représentation fine de la réalité (proche de l'entraînement) tout en spécifiant cette représentation au problème de ReID.

TF-CLIP conserve donc l'architecture à deux branches de CLIP, mais la seconde branche, initialement textuelle, devient une seconde branche visuelle spécialisée dans la modélisation temporelle.

5.1.2 Modules d'attention spatiaux et temporels

Le mécanisme d'attention, popularisé par l'architecture Transformer, repose sur le principe suivant : plutôt que de traiter chaque élément d'une séquence indépendamment ou séquentiellement, le modèle apprend à pondérer dynamiquement l'importance relative de chaque élément par rapport aux autres. Cette pondération permet au réseau de capturer des dépendances entre les différents patchs de l'image, indépendamment de la distance spatiale ou temporelle entre eux.

Lors de l'inférence de X-TFCLIP (cf Figure 15), il y a deux modules d'attention différents, le premier est temporel, c'est à dire que les différents patchs vont être mis en relation directement avec

leurs versions postérieures et antérieures. Autrement dit, on regarde l'évolution des différents patchs dans le temps pour adapter la représentation de chaque patch en fonction de ce qui se produit à un même endroit de l'image.

Le deuxième module d'attention est plus classique, on reprends les patchs au représentations modifiées par le premier module d'attention et on les mets en relation aves les autres patchs de la même temporalité qu'eux (i.e. au sein d'une même image). Dans ce module, les représentations de chaque patchs sont donc modifiés en fonction de ce qui est présent à d'autres endroits de l'image.

Le modèle apprend de telle manière à ce que la représentation finale après l'aggrégation des informations (ici par Time-Avererage Pooling), soit disctiminante d'une identité. Il aura pris en compte toutes les images, et chaque image dans son entiereté, pour générer la signature finale correspondant à la tracklet.

5.1.3 Améliorations apportées par X-TFCLIP

X-TFCLIP est une extension de TF-CLIP. Les contributions de X-TFCLIP se regroupent en trois axes principaux : la sélection de signatures par attention, des modifications architecturales du backbone CLIP, et une fonction de perte révisée, auxquelles s'ajoutent des ajustements à l'inférence.

Le redimensionnement des embeddings de position du ViT est affiné pour améliorer la qualité des représentations spatiales face aux variations de résolution, tandis que l'agrégation temporelle simple des signatures est remplacée par un mécanisme apprenant à pondérer les frames les plus discriminantes.

La robustesse de l'apprentissage est renforcée par un adoucissement des distributions cibles de la cross-entropie, ainsi que par l'ajout d'une information de position au niveau de chaque frame, combinée aux embeddings de caméra, pour mieux capturer le contexte temporel et de point de vue.

Sur le plan architectural, la combinaison entre mémoire CLIP et signatures projetées est rendue adaptative grâce à un paramètre de pondération apprenable, et la normalisation du module BNNeck passe d'une approche par batch à une approche par instance afin de mieux gérer les variations d'apparence propres à la ReID. Ces différentes améliorations, combinées, ont permis à X-TFCLIP de dépasser TF-CLIP sur l'ensemble des métriques évaluées lors du challenge AG-VPReID 2025.

En raison de sa modernité et de ses performances dépassant les autres modèles existant, au moins sur AG-VPREID, nous avons donc essayé de l'utiliser dans le cadre de nos expérimentations sur les jeux de données que nous avons sélectionnés.

5.2 Objectifs des méthodes

Partant du constat qu'il est nécessaire d'obtenir les meilleurs résultats possibles sans utiliser les jeux de données Adream pour l'entraînement, il devient nécessaire d'envisager d'autres manières d'orienter le modèle vers un espace de représentation bénéfique à la ré-identification sur Adream. Au-delà des 83 % de mAP obtenus sur Adream V1 et des 74 % obtenus sur Adream V2 par X-TFCLIP sans aucun réentraînement (cf. partie précédente), il est instructif de considérer les gains obtenus lors d'un réentraînement direct sur ces jeux de données (table 6).

Dataset d'entraînement	**Adream V1 (30 %)**	**V1 (90 %)**	**V2 (30 %)**	**V2 (90 %)**
Adream V1	91	– – –	92	77
Adream V2	90	88	93	– – –

TABLE 6 – Résultats de X-TFCLIP (mAP, en %) sur Adream V1 et Adream V2 réannoté, selon le dataset utilisé pour le réentraînement et la proportion de données disponibles à l'évaluation.

Dans chaque cas, l'évaluation est réalisée uniquement sur la partie du jeu de données qui n'a pas servi à l'entraînement (colonnes grisées : entraînement sur Adream V1 évalué sur Adream V1, respectivement entraînement sur Adream V2 évalué sur Adream V2), afin d'éviter toute fuite de

données entre entraînement et évaluation. On observe une amélioration systématique des performances à chaque réentraînement, que ce soit lorsqu'un modèle est réentraîné et évalué sur un même jeu de données, ou lorsqu'il est réentraîné sur un jeu de données puis évalué sur l'autre (Adream V1 → V2 ou Adream V2 → V1). Ce dernier point est particulièrement intéressant : il suggère qu'un réentraînement sur un jeu de données *similaire* mais distinct de la cible profite malgré tout à cette dernière, ouvrant la voie à une exploitation indirecte des données Adream sans les utiliser directement pour l'entraînement du modèle final.

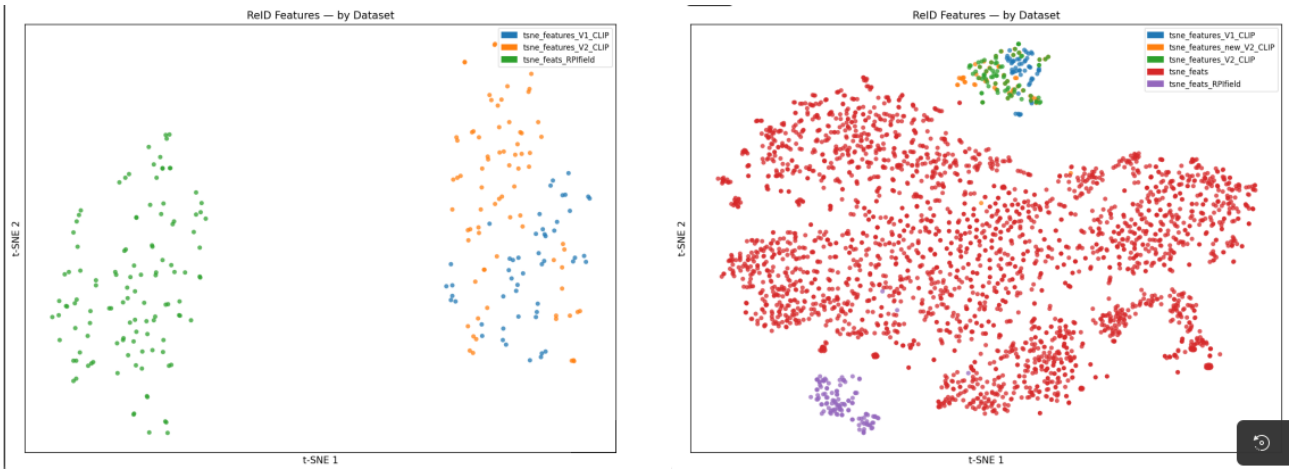
Cette observation motive l'approche suivante : utiliser l'ensemble des détection de Adream V1 comme un ensemble non annoté, non pas pour l'entraîner directement, mais pour déterminer son entourage au sein d'autres datasets annotés disponibles. L'objectif est de cerner, dans l'espace de représentation de ces datasets, la zone bénéfique à la ré-identification sur Adream, afin d'en extraire les informations utiles à l'amélioration du modèle sans jamais exposer celui-ci aux annotations d'Adream lui-même.

Reste alors à répondre à une question centrale pour la suite de ce travail : qu'entend-on précisément par entourage, et comment le cerner efficacement ?

5.3 Visualisation et Espaces de choix

Il apparaît alors rapidement nécessaire de disposer d'un critère objectif de distance entre les différents jeux de données, afin d'en déduire lesquels sont pertinents à exploiter. Pour cela, il faut convertir chaque tracklet en une signature encodant ses spécificités. C'est heureux puisque c'est précisément ce que réalisent le module CLIP ainsi que l'algorithme X-TFCLIP. Il a donc été décidé de les utiliser pour choisir les identités provenant des autres jeux de données à intégrer pour le réentraînement des modèles.

La figure 16a propose une représentation en deux dimensions des datasets Adream et RPIfield, à partir des signatures produites par le module CLIP, réduites de 2048 à 2 dimensions à l'aide de l'algorithme t-SNE [16]. L'algorithme t-SNE est une technique de réduction de dimension non linéaire qui cherche à préserver les relations de voisinage locales entre points dans l'espace de faible dimension. On cherche alors des voisinages aux jeusx de données Adream pour en extraire des identités intéressantes



(a) Adream RPIfield (b) Adream RPIfield AG-VPREID (en rouge)

FIGURE 16 – Illustration du phénomène de séparation des signatures dans l'espace latent

On observe sur ce graphique que les différents jeux de données Adream se chevauchent et apparaissent très proches les uns des autres, tandis que le dataset RPIfield se distingue nettement, plus à l'écart. On peut alors supposer que l'amélioration des résultats de Adream V1 grâce au réentraînement sur Adream V2 et inversement pourrait être la conséquence de leur proximité. Dans cette optique chercher à identifier des datasets supplémentaires qui seraient également proches pourrait définir une méthode

de réentraînement efficace et sans consommation de données propre à Adream.

Cette piste s'est toutefois révélée infructueuse : en utilisant les signatures CLIP brutes, tous les datasets restent trop marqués par leur environnement de capture respectif (éclairage, angle de caméra, arrière-plan) pour que leurs représentations se chevauchent significativement, quel que soit le dataset ajouté (figure 16b).

Le même constat peut être fait pour les signatures produites en sortie de X-TFCLIP, qui, en plus du fait de se regrouper par identité, marque malheureusement aussi une tendance à se regrouper par jeu de données, comme illustré en figure 17. Cela peut être expliqué par les modules d'attention qui sont entraînés à mettre en relation l'ensemble de la tracklet y compris ce qui n'appartient pas à une personne, un environnement qui peut varier d'un jeu de données à un autre.

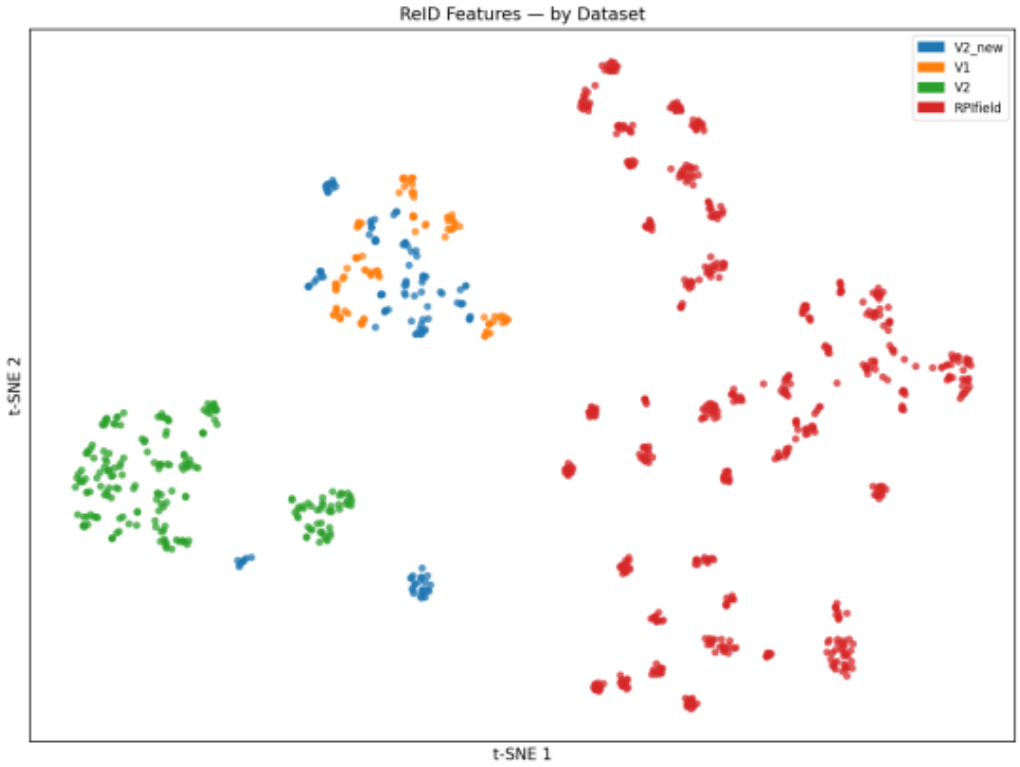


FIGURE 17 – Projection t-SNE des signatures en sortie de X-TFCLIP

Il existe cependant un troisième endroit du modèle où une image est transformée en signature : l'encodeur visuel utilisé pendant l'inférence. Celui-ci correspond à une version de CLIP modifiée par l'entraînement (ses poids sont apprennable), mais on peut raisonnablement estimer que le modèle entraîné sur AG-VPReID, capable de reconnaître des personnes de manière satisfaisante, a développé une certaine compréhension des images et, plus encore, des identités qu'elles représentent. Utiliser cet ensemble de poids semble alors justifié pour cerner les différents jeux de données et leur entourage.

L'idée sous-jacente est la suivante : si le modèle traite deux images de manière quasi identique, c'est qu'il les considère comme proches, même si visuellement elle ne le sont pas. Toujours est-il que si le modèle trouve ces tracklets proches lors de l'encodage, on peut alors substituer une image d'Adream V1 par une image proche issue d'un autre dataset annoté. C'est pour cette raison que les représentations issues de cet encodeur ont été retenues comme méthode pour déterminer les identités à ajouter aux données d'apprentissage.

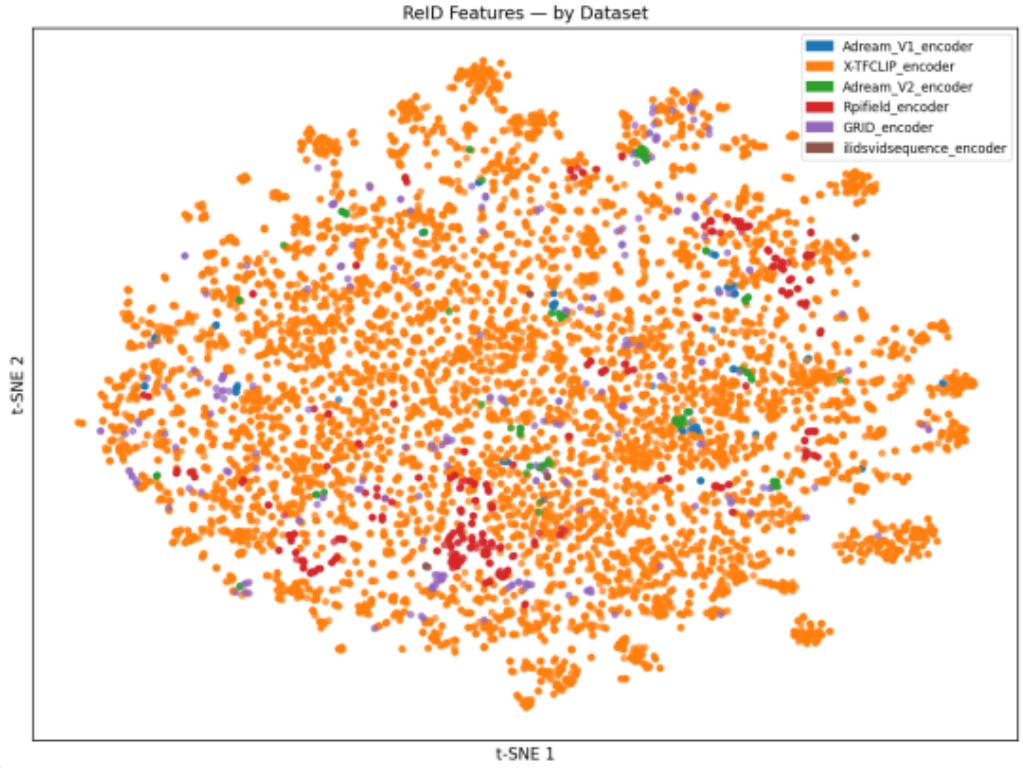


FIGURE 18 – Projection t-SNE des signatures issues de l'encodeur visuel entraîné.

On observe sur ce dernier graphique un recouvrement nettement plus marqué entre les différents jeux de données, ouvrant ainsi la voie à une substitution potentielle des identités d'Adream par les identités jugées les plus similaires dans les autres datasets annotés.

5.4 Méthodes de sélection et Résultats

Une fois ce recouvrement partiel identifié entre les représentations issues de l'encodeur visuel entraîné, il reste à définir de manière cohérente quelles tracklets utiliser pendant l'entraînement.

Ce choix a été effectué au niveau de la tracklet, selon la procédure suivante : dans un premier temps, toutes les tracklets respectant les contraintes de la méthode de sélection considérée sont identifiées.

Dans un second temps, puisque l'entraînement des appariements nécessite de disposer de plusieurs tracklets pour une même identité, l'ensemble des tracklets correspondant aux identités ainsi retenues est sélectionné, et non les seules tracklets ayant initialement satisfait le critère. Une alternative aurait consisté à sélectionner les identités dont la représentation moyenne est la plus proche du centre du dataset Adream ; ce choix a cependant été écarté au profit d'une sélection fondée sur la proximité individuelle des tracklets avec le dataset cible, jugée plus fidèle à l'objectif recherché.

Trois familles de méthodes ont été explorées pour définir ce critère de proximité.

Algorithme du plus proche voisin Pour chaque tracklet de Adream, l'unique tracklet la plus proche appartenant à un autre jeu de données est sélectionnée. L'objectif de cette approche est avant tout de constituer un ensemble d'identités et de tracklets d'une taille comparable à celle d'Adream V1, afin de pouvoir comparer directement les performances obtenues sans que la quantité de données disponibles à l'entraînement n'introduise de biais. Le modèle a ici été fine-tuné à partir de l'ensemble des jeux de donnéets disponibles (GRID, RPIfield, ENTIRe-ID, DukeMTMC-reID). Le nouveau dataset contient 29 identités provenant de ces 4 jeux de données. Les résultats sont présentés en table 7.

Adream V2 (30 %)	V2 (90 %)	V1 (30 %)	V1 (90 %)
77 / 74	50 / 46	73 / 76	54 / 53

TABLE 7 – Résultats (mAP / R1, en %) de la sélection par plus proche voisin, fine-tuning sur l'ensemble des datasets avec équilibrage des classes.

Algorithme par rayon (distance euclidienne) Cette méthode sélectionne l'ensemble des tracklets présentes dans un rayon donné autour des tracklets d'Adream. Est appelé rayon ici, la valeur de la distance euclidienne entre les signatures de dimension 2048. On récupère donc les identités correspondantes au tracklets présentes dans ces hypersphères autour des signatures de Adream pour constituer le nouveau dataset d'entraînement. Plusieurs valeurs de rayon ont été testées : $R = 0{,}35$, 0,50, 0,75 et 1,00. Les résultats sont présentés en table 8.

Rayon (nb. identités)	V2 (30 %)	V2 (90 %)	V1 (30 %)	V1 (90 %)
Rayon = 0,35 (33)	40/37	37/31	—	—
Rayon = 0,50 (330)	83/91	56/63	71/76	66/68
Rayon = 0,75 (1452)	87/97	60/62	84/83	79/84
Rayon = 1,00 (1544)	88/94	62/62	86/83	80/81

TABLE 8 – Résultats (mAP / R1, en %) de la sélection par rayon, selon la valeur du rayon et le nombre de tracklets sélectionnées. Datasets pris en compte : GRID, RPIfield, ENTIRe-ID, DukeMTMC-reID.

Algorithme par distance ajustée Ces méthodes constituent des variantes de la sélection par distance euclidienne, dans lesquelles la distance minimale requise pour retenir une tracklet n'est plus un seuil fixe unique, mais un seuil ajusté en fonction de statistiques locales calculées autour du jeu de données Adream ($R = 0.75$). Trois variantes ont été comparées :

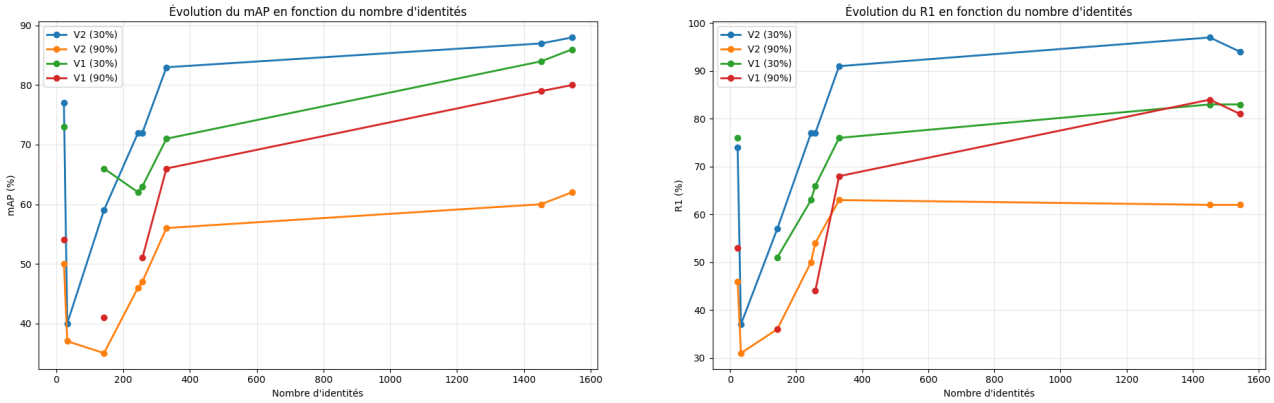
Tout d'abord, le Z-score qui normalise les distances localement en fonction de son entourage, le but étant de ne pas privilégier dans le nouveau jeu de données uniquement les zones denses. Une tracklet n'est retenue que si sa distance normalisée (son z-score) est inférieure à un seuil donné, ce qui revient à ne conserver que les correspondances significativement plus proches que la distribution locale des distances, indépendamment de l'échelle absolue de ces dernières.

Ensuite la sélection peut se faire par seuil adaptatif : le seuil de référence R est ajusté individuellement pour chaque tracklet requête en fonction de la dispersion des distances observées autour de lui. Cela permet de compenser les tracklets pour lesquels la distribution des distances est globalement plus resserrée ou plus étalée que la moyenne. Cela permet de trouver une rupture entre les tracklets proche et celle plus lointaine afin de modifier ce rayon aux alentours de cette limite.

Enfin, la proximité mutuelle fait en sorte qu'une paire de tracklets n'est retenue que si chacune des deux figure parmi les plus proches voisines de l'autre (critère de réciprocité). Cette contrainte de symétrie vise à limiter la tendance de certaines tracklets à apparaître comme proches voisines d'un grand nombre d'autres tracklets sans que la réciproque soit vraie, ce qui peut fausser la sélection.

Méthode (nb. identités)	Adream V2 (30 %)	V2 (90 %)	V1 (30 %)	V1 (90 %)
Z-score (245)	72 / 77	46 / 50	62 / 63	—
Seuil adaptatif (258)	72 / 77	47 / 54	63 / 66	51 / 44
Proximité mutuelle (143)	59 / 57	35 / 36	66 / 56	41 / 36

TABLE 9 – Résultats (mAP / R1, en %) des méthodes de sélection statistiques, fine-tuning à partir des poids AG-VPReID.



(a) mAP

(b) R1

FIGURE 19 – Performances des modèles en fonction du nombre d'identité du dataset créé, fine-tuning à partir des poids AG-VPReID.

Comme on peut l'observer, aucun de ces modèles n'arrive à s'approcher des performances trouvées par réentraînement sur Adream. Mais pire encore, aucun de ces modèles n'arrive à améliorer le modèle de base entraîné sur AG-VPREID.

Le manque de temps de ce stage n'a malheureusement pas permi de trouver une meilleure manière de réentraîner le modèle X-TFCLIP.

Il est cependant utile de remarquer une légère correlation entre le nombre d'identité choisies pour les entraîner et la peformance des modèles comme montré sur la figure 19.

6 Conclusion et perspectives

Ce stage s'inscrivait dans une problématique double : d'une part, comprendre et mobiliser l'état de l'art de la ré-identification de personnes pour l'adapter à des jeux de données vidéo horodatés et de petite taille ; d'autre part, fournir à l'algorithme de Recherche Opérationnelle développé dans le cadre de la thèse de Cyrille Equoy des signatures suffisamment discriminantes et fiables pour construire un graphe de correspondances cohérent à l'échelle d'un réseau de caméras épars.

Le travail mené a permis de répondre progressivement à cette problématique. La mise en place d'un cadre d'expérimentationn, ré-annotation d'Adream V2, méthode d'évaluation indépendante des modèles, a d'abord été nécessaire pour garantir la fiabilité des comparaisons ultérieures. La prise en compte de la temporalité des tracklets (clustering k-means, seuils adaptatifs) a ensuite démontré qu'il est possible d'améliorer sensiblement les performances d'une architecture image comme OSNet en exploitant mieux la variabilité intrinsèque d'une séquence, sans toutefois atteindre les performances d'un modèle vidéo natif.

L'évaluation de X-TFCLIP, pourtant entraîné sur un jeu de données très différent (AG-VPReID), a indiscutablement surpassé largement toutes les approches précédemment testées sur les jeux de données sélectionnés, y compris après réentraînement d'OSNet sur ces données. Cette observation a réorienté la seconde partie du stage vers une question supplémentaire : comment exploiter au mieux les données disponibles sans consommer des données pour le réentraînement ?

Les méthodes de sélection d'identités explorées n'ont pu apporter qu'une réponse partielle, sans parvenir à dépasser les performances du réentraînement direct sur Adream.

Sur le plan personnel, ce stage a été l'occasion de me confronter à un travail de recherche complet, depuis la constitution et l'annotation d'un jeu de données jusqu'à l'expérimentation et l'analyse critique de résultats parfois contre-intuitifs. Il m'a permis d'approfondir ma compréhension des réseaux de neurones et me confronter, une nouvelle fois à leur complexité. Cette expérience au sein du LAAS-CNRS, encadrée par une équipe de recherche active, conforte mon intérêt pour la robotique et m'encourage à continuer dans cette voie.

Table of Figures

Annexes

.1 Exemple d'utilisation des métriques

Considérons une requête correspondant à un individu donné, pour laquelle la galerie contient 7 images candidates. Parmi ces 7 images, 3 appartiennent effectivement à la même identité que la requête et constituent donc les images pertinentes. Les quatre autres images correspondent à des identités différentes.

Pour illustrer le calcul des métriques, notons les trois images pertinentes G_1, G_2 et G_3, et les quatre images non pertinentes I_1, I_2, I_3 et I_4.

.1.1 Cas idéal : AP = 100 % et CMC Rank-1 = 100 %

Supposons que le système retourne les images dans l'ordre suivant :

$$[G_1, G_2, G_3, I_1, I_2, I_3, I_4] \tag{5}$$

Les trois images pertinentes apparaissent donc respectivement aux rangs 1, 2 et 3. La précision aux différentes positions pertinentes est alors :

$$P(1) = \frac{1}{1} = 1 \tag{6}$$

$$P(2) = \frac{2}{2} = 1 \tag{7}$$

$$P(3) = \frac{3}{3} = 1 \tag{8}$$

L'Average Precision (AP) pour cette requête vaut donc :

$$AP_q = \frac{1}{3}\left(P(1) + P(2) + P(3)\right) \tag{9}$$

$$AP_q = \frac{1}{3}(1 + 1 + 1) = 1 \tag{10}$$

soit :

$$\boxed{AP_q = 100\,\%} \tag{11}$$

Dans cet exemple, la première image retournée est également pertinente. Le rang de la première correspondance correcte est donc :

$$rank_q = 1 \tag{12}$$

Ainsi, pour le CMC à Rank-1 (c'est à dire la fréquence pour laquelle l'image de rang 1 est effectivement une image de la même personne que la requête):

$$CMC(1) = \frac{1}{1}\mathbf{1}[rank_q \leq 1] \tag{13}$$

$$CMC(1) = 1 \tag{14}$$

soit :

$$\boxed{CMC@1 = 100\,\%} \tag{15}$$

Cet exemple correspond donc à une situation parfaite : les trois images de la bonne identité sont classées avant toutes les images non pertinentes, et la première image retournée est correcte.

.1.2 Cas défavorable : AP minimal et CMC Rank-1 = 0 %

À l'inverse, considérons le classement suivant :

$$[I_1, I_2, I_3, I_4, G_1, G_2, G_3] \tag{16}$$

Les trois images pertinentes sont maintenant placées aux trois dernières positions. Les précisions calculées aux positions pertinentes sont alors :

$$P(5) = \frac{1}{5} = 0{,}2 \tag{17}$$

$$P(6) = \frac{2}{6} = \frac{1}{3} \tag{18}$$

$$P(7) = \frac{3}{7} \approx 0{,}4286 \tag{19}$$

L'Average Precision est donc :

$$AP_q = \frac{1}{3}\left(\frac{1}{5} + \frac{2}{6} + \frac{3}{7}\right) \tag{20}$$

$$AP_q = \frac{1}{3}\left(0{,}2 + 0{,}3333 + 0{,}4286\right) \approx 0{,}3206 \tag{21}$$

soit :

$$\boxed{AP_q \approx 32{,}06\,\%} \tag{22}$$

Le CMC à Rank-1 est quant à lui nul puisque la première image retournée n'est pas pertinente. Le rang de la première correspondance correcte est ici :

$$rank_q = 5 \tag{23}$$

Par conséquent :

$$CMC(1) = \mathbf{1}[rank_q \leq 1] \tag{24}$$

$$CMC(1) = \mathbf{1}[5 \leq 1] = 0 \tag{25}$$

soit :

$$\boxed{CMC@1 = 0\,\%} \tag{26}$$

Le score final de mAP et de CMC est obtenu à partir de la moyenne des valeurs d'AP et de CMC de toutes les requêtes calculé de la même manière.

Il est important de noter qu'un CMC Rank-1 de 0 % n'implique pas nécessairement un mAP nul. Dans cet exemple, les trois images correctes sont bien présentes dans la galerie et sont finalement retrouvées par le système, mais elles apparaissent tard dans le classement. Le mAP conserve donc une valeur non nulle et permet de mesurer la qualité du classement au-delà de la seule première position.

Il est tout autant important de noter que le mAP lui-même ne dit rien de la distance entre les signatures visuelles. Il est possible d'observer un espace latent très compact mais extrêmement bien segmenté de sorte à ce que le classement de chaque requête soit optimal mais les scores de similarité en moyenne très haut même avec les images moins ressemblantes du jeu de donnée.

Le score de similarité étant la base des algorithmes de Recherche Opérationnelle pour pondérer les arêtes des graphes, cette mesure ne suffit donc pas à évaluer à elle seule la qualité des signatures traitées par les modèles utilisés.

.2 Métriques d'évaluation

Pour évaluer les performances d'un modèle de classification binaire, plusieurs métriques peuvent être calculées à partir des quatre catégories de la matrice de confusion : les vrais positifs (*True Positives*, TP), les vrais négatifs (*True Negatives*, TN), les faux positifs (*False Positives*, FP) et les faux négatifs (*False Negatives*, FN). Un **vrai positif** correspond à un échantillon correctement identifié comme appartenant à la classe positive, tandis qu'un **vrai négatif** correspond à un échantillon correctement identifié comme négatif. À l'inverse, un **faux positif** désigne un échantillon négatif classé à tort comme positif, alors qu'un **faux négatif** correspond à un échantillon positif classé à tort comme négatif. À partir de ces quatre valeurs, la **précision** mesure la proportion de prédictions positives qui sont correctes et est définie par $\text{Précision} = \frac{TP}{TP+FP}$. Le **rappel** (ou *recall*) mesure la proportion de positifs réels correctement détectés par le modèle et s'exprime par $\text{Rappel} = \frac{TP}{TP+FN}$. Le **score F1** permet de combiner ces deux métriques en calculant leur moyenne harmonique : $\text{F1} = 2 \times \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}}$. Enfin, l'**accuracy** (ou exactitude) représente la proportion totale de prédictions correctes parmi l'ensemble des observations : $\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$. Ces métriques permettent ainsi d'évaluer le comportement du modèle sous différents angles, notamment sa capacité à limiter les fausses détections (FP) et les détections manquées (FN).

Références

[1] ... ... *International Journal of ...*, 30(2), ...

[2] Shengcai Liao, Yang Hu, Xiangyu Zhu, and Stan Z. Li. Person re-identification by local maximal occurrence representation and metric learning, 2015.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[6] Shuting He, Hao Luo, Pichao Wang, Fan Wang, Hao Li, and Wei Jiang. Transreid: Transformer-based object re-identification, 2021.

[7] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification, 2019.

[8] Tianlong Chen, Shaojin Ding, Jingyi Xie, Ye Yuan, Wuyang Chen, Yang Yang, Zhou Ren, and Zhangyang Wang. Abd-net: Attentive but diverse person re-identification, 2019.

[9] Chenyang Yu, Xuehu Liu, Yingquan Wang, Pingping Zhang, and Huchuan Lu. Tf-clip: Learning text-free clip for video-based person re-identification, 2023.

[10] Ren Nie, Jin Ding, Xue Zhou, and Xi Li. Rethinking normalization layers for domain generalizable person re-identification. In *European conference on computer vision*, pages 267–284. Springer, 2024.

[11] Yoonki Cho, Jaeyoon Kim, Woo Jae Kim, Junsik Jung, and Sung eui Yoon. Generalizable person re-identification via balancing alignment and uniformity, 2024.

[12] Meng Zheng, Srikrishna Karanam, and Richard J. Radke. Rpifield: A new dataset for temporally evaluating person re-identification. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1974–19742, 2018.

[13] Ranjan Sapkota, Rahul Harsha Cheppally, Ajay Sharda, and Manoj Karkee. Yolo26: Key architectural enhancements and performance benchmarking for real-time object detection, 2026.

[14] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Learning generalisable omni-scale representations for person re-identification, 2021.

[15] Kien Nguyen, Clinton Fookes, Sridha Sridharan, Huy Nguyen, Feng Liu, Xiaoming Liu, Arun Ross, Dana Michalski, Tamás Endrei, Ivan DeAndres-Tame, Ruben Tolosana, Ruben Vera-Rodriguez, Aythami Morales, Julian Fierrez, Javier Ortega-Garcia, Zijing Gong, Yuhao Wang, Xuehu Liu, Pingping Zhang, Md Rashidunnabi, Hugo Proença, Kailash A. Hambarde, and Saeid Rezaei. Ag-vpreid 2025: Aerial-ground video-based person re-identification challenge results, 2025.

[16] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[17] Meng Zheng, Srikrishna Karanam, and Richard J Radke. Rpifield: A new datasef for temporally evaluating person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018.

[18] Cyrille Equoy, Alessandro Goldwurm, Cyrille Briand, and Frédéric Lerasle. A new path-oriented optimization procedure for consistent person reidentification in a camera network. In *2025 IEEE International Conference on Advanced Visual and Signal-Based Systems (AVSS)*, pages 1–6, 2025.

[19] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline), 2018.

[20] Guanshuo Wang, Yufeng Yuan, Xiong Chen, Jiwei Li, and Xi Zhou. Learning discriminative features with multiple granularities for person re-identification. In *Proceedings of the 26th ACM international conference on Multimedia*, MM '18, page 274–282. ACM, October 2018.

[21] Hong-You Chen, Zhengfeng Lai, Haotian Zhang, Xinze Wang, Marcin Eichner, Keen You, Meng Cao, Bowen Zhang, Yinfei Yang, and Zhe Gan. Contrastive localized language-image pre-training, 2025.

[22] Siyuan Li, Li Sun, and Qingli Li. Clip-reid: Exploiting vision-language model for image re-identification without concrete text labels, 2023.