IMT Nord Europe
École Mines-Télécom
IMT-Université de Lille

Ombeline MOINEAU
Simon GARY

RÉ-IDENTIFICATION DE PERSONNES

UV Projet P4

Encadrants : Raphaël DELECLUSE, Hazem WANNOUS
Année 2023-2024
FISE 2024

Table des matières

Table des matières **2**
Objectifs du projet **3**
Prétraitement de la base de données **4**
Rotation des images selon le sens de déplacement de l'individu 4
Suppression des images présentant trop de bruit 4
Suppression des images sans individu 5
Normalisation des séquences **6**
Première approche envisagée 6
Seconde approche envisagée 7
Création d'un graph sur une image de profondeur **8**
Résoudre le problème d'orientation des épaules 8
Récupération d'un masque pour les épaules et la tête 9
Définition des informations importantes pour le graph 11
Idée d'un GCN **11**
Références **13**

Objectifs du projet

Nous avons collaboré sur un projet de ré-identification de personnes en milieu ouvert, exploitant des images provenant d'une caméra RGB et d'une caméra de profondeur configurées en vue du dessus. Cette initiative se distingue par sa pertinence dans le contexte actuel de la vision par ordinateur et de la reconnaissance de formes. Notre projet permet de résoudre des problèmes concrets de ré-identification de personnes, en contraste avec la plupart des travaux existants qui se concentrent sur des scénarios en monde fermé, ce qui restreint leur utilité pratique.

Nous avons à notre disposition la base de données TVPR2, qui comprend 236 vidéos. Chaque vidéo présente plusieurs individus passant deux fois sous la caméra (dans deux sens différents) et marquant une brève pause à chaque passage sous la caméra. Il est pertinent de mentionner que la base de données sur laquelle nous travaillons résulte du traitement initial effectué par Raphaël sur les vidéos d'origine. En effet, Raphaël a prétraité les données et a créé une nouvelle organisation où chaque vidéo est répartie en plusieurs sous-dossiers d’images. Chaque sous-dossier correspond à un passage unique par individu. Lors du premier passage, l’individu se déplace de gauche à droite, alors que lors du deuxième, il se déplace de droite à gauche. Chaque sous-dossier de la base de données contient des images en couleur (RGB) et en profondeur (depth).

Nous avons l'intention d'exploiter les Graph Convolutional Networks (GCN) pour améliorer la précision de la ré-identification. L'objectif central de notre travail reste la reconnaissance effective d'un individu lors de son deuxième passage. Cette approche innovante, combinée à la richesse des données visuelles à notre disposition, place notre projet à la pointe de la recherche en matière de ré-identification de personnes dans des environnements ouverts. Notre approche représente un défi significatif dans le domaine, car elle se distingue de la tendance actuelle qui privilégie l'utilisation de Convolutional Neural Networks (CNN) en milieu fermé. Notre choix stratégique d'adopter les GCN dans un milieu ouvert témoigne de notre volonté d'explorer de nouvelles voies de la ré-identification.

Prétraitement de la base de données

Le succès d'un projet de ré-identification repose en grande partie sur la qualité de la base de données et le prétraitement des images. Dans cette section, nous présentons en détail les différentes étapes de prétraitement que nous avons entreprises pour garantir la fiabilité et la cohérence des données utilisées dans notre projet.

Rotation des images selon le sens de déplacement de l'individu

Afin d'assurer une uniformité sur l'ensemble des images, nous avons décidé d'effectuer une rotation sur une sélection d'images. En effet, nous souhaitons garantir un déplacement cohérent pour toutes les images, de sorte à ce que l'individu se déplace tout le temps dans la même direction, de la gauche vers la droite. Cela permettra par la suite d'obtenir des graphes homogènes pour une utilisation optimale des Graph Convolutional Networks (GCN).

Sur le deuxième passage de chaque individu, et cela pour chaque vidéo, le déplacement était tout le temps fait de droite à gauche. Par conséquent, nous avons effectué une rotation de 180 degrés sur toutes les images des deuxièmes passages (RGB et depth).

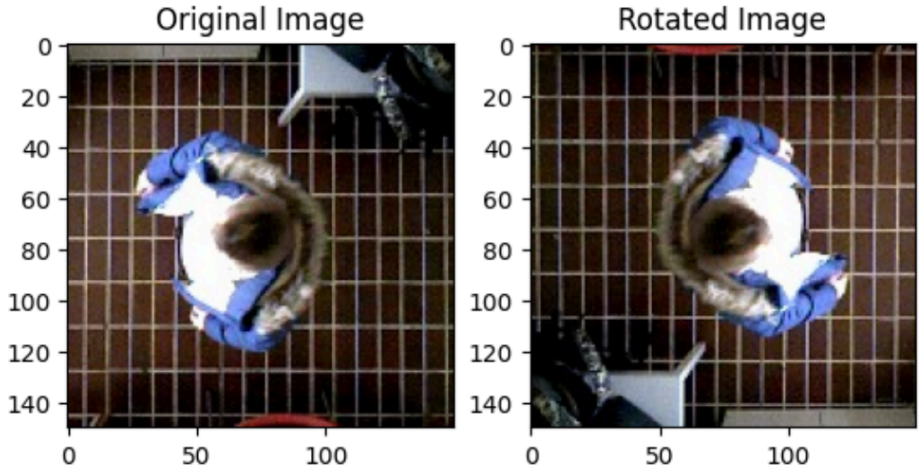


Figure 1 : Rotation d'une image de 180 degrés

Lien du notebook : 1.Process_rotate_images.ipynb .

Suppression des images présentant trop de bruit

De même, il y a certaines images qui comportent un niveau de bruit excessif, caractérisé par la présence de bandes noires indésirables, issues de la profondeur (profondeur 0), qui cachent une partie de l'individu. Ces images sont inexploitables pour nous. En effet, la qualité du graphe est altérée lorsque l'individu est dissimulé, ce qui compromet notre capacité à le reconnaître de manière fiable lors de la phase de ré-identification.

Suite à des tests approfondis, nous avons pris la décision de supprimer les images comportant plus de 20% de pixels noirs, un seuil significatif permettant d'éliminer les images bruitées. Ainsi, les bandes noires ne pourront pas occulter plus d'un cinquième de l'image.

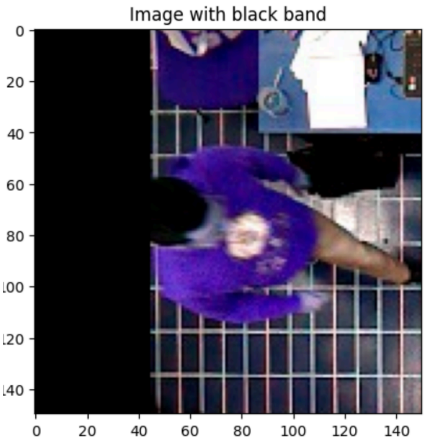


Figure 2 : Exemple d'une image présentant trop de bruit

Lien du notebook : 2.Process_black_band.ipynb

Suppression des images sans individu

Pour affiner la pertinence de notre base de données, nous avons exclu les images où aucun individu n'est présent. En effet, les scènes sans sujet ne présentent pas d'intérêt pour notre étude.

En prenant une image de profondeur test, sans individu, avec uniquement le décor (la table et les armoires) le pixel le plus petit a comme valeur de profondeur 2450. Ce qui veut dire que le point le plus haut a une profondeur de 2450 par rapport à la caméra, la caméra étant placée en top view (donc en profondeur 0).

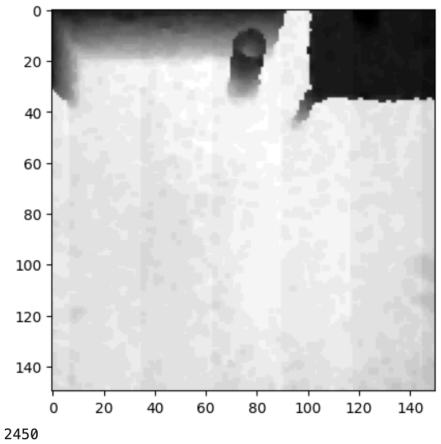


Figure 3 : Exemple d'une image sans individu, le point le plus haut a comme profondeur 2450

Le seuil de rejet a donc été défini à 2000, basé sur la valeur de profondeur des pixels. Ce qui veut donc dire qu'une image ne présentant pas de valeur de profondeur inférieure à 2000, est considérée comme une image sans individu.

Lien du notebook : 3.Process_no_human.ipynb

Normalisation des séquences

Dans cette section, nous exposons la méthode que nous avons adoptée pour standardiser les séquences, assurant ainsi une longueur fixe. Étant donné que nos vidéos ont été découpées en images, nous faisons face à une disparité dans le nombre d'images par passage. La normalisation s'avère donc essentielle pour homogénéiser ces séquences et faciliter le traitement ultérieur.

Première approche envisagée

Pour ce faire, une première approche a été envisagée, celle du DTW (Dynamic Time Warping). C'est une méthode qui consiste à détecter des motifs similaires sur deux séquences de tailles différentes afin de les rapprocher et de pouvoir normaliser la plus grande séquence par rapport à la plus petite. Cette solution aurait consisté à analyser chaque séquence d'une vidéo, à détecter des repères récurrents communs aux vidéos comme une armoire, une chaise, etc… afin de pouvoir déterminer la vitesse relative du sujet par rapport au décors à chaque moment de la séquence et ainsi nous permettre d'adapter le plus fidèlement possible la taille des séquences dans un objectif de normalisation de celles-ci.

Cependant, nous n'avons pas opté pour l'utilisation de cette solution pour plusieurs raisons.
Dans un premier temps, nous nous sommes rendu compte qu'après avoir réalisé la phase de rotation de la moitié des images lors du pré-traitement, nous obtenons donc une partie des images qui ont un décor renversé de 180° et donc une difficulté de plus pour corréler ces images entre elles à partir du décors.

Enfin, après une succincte analyse d'un échantillon de séquences, nous avons remarqué que nous avions une moyenne d'environ 50 images restantes par séquence et qu'en prenant en compte le mouvement de déplacement toujours linéaire il ne serait pas forcément nécessaire d'utiliser un DTW du fait de la taille des séquences relativement courtes ainsi qu'une simplicité du schéma de déplacement et donc de variations de la vitesse du sujet très faible au sein d'une même séquence. Nous avons pu remarquer qu'il y avait surtout un seul moment au sein d'une séquence où le sujet pouvait possiblement changer de vitesse.

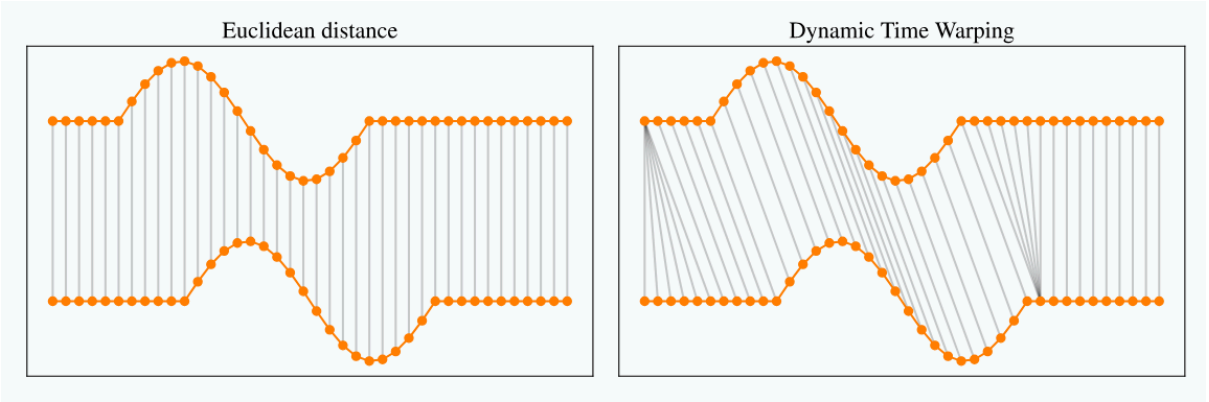


Figure 4 : Schéma d'un Dynamic Time Warping

Seconde approche envisagée

C'est donc à partir de cette dernière remarque que nous avons établi la seconde solution pour simplifier cette étape de normalisation tout en gardant le plus d'informations possibles par vidéo. Notre raisonnement s'articule autour d'une approche par vidéo, préservant ainsi la structure temporelle inhérente à chaque vidéo. Pour ce faire, nous avons pris la décision de diminuer le nombre d'images de chaque dossier en fonction du plus petit dossier au sein d'une même vidéo. Cette approche vise à éviter une perte significative en données qui résulterait d'une normalisation sur l'ensemble de la base de données totale.

Voici donc notre solution permettant d'obtenir des séquences de taille fixe, en itérant sur chaque vidéo :

1) La première étape de notre démarche consiste à déterminer la taille de la séquence la plus petite, en parcourant les sous-dossiers au sein de la vidéo. Cette information est essentielle pour standardiser la longueur des séquences.

Ensuite, nous itérons sur chaque sous-dossiers d'une vidéo :

2) Nous calculons le nombre d'images à supprimer pour chaque sous-dossier pour ajuster la séquence à la taille minimale. Pour ce faire, nous faisons *taille de la séquence - taille séquence minimale*.

3) Nous identifions la frame où le sujet se trouve au centre (sous la caméra). On utilise une fonction élaborée de la même manière que celle de Raphaël, le sujet est au centre lorsque sa profondeur est minimale, lorsqu'il est le plus proche de la caméra top view. Cette frame centrale devient notre point de référence pour le processus de normalisation.

4) Nous comptons le nombre d'images qu'il y avant et après cette frame centrale.

5) Nous déterminons combien d'images seront supprimées dans chaque portion (avant et après la frame centrale), évitant ainsi une perte disproportionnée d'informations cruciales. Pour ce faire, nous faisons *round(nb images portion / (nb images seq - 1) x nb images à suppr portion).*

6) Nous calculons la fréquence à laquelle on va supprimer les images. Le but étant d'éliminer les images à intervalles réguliers.
Pour ce faire, nous faisons *taille portion / nb images à suppr portion.*

7) Nous supprimons les images nécessaires dans le sous-dossier. Cette opération commence depuis les extrémités de la séquence, nous supprimons alors une image toutes les x fréquences arrondie. Cela permet d'assurer une normalisation équilibrée.

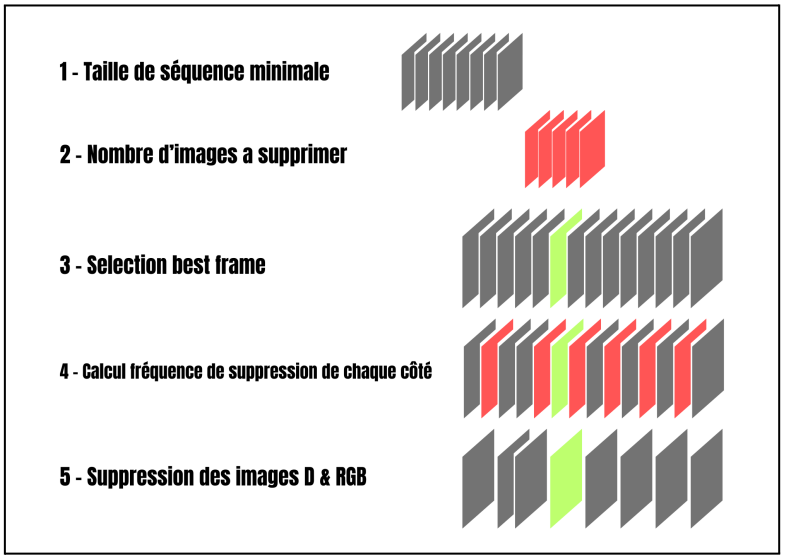


Figure 5 : Schéma d'exemple du processus de normalisation de séquence

Création d'un graph sur une image de profondeur

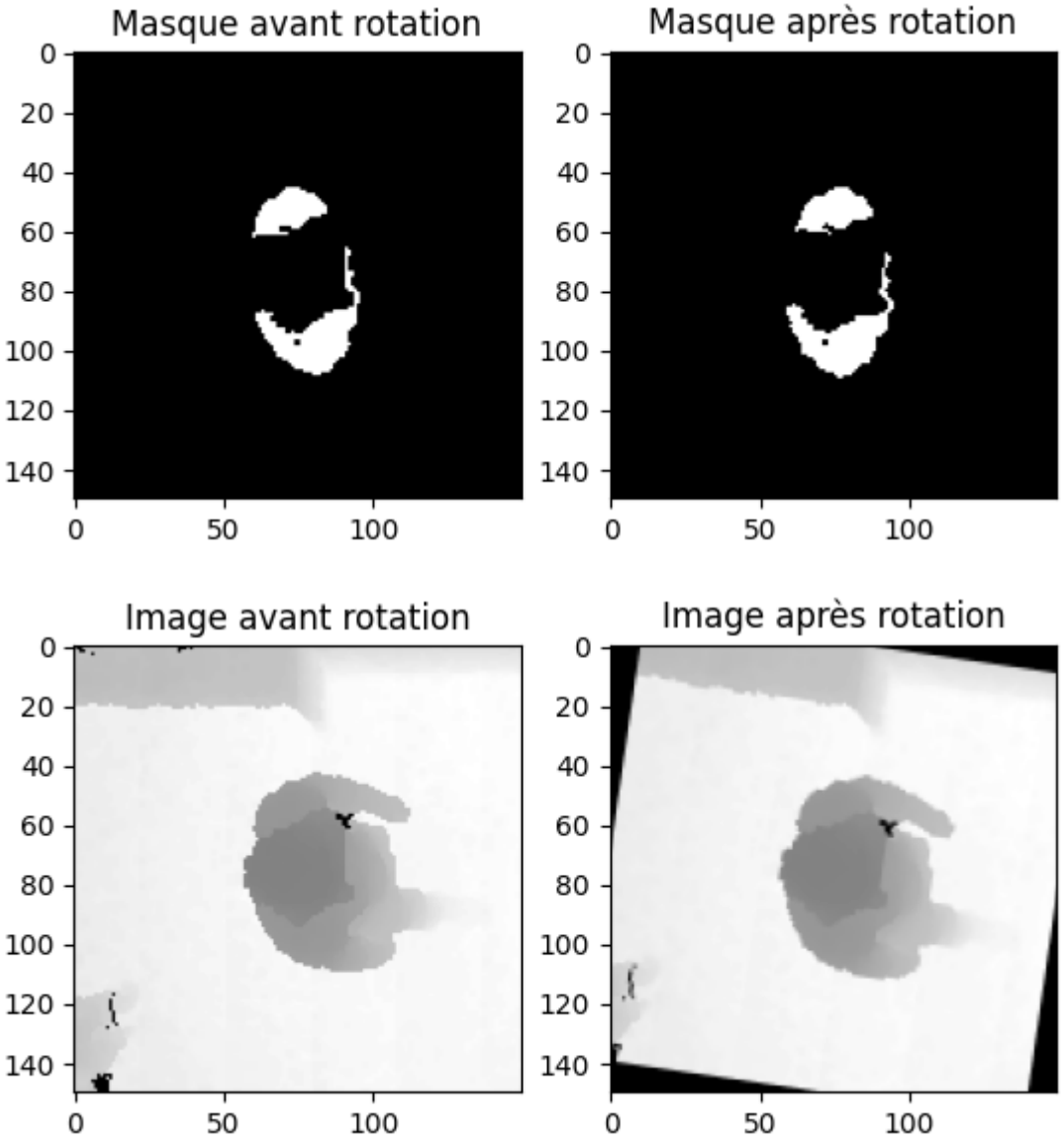
Une fois le dataset nettoyé et les images toutes orientées dans le même sens, nous pouvons maintenant commencer à créer un graph permettant d'identifier chaque individu. Pour ce faire, il nous faut placer des points à des endroits remarquables sur le sujet et de la manière la plus efficace possible.

Résoudre le problème d'orientation des épaules

La manière dont nous avons choisi de poser ces points remarquables nous a tout de suite posé un problème: celui de la rotation des épaules des sujets. En effet, nous avons choisi de placer simplement des points sur les épaules et la tête des individus en fonction des coordonnées extremum (verticales et horizontales) sur une certaine profondeur choisie. Pour ce faire, il faut alors que l'axe de l'épaule de chaque individu soit parallèle avec l'axe des ordonnées de l'image afin que tous les points de chaque graph puissent être comparables.

Pour résoudre ce problème nous avons donc créé une fonction qui repère les points extremum des épaules pour chaque image de profondeur et qui vient calculer l'angle entre l'axe des ordonnées et la droite reliant les deux points des épaules. Grâce à cet angle, nous pouvons donc orienter toutes les images et les masques permettant de réaliser

correctement les étapes suivantes et que chaque graph puisse être comparé entre eux sur les mêmes bases.



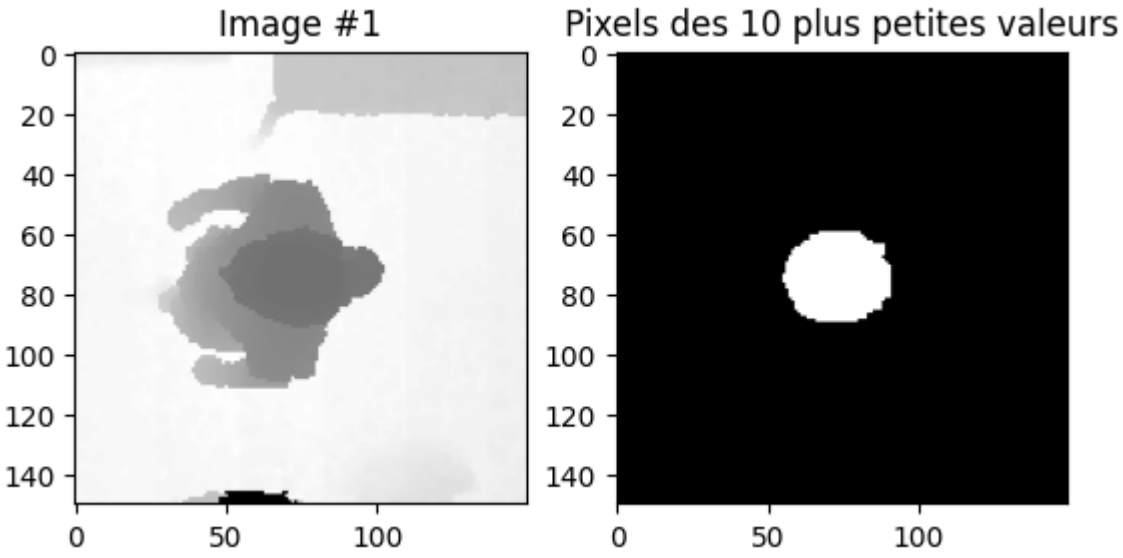
Récupération d'un masque pour les épaules et la tête

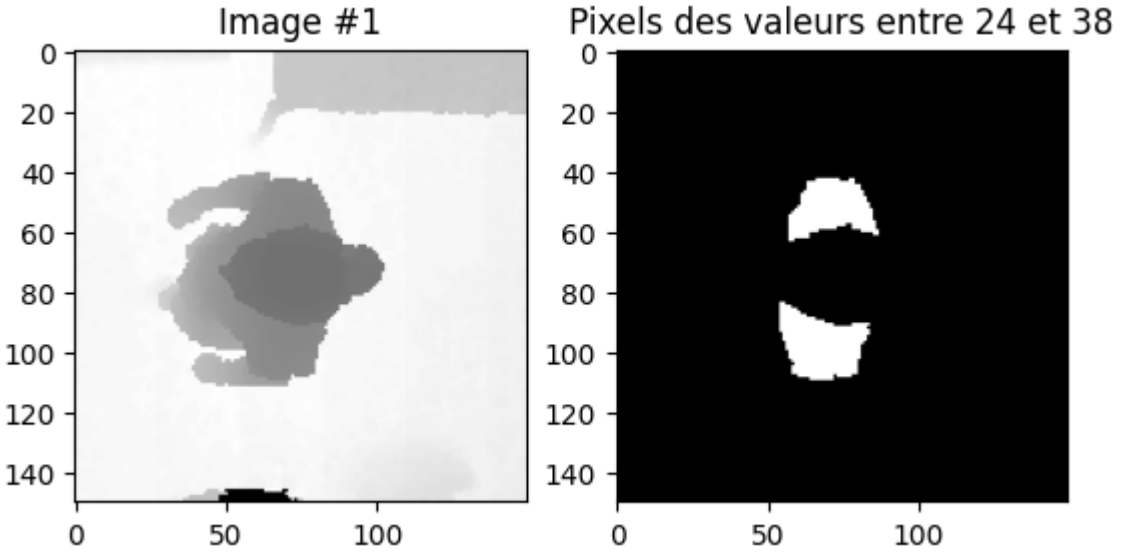
Après cette étape de rotation, il nous fallait donc définir quelles parties du corps seraient les plus judicieuses à exploiter sur ces images. Étant donnée la position de la caméra, en "top view", les informations les plus intéressantes sont la tête et les épaules. Nous avons donc créé 2 masques, un masque pour la tête et un pour les épaules. Un masque est une matrice de valeurs binaires permettant de localiser une zone précise sur une image.

Pour cette étape, le plus gros problème rencontré était la manière de détecter au mieux ces deux masques afin qu'ils correspondent à la même zone pour chaque individu. En effet, des facteurs comme la taille, la coiffure ou les habits de l'individu peuvent avoir un impact sur les masques sélectionnés. Nous avions comme donnée de départ uniquement une image de profondeur. A partir de l'image, nous avons créé une échelle de profondeur rassemblant toutes les différentes profondeur de l'image en ordre croissant. Le but était donc de

sélectionner uniquement les pixels de l'image qui comprenaient une profondeur entre la plus faible et la dixième plus faible. En faisant ainsi, on pouvait sélectionner à chaque fois le sommet du crâne de la personne, indépendamment de la taille du sujet, et la zone un peu plus basse de la tête afin d'avoir l'entièreté de la zone crânienne relativement à sa taille propre.

Pour le masque des épaules, nous avons procédé de la même manière mais avec un intervalle différent sur l'échelle de profondeur de l'image. Après analyse sur de nombreux sujets, nous avons choisi un intervalle entre la 24ème et la 38ème valeur de profondeur de l'image, ce qui permettait de sélectionner sur le masque la partie la plus intéressante des épaules du sujet.





Définition des informations importantes pour le graph

Maintenant que les masques ont été créés pour chaque image de profondeur, il nous faut déterminer les informations permettant la construction des graphs. Pour faire un graph, nous avons besoin de deux éléments: des nœuds et des liaisons entre ces nœuds.

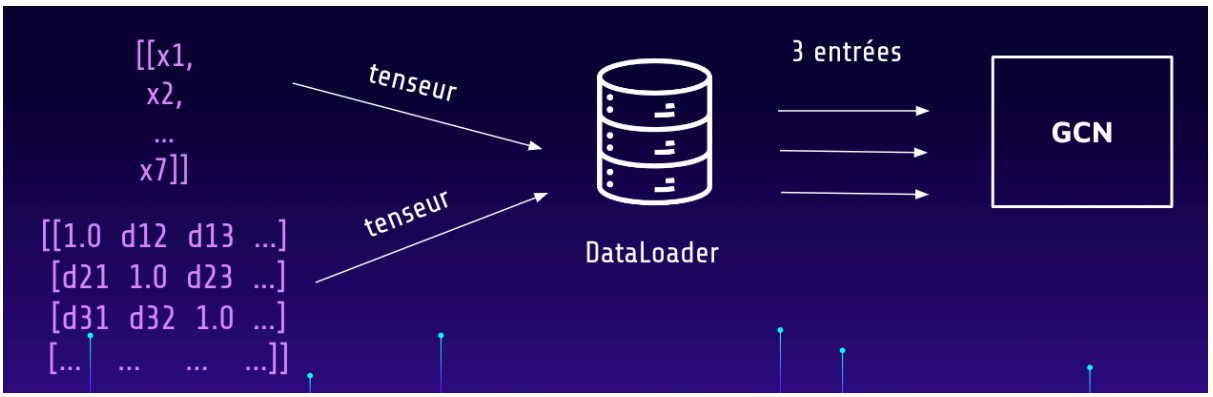
Pour les nœuds, nous avons donc fait le choix de sélectionner 7 points sur l'image de profondeur grâce aux deux masques créés précédemment. Deux points représentent les extrémités des épaules, quatre points seront sur les quatre extrémités de la tête et le dernier point sera représentant du sommet de celle-ci. Pour déterminer les coordonnées du sommet crânien, nous réalisons un masque provisoire contenant uniquement les profondeurs les plus basses et on récupère la coordonnée de barycentre de ce masque. Une fois que l'on a ces 7 points sur l'image, on peut leur associer à chacun la valeur de profondeur correspondante.

Enfin, pour obtenir les liaisons du graph entre ces nœuds, on a décidé de prendre la distance reliant chaque point entre eux pour créer la matrice d'adjacence de notre graph, une matrice qui sera donc symétrique car non-orientée.

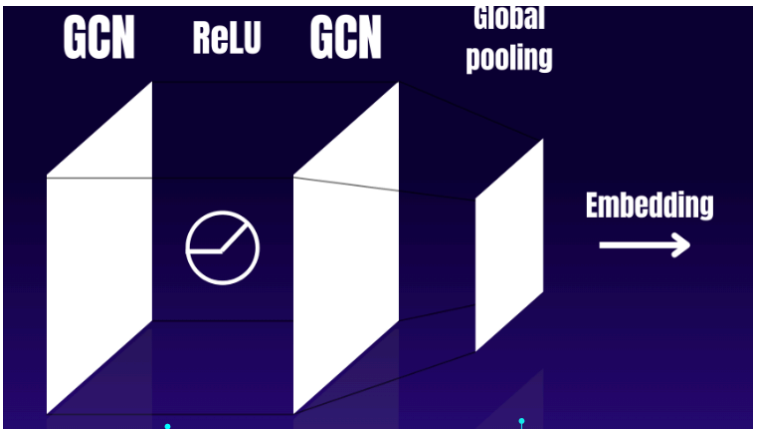
Ainsi, lors de son entraînement, notre modèle aura à sa disposition deux matrices, une matrice avec les valeurs de profondeur et une matrice avec les distances entre les nœuds. Il pourra donc prendre en compte des distances physiologiques importantes comme la longueurs des épaules, la taille de la tête, la distance tête/épaules, mais aussi les valeurs de hauteur de l'individu comme sa taille totale, mais aussi sa hauteur d'épaules. De nombreuses données très utiles et uniques à chacun pour permettre l'identification la plus exacte possible.

Idée d'un GCN

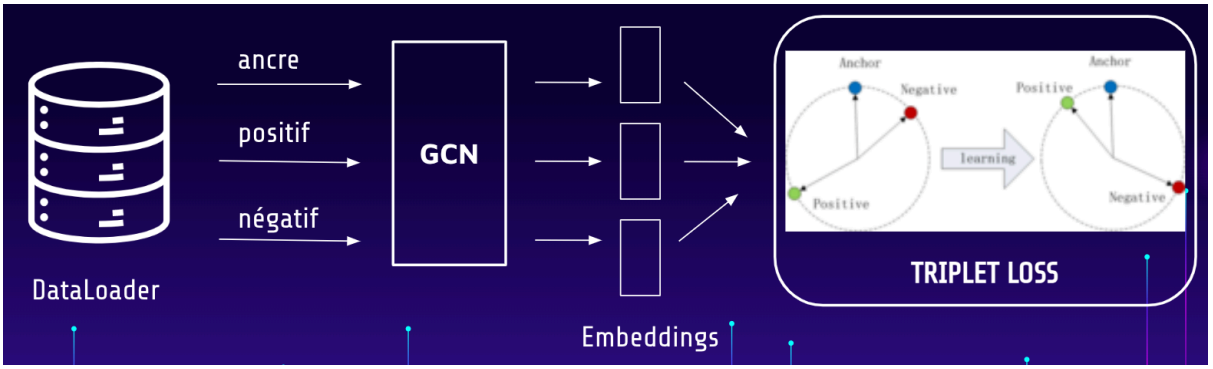
Passons désormais à la conception de notre Graph Convolutional Network (GCN), maintenant que nos graphes sont prêts. Pour ce faire, nous convertissons nos matrices de profondeur et de distance en tenseurs, que nous introduisons dans un dataloader. Ce dataloader génère ensuite trois types d'entrées - ancre, positif et négatif - que nous acheminons vers le modèle GCN, car nous allons appliquer une triplet loss.



En ce qui concerne le modèle GCN, il se compose de plusieurs couches, notamment une couche de GCN, suivie d'une couche ReLU, puis d'une autre couche de GCN, et enfin d'une couche de global pooling.



Faisons une mise au point sur la triplet loss : le dataloader produit trois types d'entrées, représentant respectivement l'ancre (le point de référence), le positif (un point similaire) et le négatif (un point différent). Ces entrées sont ensuite traitées par le GCN, les embeddings sont générés, et pendant l'entraînement, la triplet loss est calculée. Cette loss vise à minimiser la distance entre l'ancre et le positif tout en maximisant celle entre l'ancre et le négatif. La rétropropagation du gradient est ensuite utilisée pour ajuster les paramètres du modèle.



Le but de ce processus est de créer des embeddings significatifs pour les points dans nos graphes, de manière à favoriser le regroupement des points similaires et la séparation des points différents dans l'espace d'embedding. Cette approche renforce la capacité du modèle à effectuer des tâches ultérieures telles que la classification ou la recherche de similarité, contribuant ainsi à une analyse plus précise et efficace de nos données graphiques.

Références

[1]M. Martini, M. Paolanti and E. Frontoni, "Open-World Person Re-Identification With RGBD Camera in Top-View Configuration for Retail Applications," in IEEE Access, vol. 8, pp. 67756-67765, 2020, doi: 10.1109/ACCESS.2020.2985985.

[2]Tavenard, R. (Publication date not available). "An Introduction to Dynamic Time Warping.". Retrieved from https://rtavenar.github.io/blog/dtw.html