



## **1. Introduction**

Notre projet consistait à détecter des piétons dans une image en ayant préalablement construit la signature de ces derniers dans une autre image grâce à la méthode HOG (Histogram of Oriented Gradients).

## **2. Explication de notre démarche**

Dans la première image (celle d'entraînement), nous avons dans un premier temps isolé le piéton et l'avons redimensionné dans une image en 64\*64. Nous avons divisé cette dernière en 8\*8 images carrées.

Nous pouvons donc utiliser la méthode HOG sur ces 64 nouvelles images pour construire la signature de notre piéton.

La méthode HOG s'implémente de la manière suivante :

1. Prétraitement de l'image : L'image est convertie en niveaux de gris pour réduire la complexité de traitement et elle est filtrée avec un filtre gaussien de dimension 5 afin d'éviter les erreurs de calculs causées par le bruit sur l'image en entrée.
2. Calcul des gradients d'orientation : Les gradients d'orientation décrivent la direction et l'intensité des variations de luminosité dans l'image.
3. Construction des histogrammes d'orientation : Les gradients d'orientation sont regroupés en cellules, et un histogramme d'orientation est calculé pour chaque cellule. Les histogrammes d'orientation décrivent la distribution des orientations des gradients dans chaque cellule.
4. Récupération de l'orientation principale (angle et magnitude) de chacun des 64 histogrammes. On obtient ainsi le vecteur principal pour chaque cellule. Afin d'éviter des effets indésirables liés à une sur/sous exposition de soleil, on normalise l'ensemble des vecteurs.
5. Vient la phase de comparaison entre la signature du piéton (nos 64 vecteurs principaux normalisés de l'image redimensionnée du piéton) et un autre bloc d'une image. L'objectif est de savoir si oui ou non ce bloc pourrait s'apparenter à un piéton. On effectue une recherche par bloc sur la nouvelle image (en calculant la signature de chacun des blocs) puis on calcule l'erreur entre la signature du bloc et la signature que nous avons récupérée.

Nous avons essayé deux calculs d'erreurs différents: la différence classique et la différence au carré pour pénaliser encore plus les grands écarts entre les deux signatures. Ces deux types d'erreurs nous donnaient des résultats similaires.

Nous avons donc testé la recherche de piétons sur notre image d'entraînement et l'on retrouve directement notre piéton ce qui est une bonne chose.



### **3. Difficultés rencontrées et comparaison avec une méthode déjà implémentée**

Nous avons rencontré des difficultés pendant notre phase de test, sur notre deuxième image. Nous avons recherché notre piéton avec la signature de la première image avec une fenêtre glissante mais nous n'avons pas retrouvé directement notre piéton.

Nous avons donc cherché plus loin que le premier résultat et avons affiché les 15 premières images (donc les 15 plus petites erreurs) mais nous n'avons pas notre piéton.

Nous avons modifié plusieurs fois l'algorithme et effectué énormément de recherches sur internet pour comprendre ce qui ne pouvait pas aller. Nous avons finalement trouvé une fonction du module OpenCV pour effectuer une détection de piéton.

Après avoir comparé nos résultats avec la fonction *hog* d'OpenCV nous avons remarqué que, même avec cette dernière, nous obtenons plusieurs faux positifs lors de la détection de piétons en particulier si l'image contient des éléments qui ont une apparence similaire à celle d'un piéton. Nous voyons une autre manière de le faire grâce au machine learning mais cette façon de faire ne rentre pas dans le cadre de ce projet. Voici les quelques étapes que l'on aurait pu suivre :

1. Constitution d'un jeu de données d'entraînement comportant des images de piétons sous différents angles, formes, couleurs afin d'avoir une représentation exhaustive du piéton. Après utilisation de la méthode HOG, on obtient un jeu de données avec non plus une seule signature mais un ensemble de signatures qui décrit de façon fidèle notre piéton.
2. Ensuite on procède à l'apprentissage d'un modèle de classification comme le SVM à partir de notre jeu de données.
3. Après validation de nos hyper-paramètres, on peut réaliser des prédictions pour savoir si un bloc comporte un piéton ou non, sous réserve d'une précision précédemment estimée.

Il est également important de noter que la qualité des images joue un rôle important dans la précision de la détection des piétons. Les images avec une résolution élevée et une bonne qualité d'image sont généralement plus faciles à analyser pour détecter des piétons. Il est donc important de s'assurer que les images utilisées pour la détection des piétons sont de bonne qualité.



## **4. Conclusion**

Malheureusement, nous n'avons pas réussi à détecter le piéton en utilisant les images 6 et 9 comme demandé dans l'énoncé, cet échec est sûrement dû à une trop grande différence entre la position des piétons, cependant en utilisant deux autres images du dataset (les images 7 et 8) nous obtenons des résultats satisfaisant en retrouvant les piétons parmi les trois premières images. En traitant ces différentes images, nous avons appris en détail comment réaliser la méthode HOG pour construire la signature d'un élément à reconnaître.

Nous ne savons pas si nous pouvons améliorer réellement cette méthode en utilisant uniquement du traitement de l'image, nous avons essayé tout ce que nous étions en capacité d'implémenter pour trouver un piéton mais cela n'a pas fonctionné. Par contre, en mettant ce projet en lien avec nos cours d'introduction au machine learning, nous savons qu'il est réalisable de créer un modèle pour améliorer cette détection en utilisant un plus grand dataset et un classifieur.