POLYTECHNIQUE MONTRÉAL

INF6804 - Vision par ordinateur
Hiver 2025

Pratique #1 Description et comparaison de régions d'intérêt

Yassine Serroukhe Idrissi - 1933389
Raphaël Le Blanc - 2409175

DATE DE REMISE

Table des matières

1 Question 1

Les deux approches utilisées sont :

— **HOG (Histogram of Oriented Gradients)** : C'est une méthode d'extraction des caractéristiques qui analyse les variations de luminosité dans une image en calculant des gradients d'intensité.

— **CLIP** : C'est un modèle d'apprentissage profond qui associe des images et des textes en apprenant des représentations communes. Il permet la recherche d'image par description textuelle.

2 Question 2

Si les objets dans l'image ont une taille uniforme, la méthode la plus recommandée est **HOG** car elle est efficace pour comparer les objets de tailles similaires. Elle analyse les contours et textures locales de manière précise.

Même si **CLIP** fonctionnera également très bien , c'est un modèle plus lourd et couteux à entraîner , aussi on privilégie la solution la plus simple/économique.

Cependant, si les objets dans l'image varient en taille, on utilise **CLIP** car cette méthode peut reconnaître des concepts plus abstraits et généraliser grâce à son apprentissage sur des données texte-image.

3 Question 3

Pour **HOG**, la performance devrait s'améliorer car la boîte englobante réduit le bruit en se concentrant uniquement sur l'objet d'intérêt.

En revanche, pour **CLIP**, l'amélioration sera plus limitée car CLIP analyse l'image dans son ensemble. Toutefois, la boîte englobante peut tout de même aider à mieux cibler l'objet.

4 Question 4 :

Pour tester les hypothèses que nous avons émises nous avons réaliser plusieurs expériences : - Tout d'abord nous avons constitué une base de données d'images ainsi qu'un ensemble d'images requetes pour differentes catégories

-pour chacune de nos images requête, nous avons calculé la distance avec toutes les images de notre base de données. Pour CLIP, la distance est définie comme « similarité cosinus » entre les embeddings normalisés, tandis que pour HOG, nous utilisons la norme euclidienne entre les vecteurs de caractéristiques.

Nous avons évalué deux configurations : -Sans extraction de ROI : Comparaison sur l'image entière. -Avec extraction de ROI : Utilisation d'un recadrage automatique (basé sur les contours) afin de se concentrer sur l'objet d'intérêt. La difference de taille des objets et la variations de lumiere et des arrière plan detailler represente des defis pendant l'analyse des images Pour chaque image requête, nous analyson les 5 meilleurs correspondances obtenues et calculons le nombre d'images dont le nom contient le label recherché, puis nous affichons les resultas sous forme graphique (diagrammes en barres)

5 Question 5 :

5.1 Implémentation CLIP :

Pour clip nous avons utiliser la bibliothèque Hugging Face pour charger le modèle déjà entraîné "openai/clip-vit-base-patch16" puis nous avons extrait l'embeddings de clip à partir d'images, avec ou sans ROI, et on calcul la distance via la similarité cosinus entre vecteurs normalise

5.2 Implémentation HOG

Pour HOG nous avons utilise OpenCV pour le pré-traitement puis nous avons extrait des caractéristiques avec cv2.HOGDescriptor en définissant des paramètres tels que $winsize = (64, 64), blocksize = (16, 16), blockstride = (8, 8), cellsize = (8, 8), nbins = 9$.

Puis nous avons calculé la distance entre deux images à partir de la norme euclidienne entre leurs vecteurs HOG. la plus part du code a été développée par nos soins, en s'appuyant sur la documentation d'OpenCV et Hugging Face. Nous avons notamment ajusté les seuils de détection et la gestion des ROI pour optimiser la détection en fonction de nos images.

Malheureusement, pour l'implémentation de **HOG**, nous n'avons pas réussi a obtenir de bon résultats au niveau des boîtes englobantes "automatiques", nous avons donc décidé de simplement crop les images à la main et de faire une nouvelle base de données contenant les images ajustées de cette manière.

6 Question 6 et 7

6.1 HOG

Concernant l'hypothèse de la question 3, nous avons pu la vérifier avec **HOG** , grâce aux résultats suivants : Il est important de préciser ici que les seuls objets pour lesquels on a appliqué une boîte englobante sont :

— Dolphin
— Car
— Ball
— Lotus

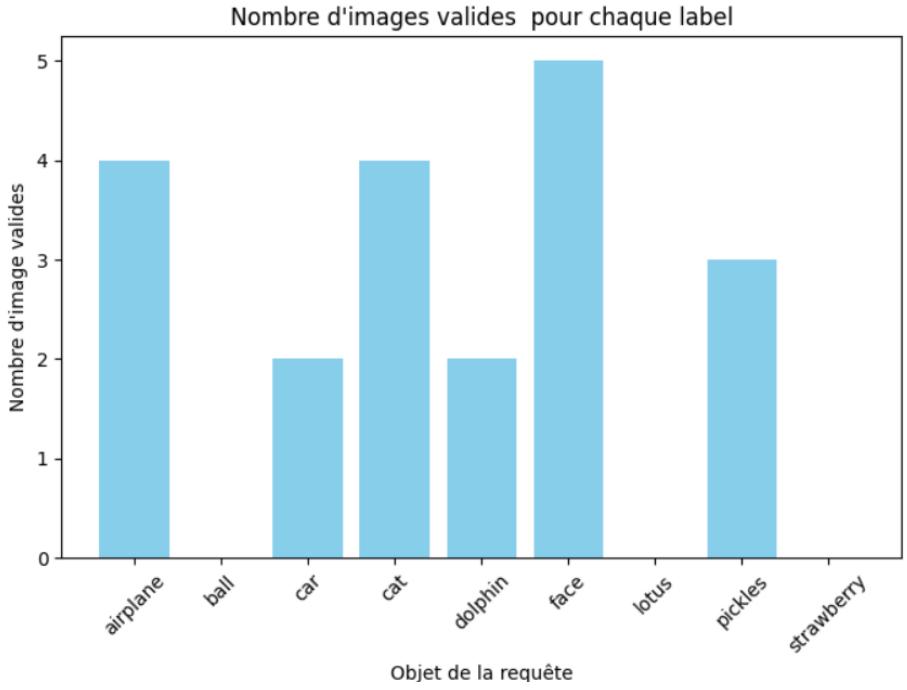

FIGURE 1 – Résultats HOG sans isoler l'objet dans une boîte englobante

FIGURE 2 – Résultats HOG en utilisant une boîte englobante

Ainsi , on peut voir que l'algorithme HOG performe bien mieux avec des boîtes englobantes que sans. On peut donc dire que l'hypothèse est vérifiée.

Pour ce qui est de l'hypothèse de la question 2 , pour la grande majorité des objets, leur taille ne changeait pas significativement entre l'image de requpete et les autres images de la base de données. Les 2 exceptions sont :

— pickles
— Lotus

6.2 CLIP

Concernant l'hypothèse de la question 3, nous avons pu la vérifier avec **CLIP** , grâce aux résultats suivants :

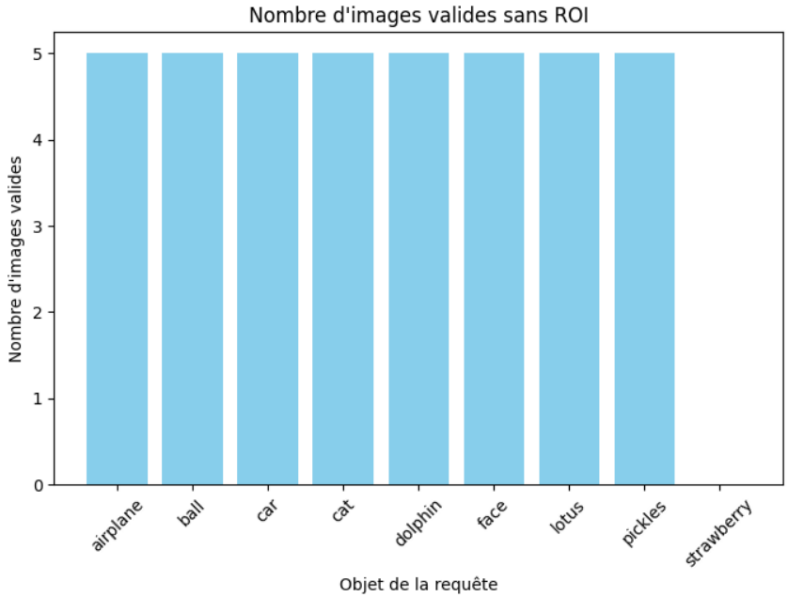

FIGURE 3 – Résultats de CLIP sans ROI ou boîte englobante

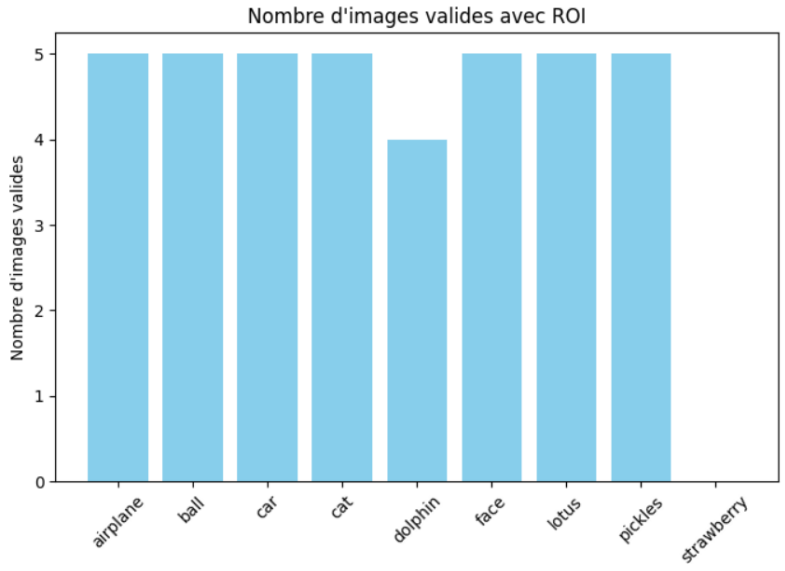

FIGURE 4 – Résultats de CLIP avec ROI

On peut remarquer que l'algorithme CLIP performe de la meme facon avec des boîtes englobantes et sans. On peut donc dire que l'hypothèse est vérifiée.

Pour ce qui est de l'hypothèse de la question 2 , pour la grande majorité des objets, leur taille ne changeait pas significativement entre l'image de requete et les autres images de la base de données. Les 2 exceptions sont :

— pickles
— Lotus

et CLIP arrive a detecter toutes les images dans la bases de donné qui ont comme label pickles et lotus se qui prouve notre hypothèse que CLIP performe bien meme si les objets sont de tailles diférentes dans les images