INF6804 : Vision par Ordinateur

TP1
Description et comparaison de régions d'intérêt

Émile Gagnon - 1961188

Pierre-Emmanuel Savoie - 1952752

Section : 01

6 février 2023

Département de génie Informatique

1 Introduction

L'objectif de ce rapport est de se familiariser avec des techniques de traitement d'images et de comparer les performances de ces différentes techniques. Les deux techniques utilisées et comparées sont les techniques d'histogramme de gradients orientés (HOG) et de transformation de caractéristiques visuelles invariante à l'échelle (SIFT). Ces techniques seront comparées grâce à une banque d'images fournie précédemment.

2 Questions

2.1 Présentation des deux approches

2.1.1 Histogramme de gradients orientés (HOG)

La méthode HOG consiste à créer un vecteur contenant plusieurs histogrammes de l'orientation du gradient pondérés par la magnitude du gradient. Il faut donc premièrement diviser l'image en plusieurs cellules, puis calculer un histogramme de l'orientation des gradients dans la cellule où le poids de chaque gradient calculé est déterminé par la magnitude de ce gradient. Ensuite, le vecteur descriptif est constitué de tous les histogrammes obtenus dans chacune des cellules.

2.1.2 Transformation de caractéristiques visuelles invariante à l'échelle (SIFT)

La méthode SIFT fonctionne en deux étapes distinctes. Premièrement, il faut détecter les points clés puis générer un vecteur descriptif pour ces points clés. La détection de points clés demande premièrement de calculer la différence de convolutions de fonctions gaussiennes ayant des écarts types variables avec l'image. Il faut donc calculer la différence de deux fonctions gaussiennes, puis la convolution du résultat avec l'image et répéter ce processus pour obtenir plusieurs convolutions différentes. Ensuite, un point clé est détecté s'il est un maximum ou minimum de ses 8 voisins et des 18 voisins de la convolution suivante et précédente. Ensuite, certains de ces points sont retirés de la liste soit parce qu'ils ont un contraste trop faible ou parce qu'ils sont situés sur une frontière entre deux régions de l'image. Les points restants sont les points clés. Chacun des points clés va ensuite être transformé en descripteur qui est construit en utilisant des histogrammes à huit cases des orientations du gradient autour du point clé pour 16 sous-régions voisines.

2.2 Hypothèses de performance pour des cas spécifiques

Dans le cas où il y a des rotations dans le même plan de l'image, on s'attend à ce que l'algorithme SIFT soit meilleur que l'algorithme HOG. En effet, la détection de point clé et le pairage d'un point clé d'une image à une autre devraient être bons, même si l'image a subi une rotation, puisque l'orientation et le gradient autour du point clé subiront une rotation similaire. Cela dit, on s'attend à ce que la détection d'images de dauphins fonctionne mieux avec SIFT qu'avec HOG puisque les images de dauphins sont très similaires avec seulement une différence de rotation du dauphin. À l'inverse, nous

pensons que pour des rotations dans un plan hors de l'image, tel qu'avec les images de cornichons, l'algorithme SIFT ne sera pas excellant puisque les points clés seront différents. Nous posons donc l'hypothèse que, pour les images de cornichons, l'algorithme HOG aura de meilleures performances puisque la distribution des textures dans les images sera relativement similaire, largement grâce à l'arrière-plan.

Lorsqu'on considère la distinction des caractéristiques extraites pour différents objets, on s'attend également à un comportement différent entre les deux algorithmes. SIFT extrait des points très ou même trop spécifiques, ce qui fait que cet algorithme est très bon pour reconnaitre le même objet, mais dans différentes circonstances. Par contre, puisque les caractéristiques extraites sont très spécifiques, SIFT aura de la difficulté à reconnaitre des objets similaires, mais légèrement différents. On s'attend donc à de très bonnes performances de SIFT avec des images comme le visage, mais à de moins bonnes performances avec des images comme les automobiles. L'algorithme HOG trouve des caractéristiques qui sont beaucoup moins distinctes à un objet précis que SIFT, lui permettant de reconnaitre plus facilement des catégories d'objets qui sont similaires avec de légères différences, comme les avions et les automobiles.

Une autre hypothèse est que SIFT aura de la difficulté à comparer des images simples avec des images complexes. En effet, puisque l'image complexe générera beaucoup de points clés, il sera facile pour l'image simple qui aura moins de points de trouver des paires avec les points clés de l'image complexe. La métrique principale pour la comparaison d'images avec SIFT est le nombre de paires de points clés que nous allons pouvoir 'matcher', les images complexes qui ont beaucoup de points clés seront alors avantagées. Notre hypothèse est alors que les images de ballons qui sont très simples auront un grand taux d'erreur avec SIFT, peut-être principalement avec les images de chats qui ont une forme similaire, mais qui sont des images plus complexes et donc auront plus de points clés.

2.3 Hypothèses de performance pour des boites englobantes

Concernant l'utilisation de boites englobantes autour de chaque objet, on s'attend à ce que l'algorithme SIFT soit légèrement plus performant, puisque SIFT extrait les points clés de l'objet, donc enlever l'arrière-plan permettra de seulement conserver les points clés voulus. Par contre, ceci ne devrait pas être une amélioration majeure, puisque l'algorithme élimine déjà les points qui se trouvent dans des régions à bas contraste, ce qui est souvent le cas en arrière-plan. Pour l'algorithme HOG, l'utilisation de boites englobantes peut autant augmenter ou diminuer la performance de l'algorithme. Des images avec un arrière-plan similaire seront plus difficiles à associer alors que des images qui ont un arrière-plan différent seront plus faciles à associer après l'utilisation de boites englobantes. On s'attend donc par exemple que HOG soit meilleur avec des images comme les lotus, mais moins bon avec des images comme les avions, les automobiles et les cornichons.

2.4 Description des expériences, données et critères d'évaluations

Afin de tester les 4 hypothèses émises précédemment, nous avons procédé en déterminant pour chaque image requête le top 1, 3 et 5 des images les plus similaires. Ceci a été effectué pour toutes les images requête sur toute la banque d'images pour une première fois et a été effectué une deuxième fois après le retrait de l'arrière-plan de chacune des images pour représenter une boite englobante. Il est ensuite possible de comparer les résultats des deux algorithmes pour certaines images requêtes spécifiques afin de vérifier les hypothèses énoncées précédemment. Ces résultats seront présentés dans la section 2.6.

Chaque image requête présentait une difficulté différente, ce qui nous a permis de tester efficacement nos hypothèses. Premièrement, les avions sont tous relativement différents et complexes, mais ont un arrière-plan très simple et similaire (une plaine avec du ciel). Les Ballons de soccer sont tous des objets très simples avec peu de caractéristiques et certains ont des arrière-plans très différents. Les automobiles sont complexes et on des arrière-plans très complexes, mais les objets et les arrière-plans sont assez similaires. Les chats ont tous beaucoup de contraste dû à leur pelage tacheté. Les dauphins ont été photographiés dans différents sauts donc les images présentent souvent des rotations dans le plan de l'image. Les visages sont tous le même objet cible, mais avec des arrière-plans différents et des angles légèrement différents. Les lotus présentent des arrière-plans très différents, particulièrement puisque l'image requête a un arrière-plan très foncé et beaucoup d'images de la banque en ont un assez pâle. Finalement, les cornichons présentent tous le même objet, mais sous des angles très différents, incluants des rotations hors du plan de l'image. De plus, une difficulté pour toutes les images est que les images ne sont pas de la même taille.

Pour les deux algorithmes, plusieurs mesures de similarité ont été testées et comparées afin d'utiliser celle avec les meilleurs résultats. Pour l'algorithme HOG, les mesures testées sont la distance L1, la distance L2, la similarité cosinus et l'intersection d'histogrammes. La mesure retenue qui présentait les meilleurs résultats a été la distance L1. Ensuite pour l'algorithme SIFT, les mesures de similarité testées sont le brute force matching et le FLANN matching (Fast Library for Approximate Nearest Neighbors). La mesure retenue a été le FLANN matching.

Pour la requête de l'image de fraise, nous savons à l'avance qu'il n'y a pas d'autres images de fraises dans la base de données. Notre objectif serait alors de voir si nos algorithmes sont capables de déceler qu'il n'y a aucune image de fraise dans la base de données. Pour faire cela, il faut adopter une stratégie différente. Nous avons décidé d'utiliser nos implémentations de HOG et SIFT et de trouver la moyenne de similitude entre les requêtes et les images de la base de données, pour chaque rang de solution. C'est-à-dire la moyenne des similitudes entre l'image qui est considérée la plus similaire à chaque requête, pour les deuxièmes plus similaires et ainsi de suite. Notre hypothèse est alors que la mesure de similarité entre la fraise et n'importe quelle autre image sera inférieure aux plus grandes moyennes de similarités des autres requêtes. Pour la mesure de similarité avec HOG, il est très simple

d'utiliser la distance comme précédemment. Par contre, avec SIFT, l'utilisation du nombre de match serait biaisée puisque certaines images ont plus de points clés que d'autres et donc plus de matchs. Nous avons donc opté pour un pourcentage de match trouvés, soit le nombre de match trouvés divisés par le nombre de points clés de l'image requête.

Finalement, la métrique d'évaluation utilisée est la précision obtenue sur le top 3 des images les plus similaires pour toutes les requêtes. Cependant, il est également intéressant d'observer le top 5 des images les plus similaires afin de voir si l'algorithme à réussi a repérer les 5 images correspondant à la requête.

2.5 Description des deux implémentations utilisées

Pour l'implémentation du code, nous avons utilisé la librairie OpenCV pour la majorité des techniques et la librairie matplotlib pour la lecture et l'affichage d'images. Nous avons tout d'abord défini une graine pour la reproductibilité des résultats puisque certaines méthodes font appel à de l'aléatoire. Afin de ne pas jouer avec cette graine pour avoir de meilleurs résultats, nous avons décidé d'une valeur fixe au tout début des expériences, soit le matricule d'un des deux étudiants.

La première implémentation est celle de HOG, pour cela nous avons utilisé l'objet *HOGDescriptor()* de la librairie OpenCV qui permet de générer des descripteurs HOG pour des images. Nous avons donc ensuite lu toutes les images de la base de données, converti en noir et blanc et calculé de descripteur HOG pour ces images. Cela nous donne une base de données de descripteurs HOG qui seront utiles pour la comparaison avec les requêtes par la suite. Par la suite, une requête à la fois, on calcule le descripteur HOG de la requête et on calcule sa distance à tous les descripteurs de la base de données. Nous avons essayé toutes les mesures de distances mentionnées dans la section précédente, mais la meilleure était celle de la distance L1. Par la suite nous pouvons trier les images en fonction de la plus petite distance et garder les 5 premières. Une fois que cela a été fait pour chaque requête, nous pouvons afficher les images et calculer les précisions top 1, top 3 et top 5.

Il est important de mentionner que pour comparer des images avec HOG, il faut que les images aient le même nombre de descripteurs et donc soient de même grandeur. Dans la partie précédente, nous avons relevé la difficulté que les images ne sont pas tous de la même grandeur. Pour contrer ce problème, avant d'utiliser une image avec HOG, celles-ci sont redimensionnées à une grandeur de 400 par 400. Nous avons testé pour plusieurs grandeurs d'images différentes, telles que 1000 par 1000 ou 4000 par 4000, mais les résultats étaient similaires et 400 par 400 était plus rapide puisqu'il y a moins d'histogrammes à calculer.

La seconde implémentation est celle de SIFT, pour cela nous avons utilisé l'objet *SIFT_ create()* ainsi que sa méthode *detectAndCompute()* qui permet de trouver les points clés et descripteurs pour une image. Pour cette technique, nous n'avons pas eu besoin de redimensionner les images puisque

le nombre de points clés ne dépend pas de la taille de l'image, simplement de sa complexité. Nous avons encore transformé les images en noir et blanc et calculé les descripteurs pour chaque image de la base de données. Tout comme pour HOG, nous avons calculé les descripteurs de chaque requête et les avons comparés aux descripteurs des images de la base de données. Pour la comparaison de descripteurs SIFT, nous avons utilisé les objets *BFMatcher()* et *FlannBasedMatcher()* d'OpenCV. Nous avons fait plusieurs tests avec les deux objets et leurs différents paramètres et avons trouvé que le *FlannBasedMatcher()* obtient de meilleurs résultats. Pour les différents paramètres, nous avons essayé d'activer le crosscheck avec le brute force matcher et plusieurs valeurs pour le nombre d'arbres et l'algorithme du Flann Matcher. Les meilleurs paramètres étaient l'algorithme 1 ainsi que 5 arbres du Flann matcher. Cependant, les résultats n'étaient toujours pas satisfaisants à notre goût, alors nous avons fait des recherches en ligne sur des améliorations possibles. Dans un blogue sur une implémentation de SIFT (https ://medium.com/@russmislam/implementing-sift-in-python-a-complete-guide-part-2-c4350274be2b), nous avons trouvé qu'il est possible de mesurer les distances des paires qui ont été trouvés et qu'il est avantageux de garder simplement les 'bonnes' paires, soit celles qui ont une plus petite distance. Notre mesure de similarité entre deux descripteurs consiste alors à trouver toutes les 'bonnes' paires et à calculer le nombre de paires trouvées, plus le nombre est élevé, plus les images sont similaires. Encore un fois, on peut trier les images en fonction de la mesure de similarité et trouver les cinq images les plus similaires à chaque image requêtes et calculer la précision top 1, top 3 et top 5.

Dernièrement, nous avons utilisé la méthode *selectROI()* d'OpenCV pour créer des boites englobantes manuellement pour chaque image. Les boites englobantes ont ensuite été sauvegardés dans la base de données pour ne pas avoir à les refaire manuellement à chaque fois. Finalement, nous avons réutilisé les implémentations de HOG et de SIFT définies précédemment pour recalculer leurs descripteur, similarité et précision top 1, top 3 et top 5.

En ce qui est de la requête de la fraise, nous avons tout d'abord testé avec HOG. Nous avons recalculé toutes les distances entre chaque requête et chaque image de la base de données, trié ces distances en ordre croissant par requête et pris la moyenne par rang de similarité avec les requêtes. Il est ensuite possible de réutiliser l'implémentation de HOG précédente pour calculer des distances à chaque image de la base de données et les comparer aux moyennes calculées. Pour SIFT, nous avons effectué les mêmes étapes, mais avec un changement de la métrique de comparaison pour utiliser le pourcentage de match trouvé.

2.6 Présentation des résultats de tests

Les résultats obtenus sont présentés dans les tableaux et les figures suivants. Le Tableau 1 présente le pourcentage de précision combiné de chacune des requêtes pour chaque algorithme. Ensuite, les figures 2.1 à 2.4 détaillent ces résultats en présentant le top 5 des images les plus similaires pour

chacune des requêtes. Pour les figures 2.1 à 2.4, la colonne de gauche contient les images requête alors que les suivantes contiennent les résultats les plus similaires, avec l'image la plus similaire dans la 2e colonne et la 5e plus similaire dans la 6e colonne. Le tableau 2 présente ensuite les résultats de similarité moyens des requêtes ainsi que ceux pour la requête fraise pour les algorithmes HOG et SIFT. Les figures 2.5 et 2.6 démontrent ensuite les résultats de la requête fraise dans le même format que les autres figures.

TABLEAU 1 – Résultats généraux

Algorithmes	Précision du Top 1 (%)	Précision du Top 3 (%)	Précision du Top 5 (%)
HOG	87.5	70.83	62.5
SIFT	37.5	33.33	32.5
HOG avec boites englobantes	87.5	58.33	50
SIFT avec boites englobantes	25	33.33	27.5

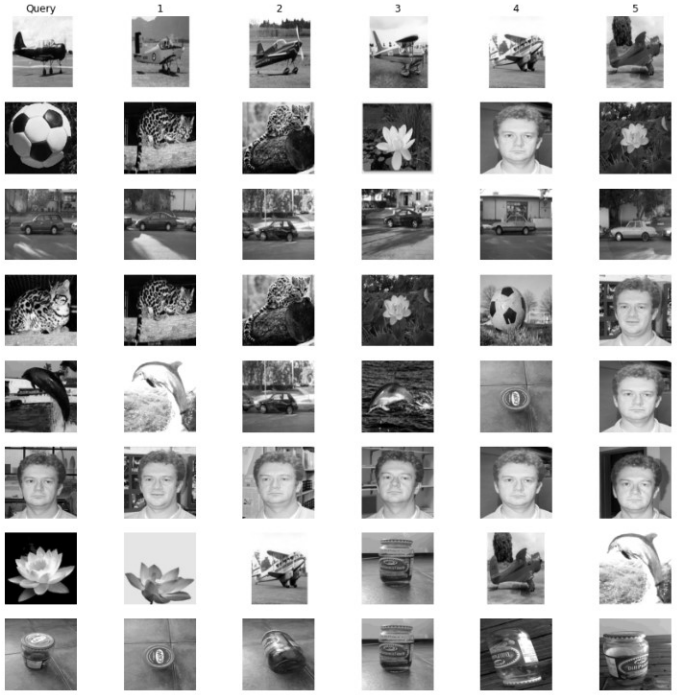

FIGURE 2.1 – Résultats de l'algorithme HOG pour chaque requête

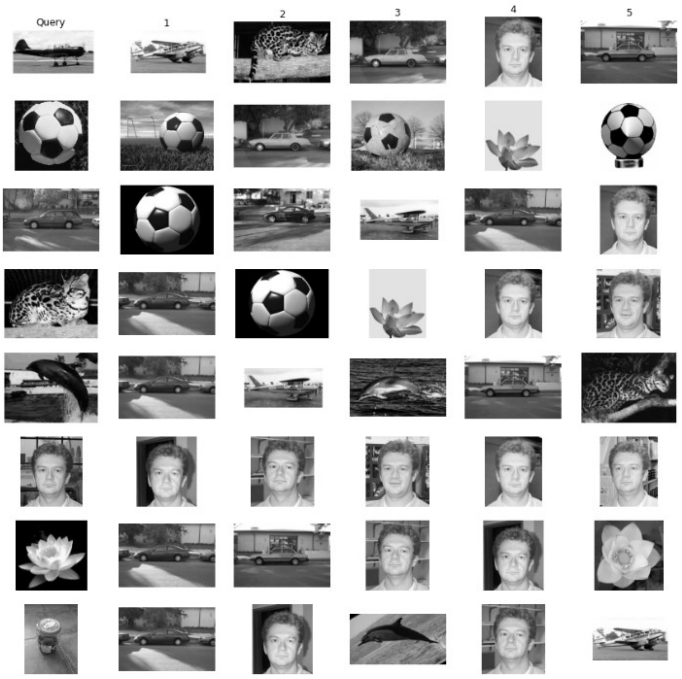

FIGURE 2.2 – Résultats de l'algorithme SIFT pour chaque requête

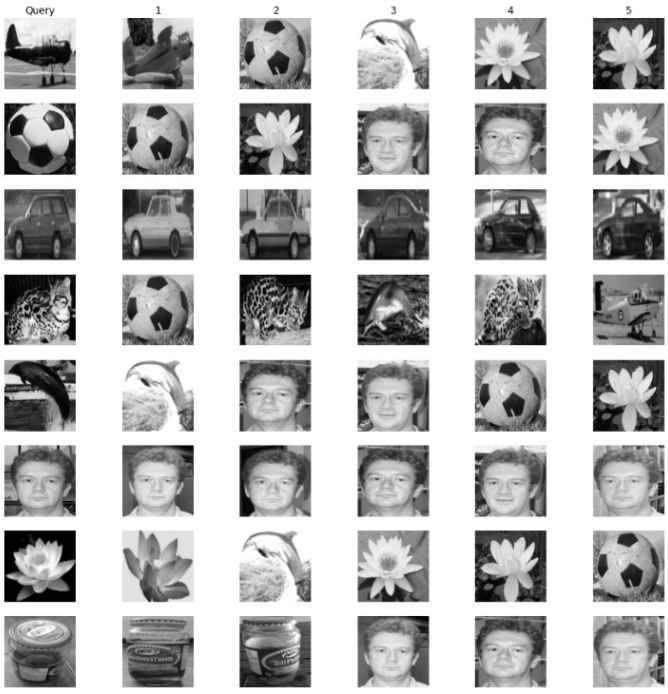

FIGURE 2.3 – Résultats de l'algorithme HOG avec boites englobantes pour chaque requête

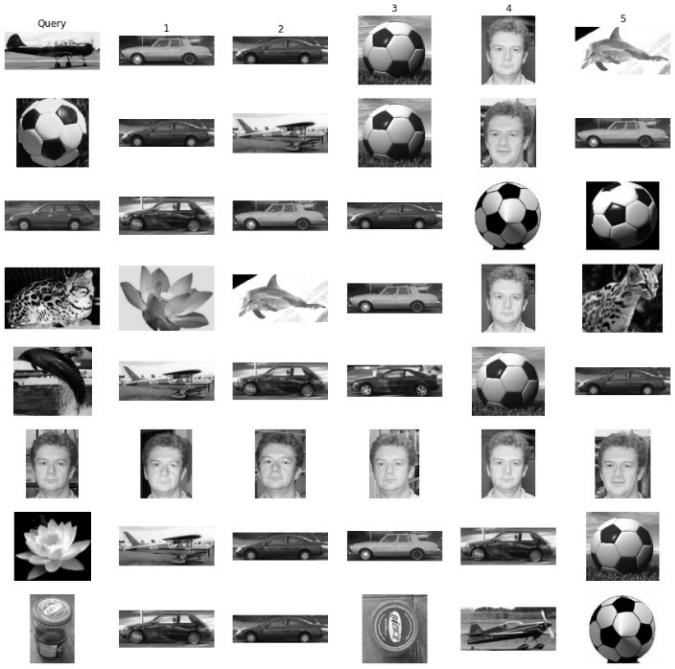

FIGURE 2.4 – Résultats de l'algorithme SIFT avec boites englobantes pour chaque requête

TABLEAU 2 – Résultats pour la requête de fraise

	Top 1	Top 3	Top 5
Moyenne des distances L1 entre les requêtes et images avec HOG	556850.5	568383.7	574364.44
Distances L1 entre la fraise et images avec HOG	575435.2	580304.06	581914.7
Moyenne des pourcentages de matchs entre les requêtes et images avec SIFT	0.11926132	0.10424358	0.09871974
Pourcentage de match entre la fraise et images avec SIFT	0.10152284	0.08426396	0.08121827

FIGURE 2.5 – Résultats de l'algorithme HOG pour la requête de fraise

FIGURE 2.6 – Résultats de l'algorithme SIFT pour la requête de fraise

2.7 Discussion des résultats et retour sur les hypothèses

Pour la première hypothèse, soit la détection d'image avec une rotation dans le plan de l'image, nous avons comparé les taux de succès sur la requête des dauphins, puisque plusieurs images dans la banque d'images sont très similaires à la requête du dauphin, mais avec une rotation dans le plan de l'image. Les figures 2.1 et 2.2 montrent que l'algorithme HOG a identifié correctement des dauphins en 1er et 3e, alors que SIFT a seulement identifié un dauphin en 3e. Cependant, il est difficile de conclure avec ce test et ainsi d'invalider notre hypothèse, considérant que SIFT a de la difficulté à reconnaitre des objets qui sont similaires, mais pas exactement les mêmes. Afin de résoudre ce problème, il faudrait comparer l'efficacité des deux algorithmes sur le même objet, mais avec des rotations dans le plan de l'image. Ensuite, nous avions également fait l'hypothèse que HOG sera meilleur que SIFT pour des rotations hors du plan de l'image, ce qui a pu être testé avec les images des cornichons. Il est possible de voir dans les figures 2.1 et 2.2 que HOG a identifiées avec succès les 5 images de cornichons alors que SIFT n'en a identifié aucun. Ceci confirme notre hypothèse que HOG est meilleur que SIFT pour des rotations hors plan, particulièrement lorsque l'arrière-plan reste le même.

Pour l'hypothèse de détection d'objets similaires contrairement à la détection d'objets d'une même catégorie, on s'attendait a voir que SIFT soit excellent pour reconnaitre le même objet, mais ait de la difficulté pour des objets de la même catégorie, alors que HOG devrait être meilleur pour des objets d'une même catégorie. En observant les figures 2.1 et 2.2, HOG a parfaitement identifié les 5 images pour les avions et les automobiles, qui sont tous des objets relativement similaires et avec un arrière-plan relativement similaire. Par contre, SIFT a identifié un avion en 1ere position seulement et des automobiles en 2e et 4e positions. Notre hypothèse que HOG est meilleur que SIFT pour reconnaitre des objets d'une même catégorie est donc validée. Par contre, les deux algorithmes ont identifié avec succès les 5 images du visage, ce qui nous permet de valider l'hypothèse que SIFT est bien meilleur avec le même objet qu'avec une catégorie d'objet, mais ne permet pas de conclure sur quel algorithme est le meilleur entre les deux pour retrouver un objet précis. Une solution à ce problème serait de poser des conditions plus difficiles sur ce même objet, comme des rotations ou des occlusions.

Pour l'hypothèse que SIFT aura de la difficulté avec des images requêtes trop simples et détectera à la place des images avec des motifs complexes, la figure 2.2 montre que SIFT a correctement identifié des ballons en 1ere, 3e et 5e position, et qu'aucune des images mal identifiées en 2e et 4e position ne sont des chats. Cette hypothèse est donc invalidée.

Finalement, pour l'hypothèse concernant les boites englobantes, il est possible de voir dans le tableau 1 que le taux de réussite de SIFT est égal ou moins bon après l'utilisation des boites englobantes, ce qui invalide notre hypothèse concernant l'utilisation des boites englobantes pour SIFT. Pour l'utilisation des boites englobantes avec HOG, il est possible de voir dans les figures 2.1 et 2.3 qu'après l'ajout de boites englobantes HOG a de moins bonnes performances pour les avions et les cornichons, la même performance pour les automobiles et de meilleures performances pour les lotus. Ceci valide

notre hypothèse que HOG puisse être meilleur ou moins bon après l'ajout de boites englobantes, selon si l'arrière-plan enlevé était similaire ou différent entre les images.

En ce qui concerne les fraises, nous pouvons voir les résultats dans le tableau 2. La méthode HOG retourne une distance L1 moyenne de 574364 pour la troisième position, ce qui est plus petit que la distance L1 obtenue pour la première position de la requête fraise, soit 575435. Cela nous dit que l'algorithme HOG est plus certain de la similarité des trois premières images lors de comparaison entre les requêtes et leurs images dans la base de données que la première image lors de la requête fraise. Cela confirme notre hypothèse et serait assez pour définir qu'il n'y a pas de fraise dans la base de données. Lorsque nous utilisons l'algorithme SIFT, nous obtenons des résultats légèrement différents. En effet, la meilleure valeur de similitude de la requête fraise, 0.1015, se situe entre la deuxième et troisième position des moyennes avec les autres requêtes, soit 0.1042 et 0.098 respectivement. Bien que cela suit le même principe qu'avec HOG, il y a plus d'ambiguïté.

3 Conclusion

Dans ce rapport, nous avons étudié deux techniques de description d'images, HOG et SIFT. Nous avons posé des hypothèses sur leur performance et validé ou infirmé ces hypothèses grâce à des expériences précises. Les résultats obtenus penchent principalement en faveur de HOG qui obtient des résultats supérieurs ou égaux à SIFT dans presque toutes les situations.