

INF6804 - TP 1

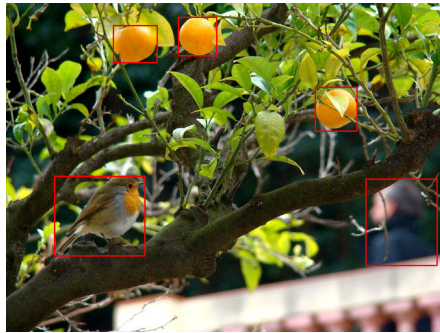
Grégoire Chapeaux - 2033122 — Mohamed Marwen Moslah - 2043911

February, 15th, 2021

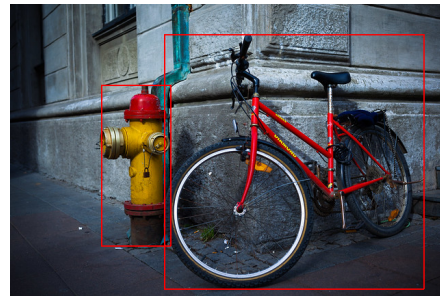
1 Partie 1

1.1 Manipulation et extraction des régions d'intérêt

Le code que nous avons implémenté pour cette partie repose sur deux méthodes : une méthode *draw_ROI(box, img, cover)* qui va dessiner la région délimitée par *box* sur l'image *img*, en recouvrant ou non la région (booléen *cover*, pour répondre à l'exigence de la couverture des camions), et une méthode principale *regions_of_interest(file_name, saving=False)* qui va parcourir le fichier *gt.json* pour rechercher les régions d'intérêt associées au fichier *file_name*, et les afficher à l'aide de la première méthode. Un paramètre permet par ailleurs de sauvegarder ou juste afficher le résultat.



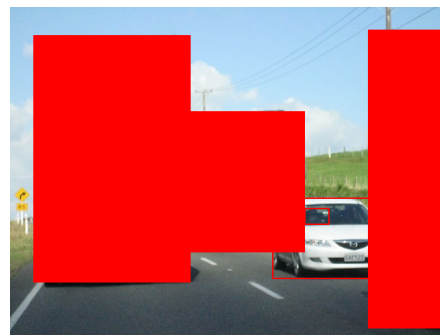
(a) Tree



(b) Bicycle



(c) Airplanes



(d) Road

Figure 1: Régions d'intérêt de différentes images

1.2 Description de régions d'intérêt par histogramme RGB

Le code de cette approche se base sur deux méthodes : une première méthode, `analysis_ROI(file_name, ROI_category, draw_hist)`, qui va parcourir le fichier `gt.json` pour lister les régions d'intérêt dans la catégorie `ROI_category` sur l'image `file_name`, et faire appel à la seconde méthode passée en paramètre `draw_hist`, ici `rgb_hist`, pour analyser ces régions d'intérêt.

Par exemple, pour analyser les régions de la catégorie `person` dans l'image `ski.jpg`, on exécute `analysis_ROI("ski", "person", rgb_hist)`, nous donnant les résultats normalisés suivants :

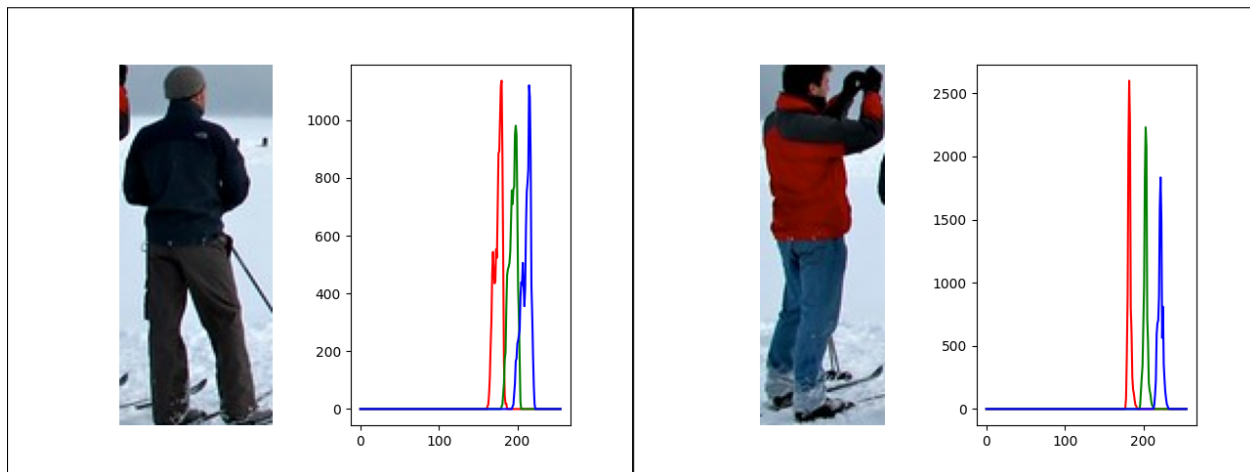


Figure 2: Analyse RGB des régions d'intérêt "person" sur l'image "ski"

Les courbes rouge, verte et bleue correspondent à leurs couleurs respectives. On remarque que les histogrammes sont dans l'ensemble plutôt semblables, malgré les différences apparentes entre les deux images. Le degré de similitude obtenu est tout à fait attendu et axiomatique étant donné que les deux images de la figure 2 partagent pratiquement le même arrière-plan (paysage de neige) avec un contraste entre la personne figurant au premier plan et son arrière-plan, ainsi qu'une bonne luminosité.

Cependant, on observe quand même une légère différence dans les hauteurs des pics de bleu, de vert et de rouge : cette nuance est très certainement issue d'une légère différence de la distribution d'intensité des couleurs, particulièrement figurant dans les manteaux des deux personnes. En résumé, les éléments qui influencent sur ces histogrammes sont le contraste, la luminosité, et la distribution de l'intensité des pixels de l'image.

1.3 Description de régions d'intérêt par histogramme HSV

On applique pour une analyse hsv la même approche que pour la description rgb, à ceci près que la fonction d'affichage de l'histogramme est cette fois-ci *hsv_hist*.

On exécute peut ainsi comparer les analyses RGB et HSV sur deux images :

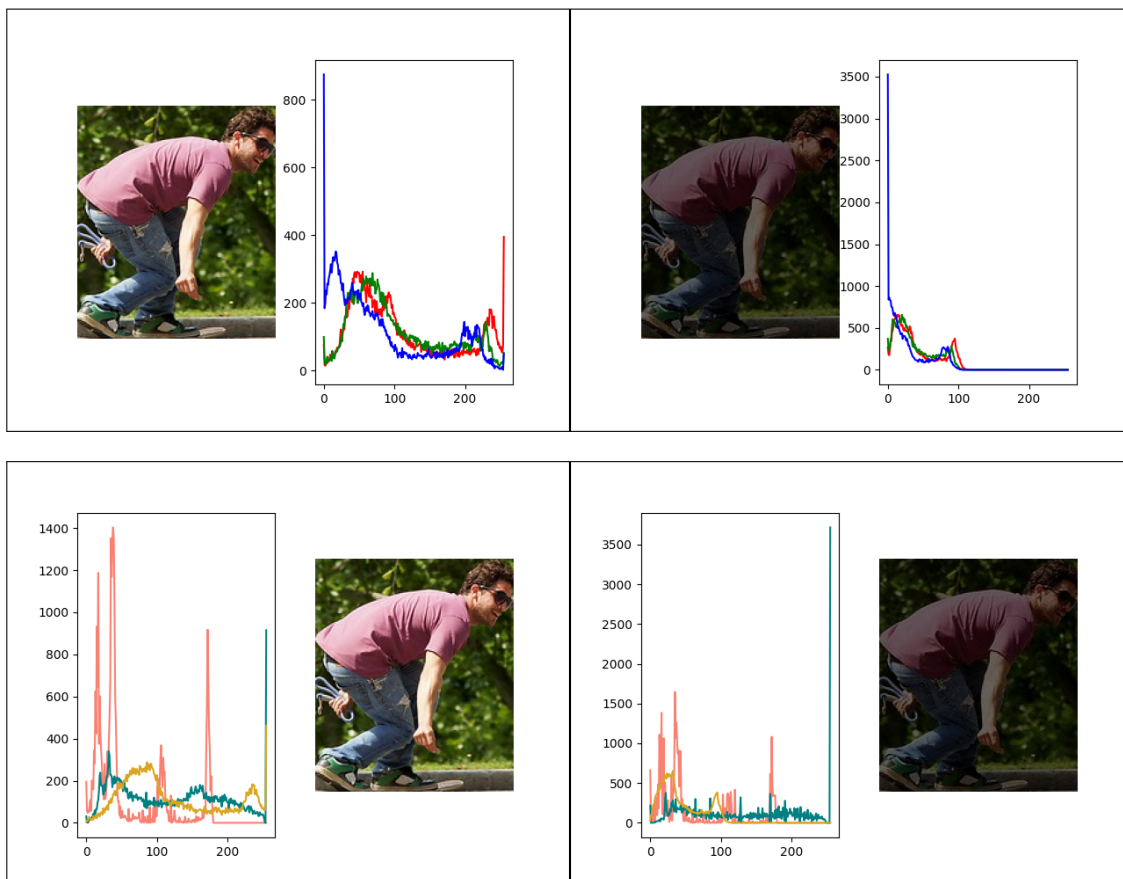


Figure 3: Analyses RGB et HSV de la région d'intérêt "personne" sur les images "skate1" et "skate2"

Les courbes rouge, verte et bleue correspondent à leurs couleurs respectives, tandis que les courbes rose, bleu-gris et orange correspondent respectivement à la teinte (Hue), la saturation (Saturation) et la valeur (Value). On analyse ici deux exemplaires d'une même image, avec deux luminosités différentes, la seconde étant bien plus sombre que la première.

On remarque dans les analyses RGB que les histogrammes, bien que gardant globalement une même tendance ($R > G > B$), sont très différents dans leurs amplitudes de couleur, et qu'il est difficile en ne comparant que les histogrammes RGB de conclure que les deux images montrent la même scène.

En revanche, la comparaison des histogrammes HSV nous permet de noter que le changement de luminosité influe peu sur la *value* (orange), légèrement sur la *saturation* (vert-bleu), et beaucoup sur la *hue* (rose). On peut en tirer une conclusion très intéressante : la valeur ne varie pas et la saturation varie peu avec la luminosité. Un même objet aura donc, peu importe la luminosité, une valeur identique et une saturation peu variée, contrairement à la teinte. Une analyse HSV pourrait par exemple aider lors d'un suivi d'objet entre plusieurs scènes de luminosité différente, et donc dans ce contexte, l'analyse via HSV donne une meilleure

aptitude en termes de stabilité et de généralisation pour des scènes dont leur luminosité varie, qu’une analyse via RGB.

2 Partie 2

2.1 Présentation des approches à comparer

On va caractériser chaque descripteur par deux approches similaires : pour une image requête donnée, on va itérer sur la base de données d’images à disposition, et calculer un score de similarité entre l’image requête et l’image de la base de données, variant entre 0 (les images ont une similarité nulle) et 1 (les images ont une similarité semblable). La nuance entre les deux approches va se faire à cet endroit sur la manière dont est calculé ce score de similarité :

2.1.1 Similarité pour le descripteur RGB

Le descripteur RGB se base sur les valeurs de Rouge, Vert, Bleu de l’image : pour chaque image, il lit celle-ci et retourne un vecteur par canal de couleur.

On va alors calculer la distance euclidienne moyenne d entre les vecteurs respectifs de chaque image, et définir la similarité suivante : $sim = \frac{1}{1+d}$.

Ainsi, pour deux images identiques, les valeurs de R, G, B sont les mêmes, donc la distance est nulle, et la similarité vaut 1. A l’inverse, plus deux images présentent des R, G, B différents, plus la distance sera importante, plus la similarité tendra vers 0.

Nous avons à l’origine considéré l’utilisation de la fonction cosinus pour la similarité. Cependant, l’inconvénient de cette approche est qu’elle ne prend pas en compte la magnitude de ces vecteurs : en d’autres termes, si une image présente exactement deux fois plus de Rouge, de Bleu et de Vert qu’une autre image, ces deux images sont toutefois similaires.

2.1.2 Similarité pour le descripteur ORB

ORB (ou bien Oriented FAST and Rotated BRIEF), est un descripteur d’image qui combine la technique FAST et BRIEF qui cherchent à trouver les keypoints dans une image (keypoints ou bien points-clé veut dire les points qui ont à leur alentours les pixels les plus importants en termes de variation d’une caractéristique particulière : par exemple la luminosité). Dans notre contexte CBIR, ces keypoints seront utilisés pour comparer ceux d’une image requête à ceux d’une image faisant partie de la base de données afin de mesurer le degré de la similitude et de l’invariance entre 2 images via une métrique de distance entre ces points, et pour déterminer plus tard si ces 2 images appartiennent à la même classe ou pas.

Les systèmes retournent deux résultats : la classe de l’image avec la plus grande similarité avec l’image requête, et un résultat "topk" : parmi les k meilleures images, la classe la plus représentée. Ce dernier résultat serait toutefois bien plus pertinent avec une base de données plus conséquente.

2.2 Hypothèse de performance pour des cas spécifiques

En se référant au cours, on identifie deux cas d’utilisation :

- Premier cas d’utilisation : Si le contenu des régions comparées est relativement uniforme, alors la meilleure approche à utiliser est le descripteur par histogramme de couleurs RGB (desc-hist-RGB). En effet, desc-hist-RGB est plus performant que ORB en ce point, car il a plus de sensibilité à détecter la variation et/ou l’uniformité des couleurs que l’ORB. On s’attend d’avoir un meilleur matching avec ORB entre une image strawberry_query et les images de database strawberry qu’avec desc-hist-RGB.
- Deuxième cas d’utilisation : On s’attend à avoir une meilleure reconnaissance des formes de "patterns" par l’ORB que par celui RGB comme le cas des hexagones dans les images de la ball. Donc on s’attend que l’ORB reconnait mieux la texture des formes qui se répètent d’une image à une autre de la même classe.

2.3 Description des expériences, des données et critères d'évaluation

- Expérience 1 : tester la sensibilité à la variation/uniformité des couleurs
- Expérience 2 : tester la sensibilité à la détection des patterns

La méthode des expériences est la suivante : étant donné le cas spécifique que l'on cherche à vérifier (exemple : capacité à reconnaître une image d'après sa couleur), on va présenter aux deux systèmes une image correspondante (par exemple strawberry pour la couleur) et l'on va observer si le système la reconnaît ou non.

Cas d'utilisation	Image requête	Classe par RGB	Classe par ORB
Couleur	strawberry	strawberry	airplane
	lotus	dolphin	ball
Patterns	cat	dolphin	airplane
	ball	dolphin	ball

La plus grande difficulté rencontrée est la petite taille de la base de données, induisant un manque d'informations avec lesquelles comparer les requêtes.

La métrique utilisée pour ces expériences est la métrique de similarité définie plus haut. Chaque image requête a été choisie en fonction de son intérêt pour le cas d'utilisation : strawberry et lotus pour leur couleur caractéristique (rouge et blanc), cat et ball pour leurs patterns spécifiques (taches du chat et hexagones sur le ballon).

On peut mesurer les performances du descripteur à l'aide de l'approche topk en calculant le nombre d'occurrences d'une image de la classe prédite parmi les k premières images de la recherche de similarité.

```
Query : data/part2/ball_query.jpg
| BEST PICTURE APPROACH : class of the most similar picture
| Predicted class : ball
|
| TOPK APPROACH : most represented class in the top k pictures
| Predicted class ball : 100. %
```

Figure 4: Résultat d'une prédiction sur ball avec ORB

2.4 Description des deux implémentations utilisées

Le descripteur ORB a été modélisé à l'aide de la page web suivante : Tutoriel ORB