

---

# Travail pratique #1

**POLYTECHNIQUE  
MONTRÉAL**  
UNIVERSITÉ  
D'INGÉNIERIE



---

INF6804 - Vision par ordinateur

9 février 2024

---

Émile Poirier

2014031

Ariste Kwawe-K-Arnov

2317869

---

## Table des matières

Introduction .....	3
Description du problème .....	4
1. Présentation des deux approches à comparer .....	4
2. Hypothèses de performance pour des cas spécifiques .....	5
3. Hypothèses de performance concernant les boîtes englobantes .....	5
Méthodologie .....	6
4. Description des expériences, des données et critères d'évaluation .....	6
5. Description des deux implémentations utilisées .....	7
Résultats .....	10
6. Présentation des résultats de tests .....	10
7. Discussion des résultats et retour sur les hypothèses .....	11
Conclusion .....	12
Références .....	13

# Introduction

Ce premier laboratoire a pour objectif de nous familiariser avec des méthodes de description et de comparaison de régions d'intérêts dans des images plus précisément, les textures. Pour étudier ce sujet, l'objectif de ce travail est de faire correspondre des images d'une base de données à des images de référence. Pour ce faire, nous étudierons le fonctionnement, l'implémentation et les performances des deux méthodes; une basée sur les matrices de co-occurrences et l'autre sur les modèles binaires locaux. L'objectif de ce laboratoire est de nous familiariser avec les deux méthodes, apprendre comment comparer des descripteurs de caractéristiques et comparer les forces et faiblesses des différentes approches.

# Description du problème

## 1. Présentation des deux approches à comparer

Les matrices de cooccurrences et les motifs binaires locaux (LBP) font partie en vision numérique des descripteurs globaux qui permettent d'analyser la texture d'une image, chacun de ces derniers possède son propre principe et est plus ou moins adapté suivant les besoins.

### Matrice de cooccurrences

Le principe consiste à rechercher dans une image  $I$ , les répétitions de niveaux de gris d'un pixel  $I(i, j)$  suivant une distance  $dx$  et une direction  $dy$  (l'angle).

Une image avec  $N$  différents niveaux de pixels produiront une matrice de cooccurrences carrée  $N \times N$ .

Dans la matrice de cooccurrences, la valeur du pixel de coordonnées  $(i, j)$  représente le nombre de fois que ce pixel est présent dans l'image.

En fonction de la distance et de la direction, on peut calculer plusieurs matrices de cooccurrences.

$$I_{dx,dy}(i, j) = \sum_{x=1}^n \sum_{y=1}^m \begin{cases} 1, & \text{si } I(x, y) = i \text{ et } I(x + dx, y + dy) = j \\ 0, & \text{sinon} \end{cases}$$

### Le descripteur LBP s'obtient en deux étapes [2] [3]:

Il examine tout d'abord un nombre de pixels  $P$  voisins (une interpolation est faite si le pixel  $p$  tombe entre deux pixels voisins sur le cercle) entourant un pixel central obtenu suivant un cercle de rayon  $R$  dans une image de façon à ce que tous les voisins qui lui sont supérieurs ou égaux valent 1 et 0 sinon.

On obtient alors un code binaire de l'image résultante. La valeur  $LBP_{(P, R)}$  de chaque pixel central est alors obtenue par la conversion de ce code binaire dans la base décimale. On termine en calculant l'histogramme de chaque image LBP qui forme alors un vecteur d'entiers caractéristique de notre image et qui représente le nombre d'occurrences de chaque code binaire dans une région de l'image.

$$BP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(i_p - i_c) 2^p$$

$$s(x) = \begin{cases} 1, & \text{si } x \geq 0 \\ 0, & \text{sinon} \end{cases}$$

## 2. Hypothèses de performance pour des cas spécifiques

Dans le cas où le contenu des images sont relativement uniformes, des ensembles de valeurs de pixels ou toutes ces valeurs peuvent être égales, que ce soit alors la matrice de cooccurrences ou les motifs binaires locaux, chacun donnera un bon résultat : pour les matrices de cooccurrences par exemple, on va trouver le ou les mêmes couple(s) d'indices  $(i, j)$  très élevés alors que les valeurs LBP seront aussi assez proches, on peut observer une matrice identité de code binaire.

Si les régions à comparer dans les images ont subi des rotations dans le plan ou non, alors les deux méthodes seront très mauvaises, car l'arrangement des pixels dans les images ne sera plus le même, autrement dit, les informations spatiales seront modifiées. Il sera de même si le contraste varie beaucoup entre les images, la luminosité des pixels change et leur arrangement par conséquent.

## 3. Hypothèses de performance concernant les boîtes englobantes

Considérant que nous nous intéressons aux textures de sujet se trouvant dans les images et que les arrière-plans ne contiennent pas d'élément de similarité si nous avons accès à la boîte englobante entourant l'objet d'intérêt alors on s'attend à un meilleur résultat des deux méthodes. Cela s'explique par le fait qu'on ne s'intéresse plus à l'image tout entière, la grande majorité de l'arrière-plan de l'image est ainsi supprimée, ce qui améliore la précision et le temps de calcul.

Par contre, dans certains cas où l'arrière-plan de l'image requête possède des caractéristiques qui apparaissaient également dans les images correspondantes de la base de données, on s'attendrait alors à de moins bons résultats avec les boîtes englobantes.

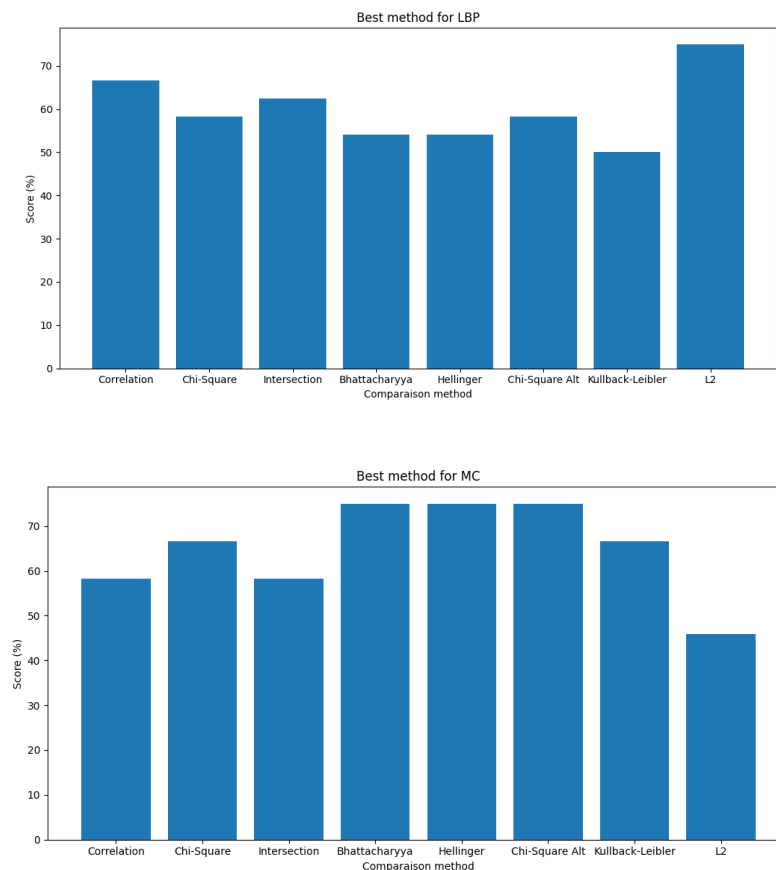
# Méthodologie

## 4. Description des expériences, des données et critères d'évaluation

Pour évaluer les performances des deux méthodes, notre stratégie était d'obtenir les trois meilleures images de la base de données correspondant à une requête et de donner un score allant de 0 à 3 selon le nombre d'images correctement identifiées. En observant les images à comparer, nous serons ainsi capables de connaître l'effet de l'uniformité, des rotations dans le plan, du contraste, l'échelle ou toute autre caractéristique de l'image. Par exemple, en étudiant les résultats sur les images de dauphins, nous serons capables de faire des conclusions sur l'effet de la luminosité et de la rotation.

Pour comparer l'effet des paramètres afin de déterminer les configurations optimales, nous utiliserons le score moyen pour toutes les requêtes. Un score de 100% indique donc que la moyenne des images correctement identifiée était de 3/3.

Pour les deux méthodes, nous allons comparer les histogrammes générés à partir des descripteurs. Nous avons effectué une comparaison des différentes métriques pour chaque méthode afin de savoir laquelle était la plus appropriée et voici les résultats. Voici des résultats pour les mêmes paramètres de chaque méthode et pour différentes métriques de comparaison :



Nous avons donc choisi la métrique  $L_2$  pour la méthode LBP et la métrique Bhattacharya pour la méthode MC pour comparer nos histogrammes. On remarque par contre que les métriques donnent des résultats assez similaires et que pour différents échantillons de données, les résultats auraient pu être différents.

Pour étudier l'effet de boîtes englobantes, nous avons décidé de choisir les paramètres optimaux à partir des images originales et par la suite aller chercher les résultats sur l'ensemble de données avec et sans boîte afin de comparer l'effet. Cependant, une autre stratégie aurait pu être d'obtenir les paramètres optimaux pour chaque ensemble de données et ensuite comparer les résultats.

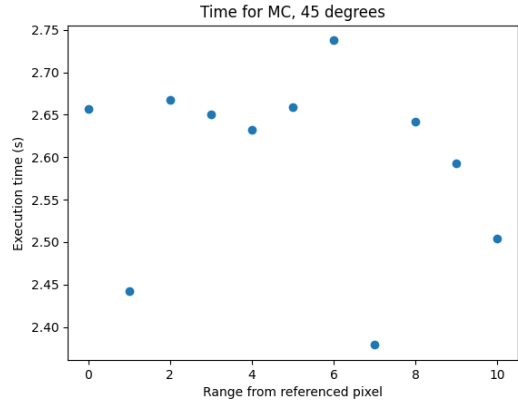
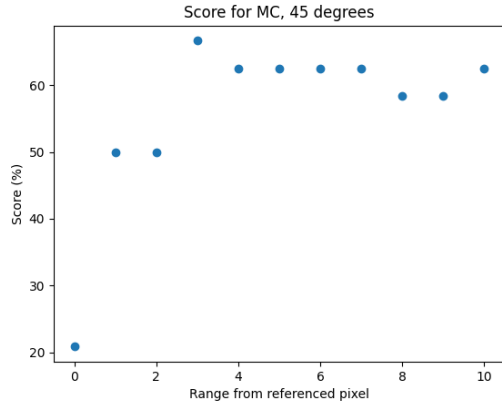
Pour la méthode LBP, nous avons uniquement étudié les résultats sur les images converties en niveau de gris à l'aide de la fonction *grayscale* de *openCV2*. Pour la méthode MC, nous avons décidé de décomposer chaque image par ses trois canaux pour trouver trois descripteurs. Une difficulté rencontrée fut que certaines images reçues ne possédaient qu'un canal. Ainsi, nous avons dû adapter le code pour gérer les différents cas en fonction de l'image traitée.

## 5. Description des deux implémentations utilisées

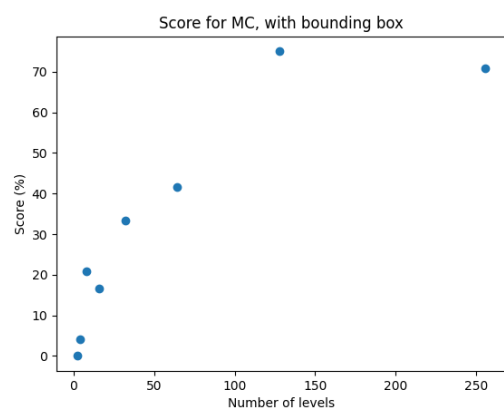
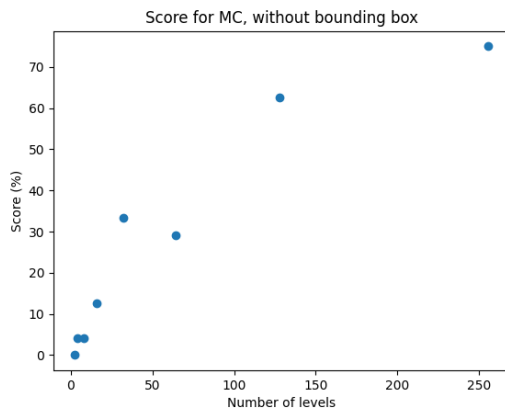
L'implémentation était la même pour les deux approches étudiées dans le laboratoire afin de conserver une consistance qui permettra de mieux de comparer les résultats. Pour les deux cas, un script écrit en Python vient initialement lire les images et les sauvegarder avec leurs couleurs originales ainsi qu'en noir et blanc. Par la suite, on vient extraire des histogrammes pour chaque image selon la méthode utilisée.

### **Matrice de Co-Occurrences :**

Pour la méthode basée sur les matrices de cooccurrences, on commence par séparer l'image reçue en entrée en trois matrices, une pour chaque canal de couleur RGB. Dans le cas où l'image ne possède pas de canaux (quand l'image reçue est en noir et blanc) on utilise alors la même matrice pour au lieu des canaux rouge, vert et bleu. La librairie open source scikit-image est utilisée pour calculer la matrice de co-occurrence, spécifiquement la fonction *skimage.feature.graycomatrix*. Cette fonction prend en entrée une matrice de données, la distance et l'angle pour la comparaison et la quantification de niveaux. En réalisant un grand nombre de simulations pour différentes valeurs de paramètre, nous avons conclu que la valeur de distance n'avait pas de grand impact sur les performances. Comme l'indiquent les figures suivantes, pour un angle de 45 degrés, la valeur optimale de rayon est de 3, mais les valeurs supérieures ont des scores assez similaires et le temps d'exécution n'est pas très affecté par ce paramètre.



Pour le choix des niveaux, nous avons testé pour tous les facteurs de N allant de 0 à 8 pour  $2^N$ . . Voici les résultats obtenus :



On remarque que plus le nombre de niveaux est élevé, plus le score est bon. Cependant, dans le cas où nous utilisons les boîtes englobantes, le score est légèrement plus élevé avec 128 niveaux. On conclut donc qu'un grand nombre de niveaux est plus idéal pour créer des descripteurs avec la méthode de matrices de co-occurrence.

La référence [4] (An integrated Color and Intensity Co-occurrence Matrix) utilise une somme de différentes matrices de Co-occurrence donc c'est avec cette stratégie que nous avons décidé d'aller puisque les résultats de l'article semblaient satisfaisants pour une petite quantité de calcul. Après avoir extrait les matrices pour chaque canal de couleur, nous avons trouvé qu'additionner ces matrices avec des poids différents permettait d'augmenter les performances. Nous n'avons trouvé que des de tripler le poids de la matrice issu du canal rouge avec le meilleur effet. En effet, en étudiant les images de la base de données, nous avons remarqué que plusieurs images, telles que les avions, lotus ou visages, comportaient beaucoup de jaune et rouge, couleurs qui sont affectées par le canal R. Finalement, nous avons trouvé par simulation que la combinaison d'angle optimale pour cette stratégie était :

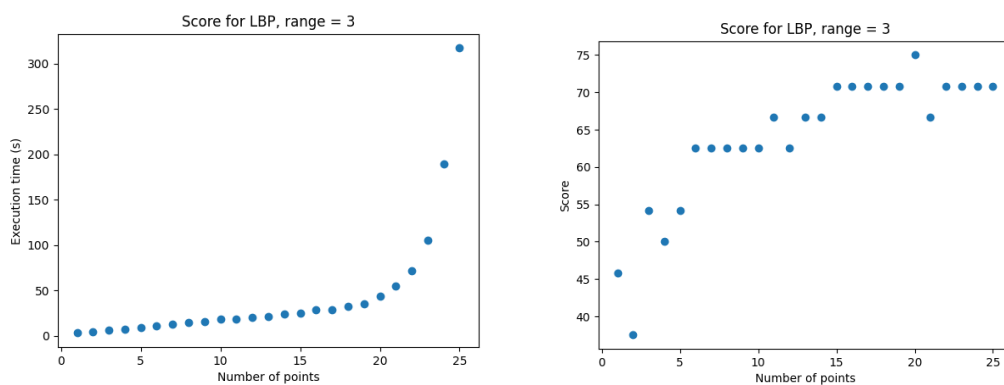
$$\theta_R = \frac{\pi}{3}, \theta_G = \frac{5\pi}{6}, \theta_B = \frac{5\pi}{6}$$



Il est important de noter que ces valeurs sont peut-être seulement optimales dans notre situation puisque notre base de données est beaucoup trop petite pour extraire des statistiques qui décrivent bien la méthode et non le comportement de l'échantillon étudié.

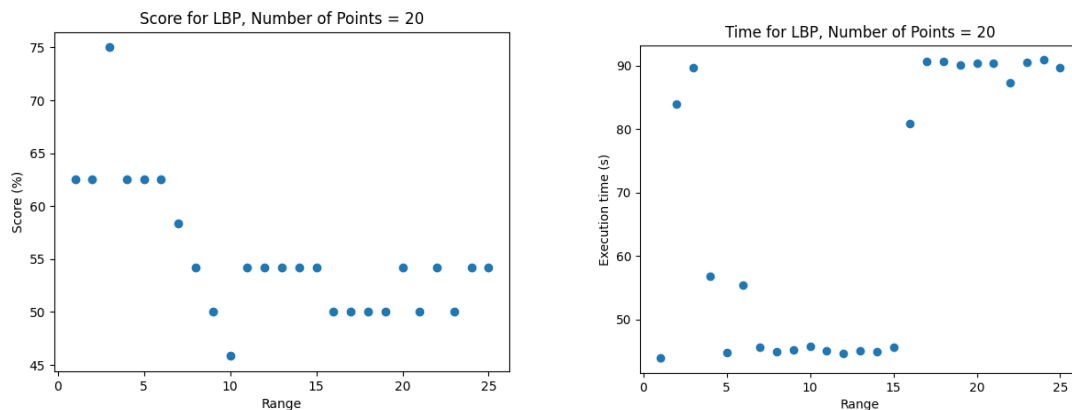
### Modèles binaires locaux :

Pour la méthode de *Local Binary Pattern*, la stratégie était similaire, nous avons testé plusieurs valeurs de paramètre pour le nombre de points ainsi que pour la distance du pixel de référence. Une partie du code est tirée du lien suivant <https://www.youtube.com/watch?v=zUdEaE8Kpac&list=PLMp-jYNu7MGL73kvx23074aZwKlvYcpwl&index=61&t=219s>.



Pour le nombre de points, nous avons obtenu que plus le nombre de points est élevé, plus le score soit bon, ce qui concorde avec la théorie puisque nous obtenons des codes définissant les textures des objets avec plus de détail. En revanche, le temps d'exécution augmente exponentiellement avec le nombre de points, avec du traitement pouvant aller à plusieurs minutes pour passer à travers toutes les images. Considérant nos besoins, nous avons jugé que 20 points étaient suffisants pour notre utilisation, ce qui donne un temps de recherche à travers la base de données d'environ 1 minute.

Pour l'optimisation du rayon, nous avons obtenu que la distance optimale fût de 3 pixels, et que pour ce paramètre, l'augmenté n'affecte pas les performances et que pour plus de 15 de rayon, le temps d'exécution augmente considérablement.



# Résultats

## 6. Présentation des résultats de tests

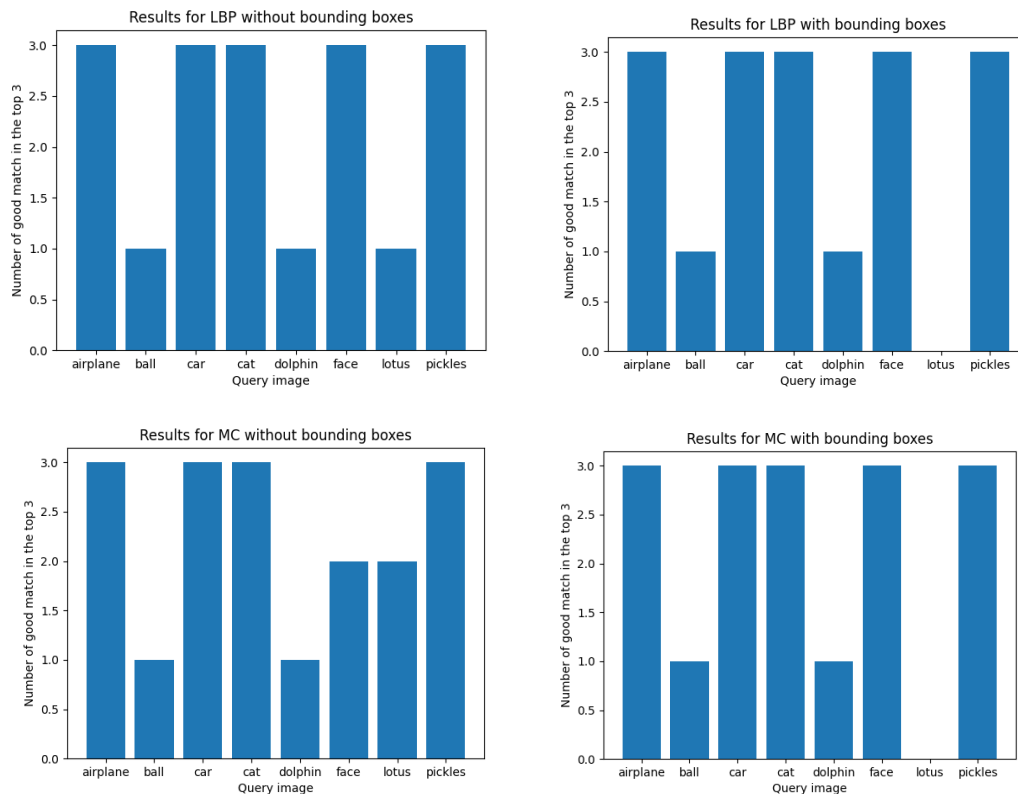


Tableau 1 Résultats moyens des différentes méthodes

Méthode	Moyenne des scores	Temps d'exécution (s)
MC	75	1.800
LBP	75	36.120
MC avec boîtes englobantes	70.83	0.695
LBP avec boîtes englobantes	70.83	13.347

Tableau 2 Propriétés uniques des différentes images

Image de référence	Particularité
Airplane	Changement dans les couleurs
Ball	Rotation et translation dans le plan
Car	Aucune couleur
Cat	Différente position avec texture unique
Dolphin	Rotation et changement de luminosité
Face	Changement d'arrière-plan
Lotus	Changement de couleur et arrière-plan
Pickles	Rotation hors plan et changement arrière-plan

## 7. Discussion des résultats et retour sur les hypothèses

En comparant uniquement les résultats moyens, on constate que les deux méthodes utilisées donnent exactement les mêmes résultats, avec et sans boîte englobante. La différence majeure entre les deux méthodes est par contre le temps d'exécution; la méthode basée sur les binaires locaux est beaucoup plus lente, soit environ par un facteur de 20. En revanche, la méthode LBP était plus facile à implémenter et à optimiser puisqu'on avait moins de paramètres à ajuster.

On considère le couple (A, B) tel que A représente le score pour les motifs binaires locaux et B pour la matrice de cooccurrences.

S'agissant de l'uniformité, on constate qu'on a de bons résultats pour les images cat (3/3, 2/3), airplane (3/3, 3/3), car (3/3, 3/3). Le LBP semble meilleur que la matrice de cooccurrences. On observe dans ces trois cas les éléments l'élément d'intérêt dans les images requêtes sont uniformes avec celles des images de la base de données, ce qui confirme bien notre hypothèse dans le cas où un groupe d'éléments est uniforme dans l'image. On note aussi que bien évidemment le résultat est le même lorsque les boîtes englobantes sont utilisées.

Pour le cas du contraste, on observe étonnamment que les images lotus donnent un bon score sans boîte englobante, certainement parce que l'arrière-plan de quelques-unes de ces images tend à se ressembler. Cependant, en utilisant les boîtes englobantes sur ces images, notre hypothèse est bien confirmée avec un score (0, 0), elles présentent un changement de contraste, auquel nos deux méthodes sont assez sensibles.

Pour ce qui concerne les résultats sur « *dolphin\_query.png* », nous avons obtenu un score de 1/3 pour les deux méthodes, avec et sans boîtes englobantes. On conclut donc que les deux méthodes utilisées n'étaient pas appropriées ce groupe d'images. La caractéristique unique de cet ensemble d'images était les changements de luminosité importants et la rotation dans le plan. Ainsi, il faudrait un descripteur qui ne dépend pas de la valeur absolue des pixels, mais plutôt de la valeur relative de ceux-ci par rapport à l'image de référence. Une solution serait d'utiliser un descripteur de contour pour identifier la forme du dauphin dans l'image. Ceci permettrait de ne pas dépendre sur la luminosité et l'orientation du sujet.

Pour améliorer nos algorithmes, une bonne stratégie serait de prendre plusieurs descripteurs différents et d'effectuer une recherche permettant de minimiser la distance entre plusieurs caractéristiques des images à comparer. Ceci permettrait d'avoir des résultats beaucoup plus robustes, mais en revanche cela augmenterait considérablement le temps de traitement.

## Conclusion

En résumé, il était question pour nous dans ce laboratoire de faire de l'analyse de texture. Partant des images de référence, on obtient pour chacune leur descripteur respectivement un histogramme à l'aide des méthodes de la matrice de cooccurrences et des motifs binaires locaux. On effectue le même traitement sur des images dans une base données. Par la suite on a comparé les descripteurs pour faire correspondre à chaque image requête sa correspondance dans la base de données. Les résultats sont assez bons pour chacune des deux méthodes, mais le LBP semble donner un meilleur résultat. Partant des hypothèses comme les boîtes englobantes, l'uniformité, le contraste ou le changement d'échelle, on constate que ces deux méthodes sont très sensibles aux deux dernières.

## Références

- [1] scikit-image, GLCM Texture Features. Consulté à l'adresse : [https://scikit-image.org/docs/stable/auto\\_examples/features\\_detection/plot\\_glcmm.html](https://scikit-image.org/docs/stable/auto_examples/features_detection/plot_glcmm.html)
- [2] Gounane, A et Sadi, S. Mémoire de fin d'études Contribution de l'approche LBP à la classification des images multispectrales. Application aux données TM de la région du Hoggar. Consulté à l'adresse : <https://dspace.ummto.dz/server/api/core/bitstreams/8212727a-ad6b-4635-99fd-6713d56de140/content>.
- [3] Bilodeau, G. INF6804 - Vision par ordinateur.
- [4] Vadivel, A. Sural, S. Majumdar, A.K. Pattern Recognition Letters, Volume 28, Issue 8, Pages 974-983 (1 Juin 2007). An integrated Color and Intensity Co-occurrence Matrix. DOI: <https://doi.org/10.1016/j.patrec.2007.01.004>
- [5] OpenCV, OpenCV Tutorials. Consulté à l'adresse : [https://docs.opencv.org/4.0.0/d9/df8/tutorial\\_root.html](https://docs.opencv.org/4.0.0/d9/df8/tutorial_root.html)