



INF6804

Vision par ordinateur

TP 1

Description et comparaison de régions d'intérêt

Félix Auger (2067772)

Arnaud Dalbec-Constant (2014288)

9 février 2024

Introduction

Le but de ce TP est de caractériser deux méthodes de description de régions d'intérêt dans des images. Ces deux méthodes sont les Matrices de Co-occurrence (MC) et les Modèles Binaires Locaux (LBP). Elles seront comparées afin de déterminer laquelle est la meilleure et dans quelles circonstances.

Présentation des deux approches à comparer

Matrices de co-occurrence (MC)

La méthode des matrices de co-occurrence ou co-occurrence matrix (CM) enregistre le nombre de fois que deux couleurs (valeurs d'intensité) i et j ont la même position relative dans une image. Pour appliquer la méthode, on détermine un certain décalage (dx, dy) , ou encore une direction et une distance. Ensuite, on examine si l'intensité de chaque pixel de l'image correspond à celle de son pixel associé dans la direction et à la distance déterminée précédemment. On obtient une matrice de co-occurrence $C_{dx, dy}(i, j)$ décrivant le nombre d'occurrences de chaque paire selon l'équation suivante, où n et m représentent la taille de l'image.

$$C_{dx, dy}(i, j) = \sum_{x=1}^n \sum_{y=1}^m \begin{cases} 1, & \text{if } I(x, y) = i \text{ and } I(x + dx, y + dy) = j \\ 0, & \text{otherwise} \end{cases}$$

Cette méthode peut être utilisée sur des images en teintes grises ou sur chaque canal (RGB) d'une image en couleurs.

Modèles binaires locaux (LBP)

La méthode des modèles binaires locaux (LBP) sert à identifier des textures dans des images. Pour chaque pixel observé, la méthode repère les pixels sur la circonférence d'un cercle de rayon arbitraire et compare l'intensité des pixels avec celui du centre. Ces résultats sont encodés par un 1 ou un 0. Cette liste de 0 et 1 peut être visualisée sous forme d'histogramme en sommant les résultats de chaque point évalué. Cette méthode fonctionne uniquement pour des images en teintes de gris. La valeur d'intensité d'un pixel i_p , avec $p \in P$ dans le voisinage d'un pixel c à la position (x_c, y_c) , est comparée à la valeur du pixel c , soit i_c . Le code binaire résultant correspond à :

$$BP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(i_p - i_c) 2^p \quad s(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

P étant une liste de pixels également espacés sur un cercle de rayon R . La Figure 1 montre comment différents types de textures peuvent être reconnus par cette méthode.

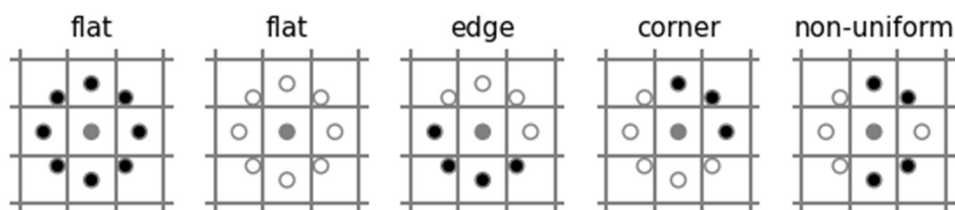


Figure 1. Représentation des différents types de texture observés par LBP. [1]

Hypothèses

Hypothèses de performance pour des cas spécifiques

Le premier cas d'utilisation est lorsque le contenu des régions comparées est relativement uniforme. La méthode des Matrices de Co-occurrence (MC) semble être mieux adaptée dans cette situation, car elle est capable de capturer les relations de co-occurrence entre les pixels ayant la même intensité, permettant ainsi une caractérisation fine des structures homogènes présentes dans les régions d'intérêts.

Le deuxième cas spécifique est lorsque le contenu des images montre une variation de luminosité importante. La méthode des Modèles Binaires Locaux (LBP) semble être mieux adaptée dans ce cas, puisqu'elle se concentre sur l'identification des textures en comparant les valeurs des pixels sur la circonférence d'un cercle. Cette approche peut être plus robuste face aux variations de luminosité, car elle se base sur des relations entre les pixels plutôt que sur des valeurs absolues.

La Figure 2 montre trois des images de dauphins provenant de la base de données. On voit bien que le niveau de luminosité diffère grandement d'une image à l'autre, donc il serait intéressant d'utiliser ces images pour tester ce cas spécifique en plus de modifier la luminosité des autres images.



Figure 2: Variation de luminosité sur les images de dauphins tirées de la base de données

Hypothèses de performance concernant les boîtes englobantes

L'utilisation de boîtes englobantes devrait améliorer les résultats, car elle permet à l'histogramme de ne pas être déformé par les textures environnantes et le bruit que représente le fond de l'image dans la prédiction de catégorie. Par exemple, sur l'image suivante, la boîte englobante permet de ne pas ajouter les textures de la route et de la maison en arrière-plan en se concentrant sur la zone rouge.



Figure 3. Exemple de boîte englobante sur l'image car_query.

Cependant, dans certains cas, il est possible que le contexte dans lequel l'objet est situé puisse contribuer à la classification, surtout dans une petite base de données où les images d'une même catégorie sont similaires comme pour la voiture. En revanche, les images telles que les lotus pourraient voir une augmentation de la performance puisque les fonds varient beaucoup en texture.



Figure 4. Comparaison des textures en arrière-plan des images lotus.

Description des expériences, des données et critères d'évaluation

Expériences réalisées

Afin de tester les hypothèses précédemment posées, les expériences suivantes ont été réalisées :

- Contenu uniforme : Comparaison du score de prédiction des deux méthodes pour les images de test original et les images de test recadrées sur la base de données originale.
- Variation de luminosité : Comparaison du score de prédiction des deux méthodes pour les images originales et les images avec augmentation de la luminosité.
- Boîtes englobantes : Chaque image a été rognée manuellement. Cela permet de se concentrer sur l'objet mis en évidence dans les images sans avoir à programmer les zones d'intérêts pour chaque image. On compare ensuite les performances de prédiction pour les deux méthodes (avec et sans boîtes englobantes).

Difficultés des images requêtes

Les images requêtes de la base de données présentent certaines difficultés :

- Les corps de l'avion et du dauphin sont à la fois sombres et uniformes. La plupart des textures se retrouvent en arrière-plan, et celui-ci diffère des autres images de la base de données.
- L'image de l'avion contient des bordures blanches importantes, ce qui crée une texture similaire à toutes les images ayant un cadre blanc ou un arrière-plan blanc et uniforme comme l'image ball_5.
- L'arrière-plan de l'image requête de lotus est complètement uniforme (tout noir), tandis que certaines autres images de lotus ont des arrière-plans plutôt texturés.
- L'image du pot de cornichon est prise dans un angle assez différent de ceux des autres photos de la base de données, ce qui pourrait rendre difficile de trouver des similitudes.

Mesures de similarité entre descripteurs et métrique d'évaluation

Pour la méthode LBP, deux variables peuvent être modifiées pour modifier le comportement de l'algorithme, soit le rayon du cercle et le nombre de points sur la circonférence du cercle. Une recherche par grille a été effectuée avec des rayons variants de 2 à 8 avec un nombre de points $k * \text{rayon}$ avec k allant de 4 à 10. Ces valeurs varient de 2 afin de limiter le nombre de tests.

Pour la méthode CM, trois variables peuvent être modifiées pour modifier le comportement de l'algorithme, soit la direction, la résolution de l'image et la distance du décalage pour atteindre le pixel dont l'intensité est à comparer. Les valeurs de direction sont testées à 0, 45, 90 et 135°, tandis que les distances varient à 1, 2, 4 et 6 pixels et la précision de la teinte varie par incréments de 4, 8, 12, 16, 20, 24, 28, 32, 40, 48, 56, 64 sur une échelle de couleur allant de 1 à 255.

Les critères de quantification pour les deux méthodes sont la classification par regroupement dans l'hyperespace des vecteurs de descripteurs à l'aide de l'algorithme K-Nearest Neighbour (KNN), ainsi que la classification en fonction du résultat de l'équation de Kulback Leibler (KB) qui quantifie la différence entre deux distributions. Pour l'algorithme KNN, des tests ont été faits avec un nombre de voisins variant de 1 à 5, puisque 5 est la quantité de redondance d'un même type d'image dans la base de données. [1][4] Cependant, les analyses se concentrent sur un score avec un seul voisin, car cette méthode offre une meilleure performance de classification, ce qui est un équivalent au calcul de la distance euclidienne.

La métrique d'évaluation pour chaque méthode (CM et LBP) et pour chaque combinaison d'hyperparamètre est simplement la précision de la classification selon KNN et KB. Ce calcul représente simplement le nombre d'images correctement classifiées sur le nombre d'images à tester, pour obtenir un score de prédiction en pourcentage.

Description des deux implémentations utilisées

Matrices de Co-occurrence (MC)

L'implémentation de la méthode des matrices de co-occurrence s'est fait à l'aide de *scikit-image*, une bibliothèque open source de traitement d'images pour Python. Le code des fonctions des descripteurs est basé sur un exemple trouvé sur GitHub. [2] Les premières étapes consistent à ouvrir une image et de la convertir en grayscale. On définit ensuite des intervalles spécifiques afin de quantifier l'intensité des niveaux de gris. La librairie *scikit-image* possède la fonction *skimage-feature-graycomatrix* permettant de calculer la matrice de co-occurrence de l'image ainsi que la fonction *skimage-feature-graycoprops* pour calculer différents descripteurs de texture qui servent à résumer les caractéristiques de l'image, par exemple le contraste, la dissimilarité et l'homogénéité. [3] Les principaux paramètres de la méthode sont la distance et l'angle entre la paire de pixels comparée, ainsi que le nombre d'intervalles

d'intensité de niveaux de gris pour effectuer (par exemple, sachant que l'intensité de chaque pixel prend une valeur entre 0 et 255, on peut diviser cette plage de valeur en 8, 32, 64, ... intervalles).

Modèles Binaires Locaux (LBP)

L'implémentation de la méthode des modèles binaires locaux est basée sur un exemple de code proposé par *scikit-image*. [1] La fonction *local_binary_pattern* permet d'appliquer l'algorithme sur une image en grayscale. On peut ensuite déterminer l'histogramme des codes binaires résultant et en faire l'analyse des textures. Les paramètres principaux sont le rayon du cercle entourant le pixel étudié et le nombre de points sur le cercle.

Pour ces deux méthodes, les fonctions principales ont été ajoutées à un script sur mesure qui automatise le calcul des résultats pour les données d'entraînement et qui les compare aux images tests. De plus, l'implémentation de la recherche par grille pour les hyperparamètres a également été conçue sur mesure pour cette étude comparative.

Présentation des résultats de tests

La Figure 5 montre un aperçu des précisions obtenues avec la méthode CM en faisant varier la distance avec les pixels comparés (proportionnel au rayon des points) et la taille de l'intervalle pour la discrétisation de l'image (axe horizontal bins_bit). Les différentes couleurs représentent les métriques (KNN ou KB) utilisées et s'il s'agit d'images recadrées ou non.

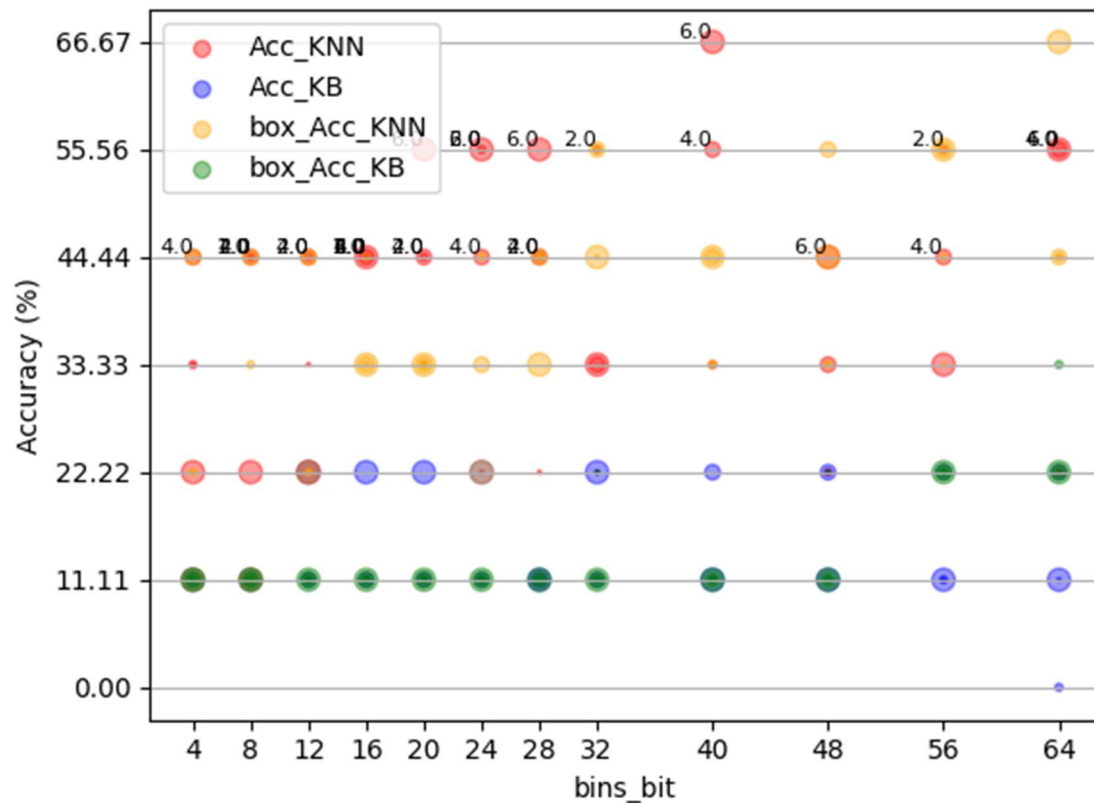


Figure 5: Mesure de la précision de CM selon les métriques KNN (1 voisin) et Kullback en faisant varier la résolution de l'image, ainsi que la distance relative (grosseur des points).

Le Tableau 1 montre les résultats de la prédiction pour la méthode CM sans boîte englobante avec chaque méthode de quantification (KNN et KB). Les hyperparamètres utilisés sont une distance de 6 pixels et une discrétisation par intervalles de 64.

Tableau 1: Résultats de prédiction pour la méthode CM sans boîte englobante

test_image	CM_pred_KNN	CM_pred_KB	pred_KNN	pred_KB	KB_3_best	KNN_3_best
car_query.jpg	dolphin	ball	FALSE	FALSE	['ball', 'dolphin', 'lotus']	['dolphin' 'lotus' 'lotus']
airplane_query.jpg	airplane	ball	TRUE	FALSE	['ball', 'dolphin', 'ball']	['airplane' 'lotus' 'face']
strawberry_query.jpg	dolphin	ball	FALSE	FALSE	['ball', 'dolphin', 'lotus']	['dolphin' 'car' 'car']
dolphin_query.jpg	airplane	ball	FALSE	FALSE	['ball', 'dolphin', 'ball']	['airplane' 'car' 'airplane']
face_query.jpg	face	ball	TRUE	FALSE	['ball', 'lotus', 'ball']	['face' 'face' 'car']
ball_query.jpg	ball	ball	TRUE	TRUE	['ball', 'ball', 'dolphin']	['ball' 'dolphin' 'airplane']
cat_query.jpg	cat	lotus	TRUE	FALSE	['lotus', 'cat', 'ball']	['cat' 'lotus' 'cat']
lotus_query.jpg	lotus	ball	TRUE	FALSE	['ball', 'dolphin', 'lotus']	['lotus' 'dolphin' 'car']
pickles_query.jpg	pickles	dolphin	TRUE	FALSE	['dolphin', 'ball', 'pickles']	['pickles' 'pickles' 'pickles']

La précision de prédiction avec le classificateur KNN est de 66.67% (6/9). La précision de prédiction avec le métrique KB est de 11.11% (1/9).

Le Tableau 2 montre les résultats de la prédiction pour la méthode CM avec boîte englobante avec chaque méthode de quantification (KNN et KB). Les hyperparamètres utilisés sont une distance de 6 pixels et une discrétisation par intervalles de 64.

Tableau 2: Résultats de prédiction pour la méthode CM avec boîte englobante

test_image	CM_pred_KNN	CM_pred_KB	pred_KNN	pred_KB	KB_3_best	KNN_3_best
strawberry_query.jpg	lotus	ball	FALSE	FALSE	['ball', 'dolphin', 'dolphin']	['lotus' 'dolphin' 'pickles']
airplane_query.jpg	airplane	ball	TRUE	FALSE	['ball', 'ball', 'lotus']	['airplane' 'face' 'face']
ball_query.jpg	airplane	ball	FALSE	TRUE	['ball', 'ball', 'lotus']	['airplane' 'lotus' 'ball']
car_query.jpg	dolphin	ball	FALSE	FALSE	['ball', 'dolphin', 'ball']	['dolphin' 'car' 'lotus']
cat_query.jpg	cat	cat	TRUE	TRUE	['cat', 'cat', 'lotus']	['cat' 'cat' 'ball']
dolphin_query.jpg	dolphin	ball	TRUE	FALSE	['ball', 'dolphin', 'ball']	['dolphin' 'lotus' 'ball']
face_query.jpg	face	lotus	TRUE	FALSE	['lotus', 'ball', 'ball']	['face' 'face' 'car']
lotus_query.jpg	lotus	ball	TRUE	FALSE	['ball', 'ball', 'dolphin']	['lotus' 'airplane' 'pickles']
pickles_query.jpg	pickles	ball	TRUE	FALSE	['ball', 'dolphin', 'lotus']	['pickles' 'pickles' 'pickles']

La précision de prédiction avec le classificateur KNN est de 66.67% (6/9). La précision de prédiction avec le métrique KB est de 22.2% (2/9).

La **Error! Reference source not found.** montre un aperçu des précisions obtenues avec la méthode LBP en faisant varier le rayon (proportionnel au rayon des points) et le nombre de points (axe horizontal). Les différentes couleurs représentent les métriques (KNN ou KB) utilisées et s'il s'agit d'images recadrées ou non.

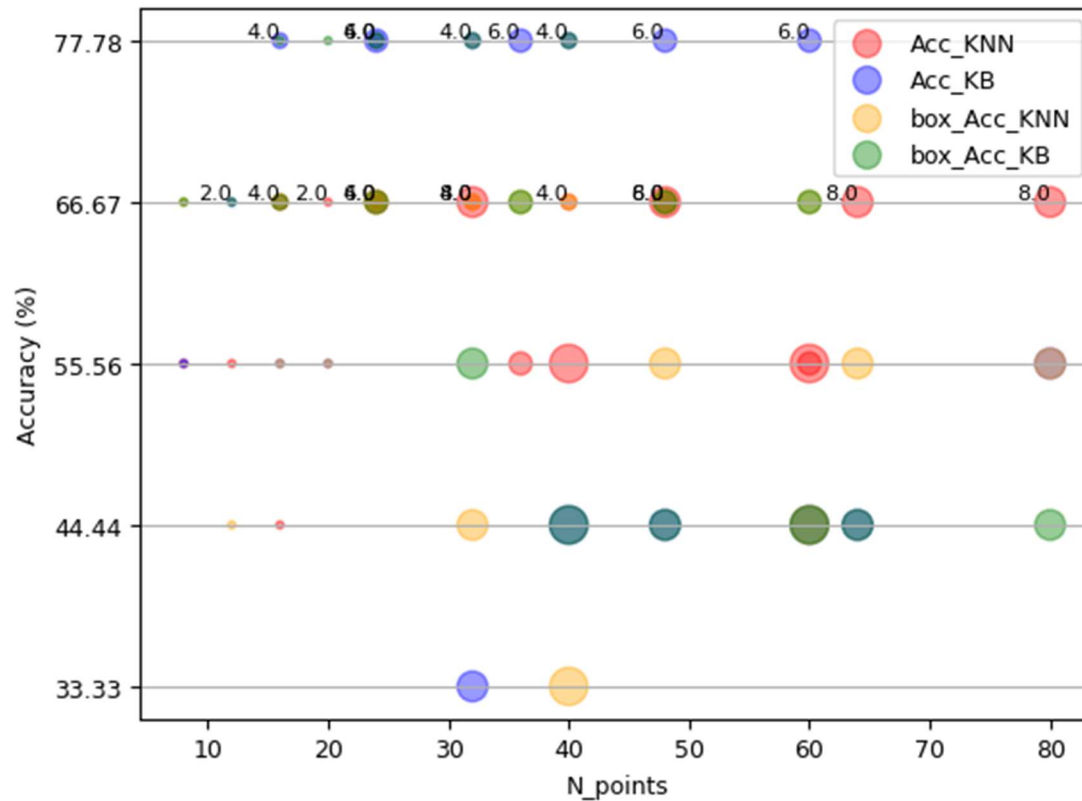


Figure 6: Mesure de la précision de LBP selon les métriques KNN (1 voisin) et Kullback en faisant varier le nombre de points ainsi que le rayon (représenté par la taille des points).

Les meilleurs hyperparamètres pour cette méthode semble être un rayon de 4 et un nombre de points de 16, puisque les 4 méthodes de classification offrent une précision entre 66.67 et 77.78%.

Le Tableau 3 montre les résultats de la prédiction pour la méthode LBP sans boîte englobante avec chaque méthode de quantification (KNN et KB). Les hyperparamètres utilisés sont un rayon de 4 pixels et un nombre de points de 16.

Tableau 3: Résultats de prédiction pour la méthode LBP sans boîte englobante

test_image	LBP_pred_KNN	LBP_pred_KB	pred_KNN	pred_KB	KB_3_best	KNN_3_best
car_query.jpg	car	car	True	True	['car', 'car', 'car']	['car' 'face' 'car']
airplane_query.jpg	airplane	airplane	True	True	['airplane', 'airplane', 'dolphin']	['airplane' 'dolphin' 'lotus']
strawberry_query	airplane	dolphin	False	False	['dolphin', 'dolphin', 'airplane']	['airplane' 'dolphin' 'pickles']
dolphin_query.jpg	car	dolphin	False	True	['dolphin', 'car', 'car']	['car' 'car' 'dolphin']
face_query.jpg	face	face	True	True	['face', 'face', 'face']	['face' 'car' 'face']
ball_query.jpg	lotus	cat	False	False	['cat', 'lotus', 'dolphin']	['lotus' 'cat' 'lotus']
cat_query.jpg	cat	cat	True	True	['cat', 'cat', 'ball']	['cat' 'cat' 'ball']
lotus_query.jpg	lotus	lotus	True	True	['lotus', 'ball', 'ball']	['lotus' 'pickles' 'pickles']
pickles_query.jpg	pickles	pickles	True	True	['pickles', 'pickles', 'pickles']	['pickles' 'pickles' 'lotus']

La précision de prédiction avec le classificateur KNN est de 66.67% (6/9). La précision de prédiction avec le métrique KB est de 77.8 % (7/9).

Le Tableau 4 montre les résultats de la prédiction pour la méthode LBP avec boîte englobante avec chaque méthode de quantification (KNN et KB). Les hyperparamètres utilisés sont un rayon de 4 pixels et un nombre de points de 16.

Tableau 4: Résultats de prédiction pour la méthode LBP avec boîte englobante

test_image	pred_KNN	pred_KB	pred_KNN	pred_KB	KB_3_best	KNN_3_best
strawberry_query	pickles	pickles	False	False	['pickles', 'pickles', 'pickles']	['pickles' 'pickles' 'pickles']
airplane_query.jpg	airplane	airplane	True	True	['airplane', 'airplane', 'car']	['airplane' 'car' 'face']
ball_query.jpg	ball	ball	True	True	['ball', 'airplane', 'dolphin']	['ball' 'airplane' 'cat']
car_query.jpg	car	car	True	True	['car', 'lotus', 'car']	['car' 'lotus' 'lotus']
cat_query.jpg	cat	cat	True	True	['cat', 'cat', 'car']	['cat' 'cat' 'ball']
dolphin_query.jpg	cat	cat	False	False	['cat', 'dolphin', 'face']	['cat' 'dolphin' 'face']
face_query.jpg	face	face	True	True	['face', 'face', 'face']	['face' 'face' 'face']
lotus_query.jpg	pickles	lotus	False	True	['lotus', 'ball', 'pickles']	['pickles' 'ball' 'ball']
pickles_query.jpg	pickles	pickles	True	True	['pickles', 'pickles', 'pickles']	['pickles' 'pickles' 'pickles']

La précision de prédiction avec le classificateur KNN est de 66.67% (6/9). La précision de prédiction avec le métrique KB est de 77.8 % (7/9).

Le Tableau 5: Temps d'exécution pour chaque méthode Tableau 5 présente un résumé du temps de calcul des meilleurs hyperparamètres pour chacune des méthodes. Puisque plusieurs combinaisons d'hyperparamètre offrent une performance similaire, ce tableau servira à analyser le temps de calcul pour choisir une paire de paramètres pour chaque méthode.

Tableau 5: Temps d'exécution pour chaque méthode

Méthode	CM			LBP		
Paramètre	Nombre de points par intervalle			Nombre de points		
	16	32	64	16	32	64
Temps (s)	1.2	0.8	0.8	6.2	9.4	16.5

Le Tableau 6 permet d'analyser l'impact de la modification de l'uniformité des images tests sur la classification avec la base de données originale.

Tableau 6: Évaluation de l'impact de l'uniformité sur la classification.

Images tests	Précision (%) LBP		Précision (%) CM	
	KNN	KB	KNN	KB
Originales	66.67	77.75	66.67	11.11
Images uniformisées	33.34	33.34	33.34	11.11

Le Tableau 7 et Tableau 8 présentent une comparaison des images de test originales avec les mêmes images ayant été modifiées par l'augmentation de leur luminosité.

Tableau 77: Comparaison de l'impact de la luminosité sur les performances de la méthode LBP

Images tests	Précision sans boîte (%)		Précision avec boîte (%)	
	KNN	KB	KNN	KB
Originales	66.67	77.75	66.67	66.67
Luminosité augmentée	55.56	66.67	44.44	44.44

Tableau 88: Comparaison de l'impact de la luminosité sur les performances de la méthode CM

Images tests	Précision sans boîte (%)		Précision avec boîte (%)	
	KNN	KB	KNN	KB
Originales	66.67	11.11	66.67	22.22
Luminosité augmentée	55.56	22.22	33.33	11.11

Discussion des résultats et retour sur les hypothèses

Méthode CM

La Figure 5 permet de visualiser l'effet de la distance entre les pixels, et de la résolution de l'image sur le score de prédiction de la méthode CM. On voit que c'est une distance de 6 pixels qui a permis d'obtenir les deux meilleurs scores (66.7%). Un de ces points avait une discrétisation par intervalles de 40, alors que l'autre était de 64. Pour une précision équivalente, une discrétisation par intervalles plus grands est à privilégier, car un nombre réduit d'intervalles diminue le temps de calcul, comme le montre le Tableau 5. C'est pourquoi la combinaison

d'hyperparamètres utilisée pour obtenir les résultats des Tableaux 1 et 2 sont une distance de 6 pixels et une discrétisation de l'image par intervalles de 64.

D'ailleurs il a été observé que la modification de l'orientation ou les deux pixels comparés à des valeurs de 45, 90 et 135° réduisait les performances ou les gardait équivalentes dans certains cas. Une orientation de 0° a donc été retenue.

Méthode LBP

La Figure 6 permet de visualiser l'effet du rayon du cercle et du nombre de points sur le score de prédiction de la méthode LBP. On voit que ce sont les rayons de 4 et 6 pixels qui ont permis d'obtenir les meilleurs scores (77.8%). Il semble que la méthode LBP fonctionne mieux pour des plus petits rayons, car ceux-ci permettent de mieux capter les textures locales. Les meilleurs scores obtenus avaient des nombres de points par cercle allant de 16 à 60. Pour une précision équivalente, un nombre de points plus faible est à privilégier, car cela diminue le temps de calcul, comme le montre le Tableau 5. C'est pourquoi la combinaison d'hyperparamètres utilisée pour obtenir les résultats des Tableaux 3 et 4 est un rayon de 4 pixels et un nombre de points de 16.

Comparaison des méthodes de quantification (KNN vs KB)

Il est intéressant de comparer quelle est la meilleure méthode de quantification entre K-Nearest Neighbour (KNN) et Kullback Leibler (KB). Pour la méthode CM, on voit clairement sur la Figure 5 que les points bleus et verts correspondant à KB ont tendance à se situer vers le bas, à des niveaux de précision faibles, tandis que les points rouges et jaunes correspondant à KNN (k=1) se situent plutôt vers le haut, à des niveaux de précision plus élevés. On en conclut que KNN permet généralement d'obtenir un meilleur score de prédiction pour la méthode CM. Malgré que les points sur la Figure 6 sont plus éparpillés, on voit tout de même une tendance inverse : les points bleus et verts (KB) sont plutôt vers le haut et les points jaunes et rouges (KNN) sont plutôt vers le bas. On en conclut que KB permet généralement d'obtenir un meilleur score de prédiction pour la méthode LBP. L'intérêt d'utiliser deux critères d'évaluation est donc justifié, puisque chacun permet d'obtenir de meilleurs résultats soit pour CM ou pour LBP.

Comparaison entre CM et LBP

Le meilleur score de prédiction de la méthode CM (en utilisant la méthode de quantification KNN) est de 66.7%, tandis que le meilleur score de prédiction de la méthode LBP (en utilisant la méthode de quantification KB) est de 77.8%. Sachant que le meilleur score possible est de 88.9% (8/9) étant donné qu'il n'y a pas d'images de fraise dans la banque de données, les scores obtenus pour chaque méthode démontrent un bon niveau de précision. Cependant, affirmer que la méthode LBP est plus précise que la méthode CM sur la base de ces statistiques manquerait de vigueur, étant donnée la faible taille de l'échantillon (uniquement 9 images requêtes).

Pour le temps de calcul, on observe sur le Tableau 5 que la méthode CM s'exécute en environ 1 seconde (entre 0.8 et 1.2 sec. selon le nombre de points par intervalle), tandis que la méthode LBP s'exécute en environ 10 secondes (entre 6.2 et 16.5 secondes selon le nombre de points sur le cercle). Ce facteur 10 entre le temps d'exécution de chaque méthode confère un avantage significatif à la méthode CM pour des applications à grande échelle.

Cas régions uniformes

Pour le cas spécifique des régions uniformes, on peut réduire la variation dans les images en prenant comme images de tests les images rognées générées pour les tests avec boîte englobante. En procédant ainsi, il est possible d'observer si l'uniformisation de l'image test lui permettra d'être mieux classifiée. Le Tableau 6 montre que la précision de prédiction pour la méthode LBP passe de 66.7% à 33.34% pour KNN et de 77.8% à 33.3% pour KB lorsqu'on uniformise l'image. Pour la méthode CM, le score passe de 66.7% à 33.3% pour KNN et reste inchangé pour KB à 11.1%. Ces résultats sont non concluants étant donné qu'en général la prédiction devient moins performante quand l'image est uniformisée, et ce autant pour les deux méthodes.

Cas variation de luminosité

Pour tester l'impact de la variation de luminosité sur la performance des deux méthodes, les images de test sont modifiées avec une augmentation de la luminosité de 19.6% (50/255). Les deux figures montrent l'impact de l'augmentation de la luminosité sur les images de test.



Figure 7 : Images de test originales.



Figure 88 : Images de test avec luminosité augmentée de 19.6%

Il est possible de remarquer que cette augmentation cause une saturation sur certaines images. Par exemple, sur le ballon, il n'y a pas beaucoup de changements, car les couleurs sont déjà presque saturées. Aussi, cette augmentation de luminosité augmente la variation globale de luminosité dans l'image, puisque les zones pâles deviennent plus blanches et les zones sombres restent sombres. Les Tableaux 7 et 8 montrent que, autant pour la méthode CM que LBP, avec ou sans boîte englobante, le score de prédiction est généralement plus faible lorsque la luminosité est augmentée. Cependant, les résultats obtenus sont insuffisants pour conclure de façon significative sur la question de quelle méthode est la mieux adaptée pour les images avec des variations de luminosité importante.

Avec ou sans boîte englobante

Le seul cas où l'utilisation de boîte englobante a eu un impact est pour la méthode CM avec le métrique KB, alors que le score de prédiction est passé de 11.1% à 22.2% (cette différence est d'ailleurs bénigne). Pour toutes les autres combinaisons de méthode de métrique, la précision a été la même avec et sans boîte englobante. On ne peut donc pas conclure sur l'intérêt de les utiliser. L'hypothèse initiale qui supposait que les boîtes englobantes devraient améliorer la précision des résultats ne peut donc pas être confirmée. On suppose tout de même que les arrière-plans des images peuvent fournir des textures qui aident à la classification.

Améliorations

Certains tests pourraient être améliorés pour obtenir de meilleurs résultats. D'abord, le nombre d'images dans la base de données devrait être augmenté. Cela permettrait d'avoir un échantillon plus représentatif et normalisé, ce qui permettrait probablement d'obtenir une meilleure précision et une meilleure confiance sur l'impact de la luminosité et de l'uniformité. De plus, des boîtes englobantes permettent de focaliser les algorithmes davantage sur la région d'intérêts, mais il reste que l'arrière-plan reste visible et possède des textures qui affectent les résultats. Ainsi, avoir un algorithme qui repère l'objet principal de l'image et supprime l'entièreté de l'arrière-plan (en le mettant tout en blanc ou tout en noir) pourrait mener à une meilleure précision de prédiction. Afin d'obtenir des résultats plus concluants sur l'impact de la variation de la luminosité, il faudrait avoir des photos similaires prises avec différents niveaux d'éclairage pour changer les gradients dans l'image et non pas juste augmenter uniformément l'intensité sur chaque pixel.

Bibliographie

- [1] « Local Binary Pattern for texture classification — skimage 0.22.0 documentation ». Consulté le: 5 février 2024. [En ligne]. Disponible à: https://scikit-image.org/docs/stable/auto_examples/features_detection/plot_local_binary_pattern.html
- [2] 水無月 MINAZUKI, « renbooo/Gray-Level-Cooccurrence-Matrix ». 26 octobre 2023. Consulté le: 5 février 2024. [En ligne]. Disponible à: <https://github.com/renbooo/Gray-Level-Cooccurrence-Matrix>
- [3] « skimage.feature — skimage 0.23.0rc0.dev0 documentation ». Consulté le: 5 février 2024. [En ligne]. Disponible à: <https://scikit-image.org/docs/dev/api/skimage.feature.html#skimage.feature.graycoprops>
- [4] « sklearn.neighbors.KNeighborsClassifier ». Consulté le 9 février, 2024. [En ligne]. Disponible à: <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>