
Rapport TP1

**POLYTECHNIQUE
MONTRÉAL**

UNIVERSITÉ
D'INGÉNIERIE



Description et comparaison de régions d'intérêts

INF6804 – Vision par ordinateur

Travail présenté à Guillaume-Alexandre Bilodeau

Hiver 2023

Département de génie informatique et génie logiciel
École Polytechnique de Montréal

Dernière mise à jour : 5 février 2023 à 21:31:52

Guillaume Girouard
Pierre-Emmanuel Rebours

1954899
2165286

Table des matières

| | |
|--|----|
| Table des matières..... | 1 |
| 1. Présentation des 2 méthodes | 2 |
| 1.1 Histogramme des gradients orientés (HOG) | 2 |
| 1.2 Scale-invariant feature transform (SIFT) | 3 |
| 2. Hypothèses de performance..... | 4 |
| 2.1 Sans boîte englobante..... | 4 |
| 2.2 Avec boîte englobante | 4 |
| 3. Description des expériences | 4 |
| 3.1 Protocole | 4 |
| 4. Implémentation des méthodes..... | 6 |
| 4.1 Méthode HOG | 6 |
| 4.2 Méthode SIFT | 7 |
| 4. Présentation des résultats | 8 |
| 5.1 Hypothèse 1 | 8 |
| 5.2 Hypothèse 2 | 9 |
| 5.3 Hypothèse 3 | 10 |
| 5.4 Aparté sur le temps d'exécution des méthodes | 12 |
| 6. Discussion des résultats:..... | 13 |
| 6.1 Discussion Hypothèse 1 | 13 |
| 6.2 Discussion Hypothèse 2 | 14 |
| 6.3 Discussion Hypothèse 3 | 14 |
| 6.4 Discussion sur le temps d'exécution des méthodes. | 15 |
| 6.5 Amélioration générale des tests exercés | 15 |
| Annexe A – Graphe d'optimisation des hyperparamètres | 16 |
| Annexe B – Graphe du temps d'exécution par rapport au nombre d'octave considéré pour SIFT :..... | 19 |
| Annexe C – Bibliographie..... | 20 |

1. Présentation des 2 méthodes

1.1 Histogramme des gradients orientés (HOG)

La méthode HOG est une méthode de vision par ordinateur qui fournit un descripteur pour caractériser une image en se basant sur le calcul du gradient.

Pour calculer le gradient d'une image, on applique un filtre tel que le filtre de Sobel par convolution avec l'image. Dans le cas de Sobel, deux noyaux de convolution sont appliqués pour réagir aux fortes variations d'intensité de pixel en x et en y. Le résultat de cette opération nous permet d'obtenir les contours des objets à l'intérieur de l'image. Les deux filtres de Sobel sont les suivants :

$$Sobel_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}; Sobel_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Dans le cadre d'une image RGB, le gradient est calculé sur chacune des trois composantes: on ne garde pour chaque pixel que celui qui a la valeur la plus grande.

En réalisant le calcul de gradient sur notre image, on peut en déduire l'amplitude du gradient ainsi que son orientation donnée par les formules suivantes:

$$F_G = \sqrt{G_x^2 + G_y^2}$$
$$\theta_G = 2 \cdot \arctan\left(\frac{G_y}{F_G \cdot G_x}\right)$$

avec:

- F_G , l'amplitude du gradient G
- G , l'orientation du gradient G
- G_x , le gradient selon l'axe horizontal
- G_y , le gradient selon l'axe vertical

À partir de l'amplitude et de l'orientation du gradient, on va pouvoir déterminer notre descripteur. Pour cela, on divise notre image en cellule de mêmes tailles (quelques pixels). Ces cellules font-elles même partie de blocs se superposant sur l'image: la taille des blocs est de l'ordre de quelques cellules. Pour chaque cellule, on détermine l'histogramme des gradients orientés de cette cellule: pour se faire, après avoir défini au préalable plusieurs intervalles d'orientation pour nos histogrammes, on compte, pour chaque intervalle, le nombre d'orientations des gradients de la cellule qui lui appartiennent. L'amplitude de chaque gradient sert de poids lors du comptage de l'orientation: plus ce dernier est important, plus l'orientation associée donnera une valeur plus grande au compteur de l'intervalle qui lui est associé. Une fois les histogrammes calculés pour chaque cellule vient le rôle des blocs superposés: pour chaque bloc, on normalise les histogrammes de chacune des cellules incluses par rapport au bloc et on concatène les valeurs de nos histogrammes (ce qui nous donne, par bloc, un vecteur dont la taille vaut le nombre de cellules incluses dans le bloc multiplié par le nombre d'intervalles choisi pour nos histogrammes): cette normalisation permet notamment de limiter l'impact du contraste sur la description de l'image.

Notre descripteur est obtenu en concaténant les vecteurs trouvés pour chaque bloc: on a donc un unique vecteur dont la taille est égale au nombre de blocs superposés multiplié par le nombre de cellules incluses dans les blocs multiplié par le nombre d'intervalles choisi pour nos histogrammes.

Il y a plusieurs hyperparamètres principaux pour cette méthode:

Tout d'abord, il y a les différents intervalles à choisir pour construire nos histogrammes ainsi que leur nombre. Mais il y a aussi la taille des cellules ainsi que celle des blocs superposés qu'il convient de choisir. Dans une moindre mesure, il y a également le pas pour le placement des blocs superposés (ex: de cellule en cellule, de 2 cellules à l'horizontale et d'une cellule à la verticale...)

1.2 Scale-invariant feature transform (SIFT)

L'approche SIFT est une méthode pour trouver une description du contenu d'une image par l'intermédiaire points clés. Pour la réalisation de cette méthode, 4 étapes sont présentées pour se retrouver avec le descripteur de l'image.

Premièrement, un filtre passe-bas gaussien est appliqué à différentes puissances d'un coefficient d'échelle k et sur différents taux de résolution (octaves) de l'image. Par la suite, une différence de gaussienne (DoG) est obtenues entre chaque voisin de la même octave. Le calcul des DoG se calcule alors comme suit:

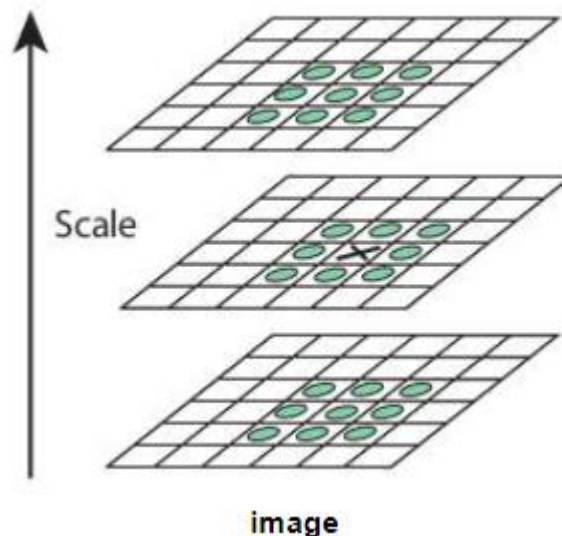
$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$D(x, y, (k_i - k_j)) = (G(x, y, k_i) - G(x, y, k_j)) \cdot I(x, y)$$

avec:

- G , la fonction de filtre passe-bas gaussien
- σ , la fréquence spatiale de coupure du filtre
- D , la différence de gaussienne
- k_i, k_j , les coefficients de changement d'échelle
- I , l'image

Par l'intermédiaire de la première étape, il est possible d'identifier un premier lot de point clé associé aux extrémums locaux détectés dans un voisinage de pixel. Ce voisinage comprend non seulement les pixels entourant le point sur la même image, mais aussi sur celles des facteurs d'échelle comme montré sur l'image suivante:



Source: https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html

La deuxième étape sert à filtrer le premier lot de point clé. Les points situés dans l'arrière-plan ou aux contours de l'image sont peu intéressants à conserver. Ainsi, on se débarrasse des points situés dans

des zones possédant de faibles contrastes et où le gradient est fortement perpendiculaire et faiblement parallèle aux frontières.

Par la suite, on assigne une orientation et une magnitude du gradient de chaque point clé et son alentour sous forme d'histogramme. Le pique le plus élevé représentera l'orientation globale du point clé.

La dernière étape sert à la construction du descripteur de chaque point clé. Chaque descripteur est sur 128 valeurs où est construit un histogramme de l'orientation du gradient autour des points clés.

L'histogramme est sur 8 valeurs et correspond à 16 régions autour du point clé.

2. Hypothèses de performance

2.1 Sans boîte englobante

De la compréhension faite des deux approches, il serait à penser que SIFT risque de donner de meilleures performances de détection pour des objets ayant subi une rotation sur le plan de l'image ou hors plan. La technique SIFT se veut robuste par sa complexité d'implémentation. Effectivement on assigne une orientation sur chaque point clé de l'image (étape 3). Cette opération rend le descripteur des points clés invariants à la rotation de l'objet dans le plan et permet une certaine tolérance des rotations hors plan image.

Par la suite, une hypothèse possible à émettre serait que HOG risque d'être meilleur que SIFT pour identifier des objets de forme caractéristique, mais d'allure ou de modèle légèrement différent. Par exemple, deux voitures de modèle différent. En effet, SIFT est souvent plus utilisé dans un contexte de devoir retrouver exactement le même objet dans des scènes au lieu de devoir classifier un type d'objet par rapport à une base de données. Ainsi, parce que HOG porte une attention plus particulière aux contours de l'image propre par calcul de gradient, il serait à penser que cette méthode serait plus apte à classifier des objets de ce contexte.

2.2 Avec boîte englobante

Supposons à présent que l'on ait accès aux boîtes englobantes, indiquant ainsi la position de l'objet d'intérêt.

La méthode dont les performances se verront davantage améliorées est la méthode HOG: HOG est sensible, contrairement à SIFT, à la position de l'objet sur l'image et à son échelle, dans la mesure où il n'y a pas de référentiel permettant de situer les descripteurs de chaque bloc sur les éléments qui composent l'image. La boîte englobante assure que la méthode soit appliquée à l'objet d'intérêt avec un arrière-plan et un contexte d'image réduit: on sait où se trouve l'objet, on a simplement à calculer nos descripteurs sur la zone définie par la boîte englobante. Cela facilitera grandement la comparaison entre nos descripteurs: on saura que ces derniers seront bien ceux associés à l'objet d'intérêt.

SIFT resterait compétitif, mais ces gains en termes de performance seraient moindres: contrairement à HOG, SIFT repose sur l'usage de point clé qui permette de faire des correspondances entre des images à différentes échelles et où l'objet se trouve à des positions différentes.

3. Description des expériences

3.1 Protocole

Le travail à réaliser est de réaliser des recherches d'image par le contenu (CBIR). Un nombre d'images requêtes est donné, avec, pour chacune d'elle, 5 images correspondantes dans la base de données à

l'exception de l'image requête strawberry_query.jpg. L'objectif est alors de faire ressortir les 3 images les plus correspondantes à l'image requête par comparaison des descripteurs entre l'image requête et chaque image de la base de données. Cette mesure de similarité entre descripteurs d'une image à l'autre est la distance entre descripteurs pour HOG et le calcul de la moyenne des distances de chaque point correspondant pour SIFT. Pour les deux méthodes, la distance euclidienne est celle privilégiée.

Par rapport à la validation des hypothèses présentée précédemment, un protocole expérimental est établi comme suit: premièrement, une liste des difficultés présentes pour chacune des catégories d'image dans la base de données est établie. Par l'identification de ces difficultés, il nous sera possible d'attribuer les lots d'image d'intérêt à la validation des deux premières hypothèses formulées. Le tableau suivant donne une description des caractéristiques de chaque image :

TABLE 3.1-1 TABLEAU DE CLASSIFICATION DES ÉCHANTILLONS D'INTÉRÊT PAR HYPOTHÈSE

| Catégorie d'image | Description de la difficulté d'identification de l'image | Attribution en échantillon aux hypothèses |
|-------------------|--|---|
| Airplane | Objet de couleurs différentes, forme différente | Hypothèse 2 et 3 |
| Ball | Translation de l'objet sur l'image, occlusion de l'objet par le gazon, grosseur de l'objet, rotation en plan de l'image, rotation hors plan de l'image | Hypothèse 1 et 3 |
| Car | Contours différents, couleurs différentes | Hypothèse 2 et 3 |
| Cat | Éclairage changeant, occlusion de l'image | Hypothèse 3 |
| Dolphin | Rotation hors plan, rotation dans le plan, éclairage changeant | Hypothèse 1 et 3 |
| Face | Expression faciale changeante | Hypothèse 2 et 3 |
| lotus | Rotation en plan, rotation hors plan, grosseur changeante, translations | Hypothèse 1 et 3 |
| pickles | Rotation hors plan, rotation en plan, grosseur changeante, détails de l'arrière-plan. | Hypothèse 1, 2 et 3 |
| strawberry | Aucune représentation dans la base de données | Hypothèse 3 |

Pour la troisième hypothèse, une base de données alternative contenant les images coupées manuellement aux boîtes englobantes sur l'objet d'intérêt est créée. Pour cette hypothèse, chaque image requête est considérée dans l'échantillon de l'expérience.

Une fois un lot d'échantillons d'intérêt identifié, des métriques sont établies pour mesurer la performance face aux hypothèses.

En considérant la topologie de test de performance de la métrique attribuée aux hypothèses 1 et 2, on procède par l'identification des trois images de la base de données les plus correspondantes à l'image requête. Par la suite, on considère une incrémentation de score positif pour chaque image ressortie qui a la même classe que l'objet requête. Le taux de performance ressortie est alors le ratio entre la somme de tous les positifs trouver pour chaque image requête faisant partie de l'échantillonnage sur le score maximal possible, et donc :

$$\text{taux de positivité} = \frac{\text{nombre d'images bien identifiés}}{3 \cdot (\text{nombre d'image requête de l'échantillon})}$$

Par la suite, pour l'hypothèse 3, on considère en mesure de performance chaque image requête mise à notre disposition. Parce que l'image de la fraise n'a pas de représentation dans la base de données, on rajoute une logique de rejet sur l'algorithme de classification. De ce fait, on représente alors comme score positif les images rejetées quand la requête de la fraise est demandée. Cette logique de rejet est implémentée par l'intermédiaire d'un seuil considéré comme un hyperparamètre du problème. Pour une mesure de distance entre descripteur trop grande par rapport au seuil, une étiquette Undefined sera apposée dans le classement au lieu de la représentation visuelle de l'identification. Pour considérer

l'image requête de la fraise, le score positive s'incrémente par rapport au nombre d'étiquettes Undefined présentes dans le classement. Le taux trouvé sera alors :

$$\text{taux de positivité} = \frac{(\text{nombre d'images bien identifiés}) + (\text{nombre de undefined pour l'image fraise})}{3 \cdot (\text{nombre d'image requête de l'échantillon})}$$

À titre d'exemple, pour un échantillon ne comprenant que l'image requête du chat, si les trois images les plus correspondantes sont respectivement : pickles_2.jpg, cat_1.jpg et cat_4.jpg, le taux de performance sera de $2/3 = 0.667$ pour cette expérience.

À titre d'exemple supplémentaire, pour un échantillon ne comprenant que l'image requête de la fraise, si on trouve le top 3 correspondant de : ball_2.jpg, Undefined, Undefined, le score de la fraise sera de $2/3 = 0.667$ pour cette expérience.

Par cette métrique, il est possible d'exercer une optimisation des méthodes sur la sélection des hyperparamètres pour adapter les méthodes de descripteurs aux besoins de l'expérience. Le choix des hyperparamètres est alors fixé de la manière suivante :

- On commence en mettant la valeur de chaque hyperparamètre par ceux prédéfinis par défaut dans les fonctions prises.
- On fait varier un premier hyperparamètre de la méthode sur une échelle adaptée.
- On trouve la valeur donnant le taux de performance le plus élevé par rapport à la métrique d'évaluation définit plus haut en considérant toutes les images requêtes. Si plusieurs valeurs donnent un résultat optimal, on regarde pour chacune de ces valeurs, les performances obtenues en faisant varier l'hyperparamètre suivant dans notre processus itératif.
- On fixe l'hyperparamètre optimal trouvé comme nouvelle valeur et on applique la même stratégie au prochain jusqu'à déterminer tous les hyperparamètres optimaux.

L'optimisation de l'hyperparamètre du seuil de rejet dans le classement nous apporte un seuil de distance qui converge en performance face à de grandes valeurs de seuil ([voir annexe](#)). Il est à remarquer que le nombre d'images servant à tester les performances de rejet n'est pas assez conséquent. Ainsi, pour conserver de bonnes performances d'identification, il est préférable d'enlever la logique de seuil lors de l'identification pour l'hypothèse 3.

Pour voir les graphes de performances, veuillez vous référer [à l'annexe](#).

Pour plus de détail sur le déroulement du processus d'optimisation, veuillez vous référer au code.

4. Implémentation des méthodes

4.1 Méthode HOG

L'implémentation de la méthode HOG repose sur la bibliothèque open CV: les outils étant déjà présents, avec une flexibilité satisfaisante, il a été décidé de directement utiliser l'implémentation offerte par openCV. Il aurait été possible d'utiliser l'implémentation de la méthode faite dans la bibliothèque skimage. Cependant, bien que l'usage de ces méthodes soit plus simple et qu'elle propose une visualisation de nos descripteurs directement sur l'image (ce qui n'est pas possible avec openCV), cette dernière laissait moins de possibilités en termes de changement d'hyperparamètre. De plus, la façon dont elle fonctionnait restait plus opaque.

Une première chose à faire avant de déterminer les descripteurs HOG et de s'assurer que l'on va pouvoir les comparer: notre vecteur après avoir appliqué la méthode HOG sur une image doit conserver sa dimension. L'implémentation de la méthode HOG utilisée ne permet pas de définir le nombre exact de

cellules et de blocs par image (donc indépendamment de la dimension de cette dernière). Pour s'assurer que les vecteurs aient la même dimension, il a été décidé de changer la résolution des images pour qu'elles aient toutes la même taille et ainsi leurs vecteurs, correspondant aux descripteurs HOG, auront la même dimension. Comme la méthode ne sera pas appliquée pour détecter uniquement un objet spécifique, mais bien des objets de nature et de forme différente, choisir une taille arbitraire n'est pas forcément évident. À défaut de mieux, nous avons décidé de calculer la moyenne des résolutions des images selon l'axe vertical et horizontal.

On commence par créer l'objet "HOGDescriptor" qui est défini par les paramètres que l'on souhaite donner à la méthode:

```
hog=cv2.HOGDescriptor(winSize,blockSize,blockStride,cellSize,nbins)
```

Les principaux sont la taille de notre fenêtre de détection (winSize), des cellules (cellSize), des blocs (blockSize), le nombre d'intervalles pour la constitution de nos histogrammes (nbins) et le pas entre chaque bloc superposant (blockStride): tous ces paramètres sont donnés en pixels sauf bien entendu nbins. Pour le reste des paramètres, on a gardé leur valeur par défaut. À noter que la normalisation par bloc se fait suivant une normalisation euclidienne (L2-Norm): ce paramètre peut être changé l'or de l'initialisation de l'objet HOGDescriptor, mais nous avons préféré le garder ainsi: Les résultats d'une étude de l'article [1] (cf. bibliographie à la fin du rapport) à la page 5 figure4.c, montre que la normalisation par bloc se fait bien avec une normalisation euclidienne et que changer de norme n'améliore pas drastiquement les résultats.

Pour le reste des hyperparamètres, nous avons cherché empiriquement des valeurs qui se rapprochent d'un optimal ([cf. graphe de performance en annexe](#))

TABLE 4.1-1 CHOIX D'HYPERPARAMÈTRES POUR LA MÉTHODE HOG

| paramètre | Description | Valeur attribuée |
|--------------------|--|------------------|
| blockSize | Taille des blocs superposés (en pixels) | 77x77 |
| blockStride | Pas entre chaque bloc superposé (en pixels) | 11x11 |
| Cellsize | Taille d'une cellule (en pixels) | 11x11 |
| nbins | Nombre d'intervalles pour construire l'histogramme | 9 |

Une fois notre objet de type HOGDescriptor initialisé, il est possible d'appliquer la méthode à l'image à l'aide de la méthode .compute() de l'objet:

```
hog.compute(img)
```

Où img est l'image en niveau de gris dont on veut tirer les descripteurs.

Cette méthode renvoie donc l'histogramme des gradients orientés de chacun des blocs en les concaténant (donc on a un vecteur de dimension 1 avec *nombre de bloc × nombre de cellule par bloc × nombre d'intervalle (ie nbins)* valeurs)

4.2 Méthode SIFT

Pour implémenter la méthode SIFT, la librairie de traitement d'image d'openCV est utilisée. Cette librairie donne accès à une multitude d'hyperparamètres pour permettre de calibrer le filtrage des points clés et augmenter ainsi la performance de détection. De ces hyperparamètres, les valeurs suivantes ont été déterminées par optimisation du taux de performance décrit dans le protocole

TABLE 4.2-1 CHOIX D'HYPERPARAMÈTRES POUR LA MÉTHODE SIFT

| paramètre | Description | Valeur attribuée |
|--------------------------|---|------------------|
| nOctaveLayers | Nombre de niveaux d'échelle de résolution de l'image utiliser dans les calculs de DoG | 1 |
| contrastThreshold | Valeur seuil pour filtrer les points dans les surfaces uniformes | 0.05 |
| edgeThreshold | valeur seuil pour filtrer les points dans les contours de l'image | 10 |
| sigma | coefficient de coupure de fréquence spatiale des filtres gaussiens dans le calcul des DoG | 1.6 |

Il est à remarquer que l'optimisation détermine qu'une seule octave est nécessaire pour obtenir la meilleure performance de détection. Ce résultat s'explique par le fait que la grandeur de l'objet varie très peu sur la majorité des images de la base de données. Ainsi, il est préférable de garder des points clés plus stables que d'obtenir l'invariance de la grosseur de l'objet sur l'image.

La procédure de détermination des résultats prend alors la tournure suivante. Il aurait été souhaitable de fixer la résolution des images requêtes et des images de la base de données par rapport à la moyenne de résolution de ces dernières comme dans HOG. Cependant, il est observé que d'utiliser la valeur fixe de résolution de 256x256 (valeur trouvée de façon empirique) entraîne une hausse de performance face au taux de performance de la méthode. Cette manipulation nous permet d'enlever la résolution de l'image comme facteur d'influence du calcul. Par exemple, l'image `pickels_query.jpg` possédant une résolution plus élevée générerait une plus grande quantité de points clés des autres images venant ainsi biaiser les résultats. Par la suite, on trouve les descripteurs des deux images en appliquant l'algorithme SIFT avec les hyperparamètres définis plus haut. C'est alors que la classe `BFMatcher` d'opencv est utile. Il est possible de créer un objet `BFMatcher` avec comme paramètre `cv2.NORM_L2` pour calculer la distance d'après la norme euclidienne et `crossCheck=True` pour n'avoir en retour que les points ayant matché entre les deux images par rapport à un certain seuil de distance fixé par OpenCV. Finalement, on détermine un score de reconnaissance d'image par rapport à la moyenne de distance euclidienne des points ayant matché.




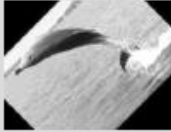

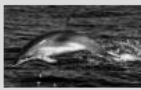










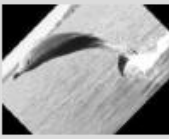


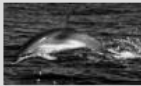


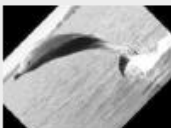





4 Présentation des résultats

Pour cette section, les tableaux des résultats seront montrés par hypothèse. Le classement des trois images les plus correspondantes de la base de données face aux échantillons requête sera présenté pour les méthodes SIFT et HOG.

5.1 Hypothèse 1

La première hypothèse était que SIFT serait plus performante que HOG face à une image ayant subi des rotations planaire et non-planaire à l'image. Pour considérer cette hypothèse, les images requêtes considérées pour montrer la robustesse face à des rotations hors plan et en plan de l'image sont le chat, le lotus, le dauphin et le pot de cornichon.

TABLE 5.1-1 TABLEAU DES RÉSULTATS DE L'HYPOTHÈSE 1






















| Image requête | Classement Top 3 pour SIFT | | | Classement Top 3 pour HOG | | |
|---|---|---|---|--|--|--|
|  | 1 Distance: 229.33  | 2 Distance: 245.59  | 3 Distance: 249.04  | 1 Distance: 48.09  | 2 Distance: 48.31  | 3 Distance: 48.34  |
|  | 1 Distance: 244.53  | 2 Distance: 252.41  | 3 Distance: 254.26  | 1 Distance: 50.93  | 2 Distance: 51.73  | 3 Distance: 51.78  |
|  | 1 Distance: 246.93  | 2 Distance: 252.47  | 3 Distance: 267.58  | 1 Distance: 45.96  | 2 Distance: 46.57  | 3 Distance: 47.46  |
|  | 1 Distance: 291.13  | 2 Distance: 302.68  | 3 Distance: 307.72  | 1 Distance: 35.47  | 2 Distance: 36.6  | 3 Distance: 38.97  |

Taux de performance pour SIFT : 0.75
Taux de performance pour HOG : 0.5

5.2 Hypothèse 2

La deuxième hypothèse était que HOG aurait de meilleurs résultats pour identifier des objets de forme caractéristique, mais d'allure ou de modèle légèrement différent. Les images requêtes les plus pertinentes à considérer pour confirmer cette hypothèse sont la voiture, le visage et l'avion.

TABLE 5.2-1 TABLEAU DES RÉSULTATS DE L'HYPOTHÈSE 2

| Image requête | Classement Top 3 pour SIFT | | | Classement Top 3 pour HOG | | |
|---|---|---|--|---|--|--|
|  | 1 Distance: 314.14  | 2 Distance: 316.11  | 3 Distance: 316.68  | 1 Distance: 42.48  | 2 Distance: 43.63  | 3 Distance: 43.92  |
|  | 1 Distance: 256.21  | 2 Distance: 272.93  | 3 Distance: 273.24  | 1 Distance: 41.41  | 2 Distance: 41.88  | 3 Distance: 42.27  |
|  | 1 Distance: 243.53  | 2 Distance: 255.39  | 3 Distance: 256.77  | 1 Distance: 46.73  | 2 Distance: 47.78  | 3 Distance: 48.19  |

Taux de performance pour SIFT :0.78

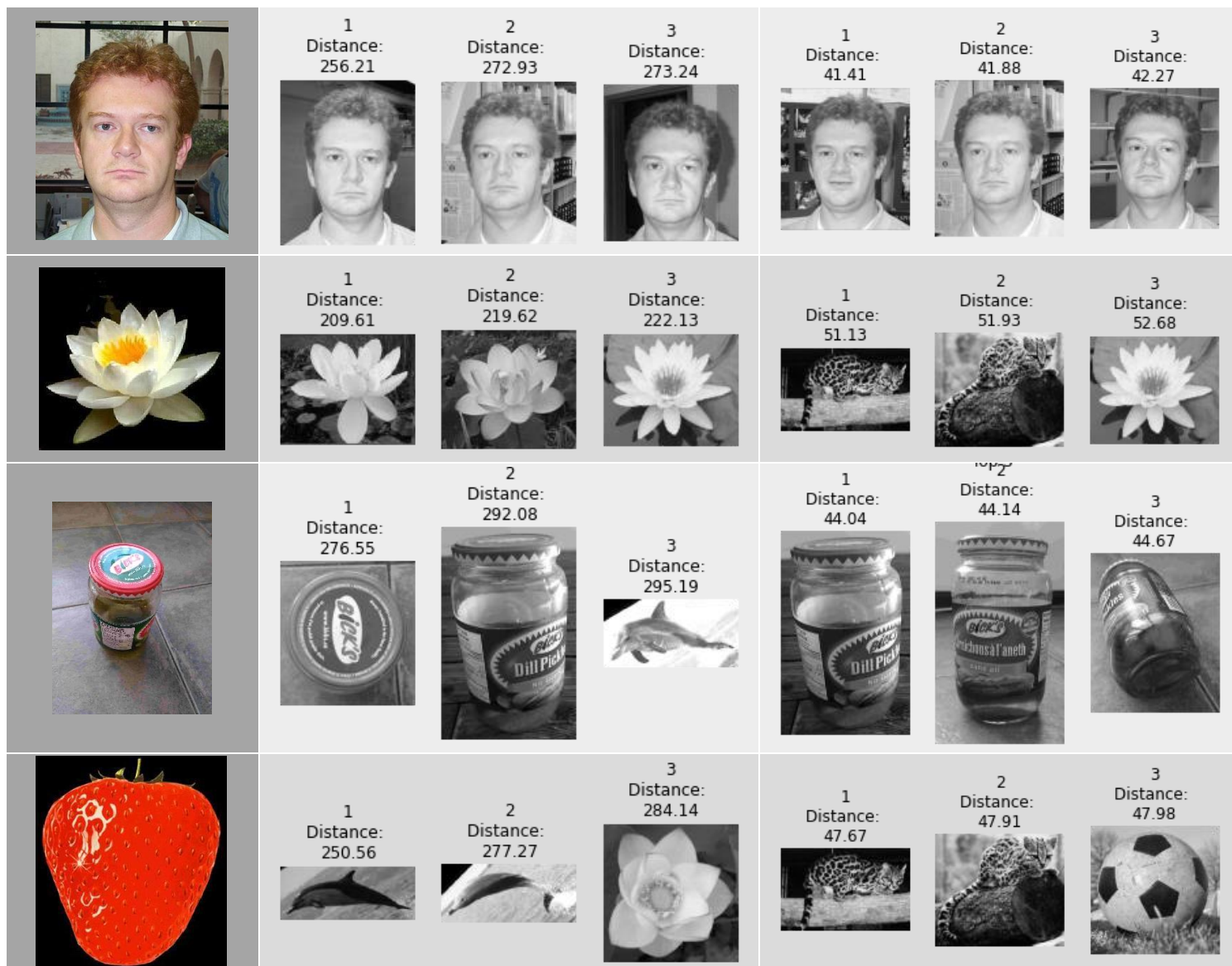
Taux de performance pour HOG :0.89

5.3Hypothèse 3

La troisième hypothèse était que le fait d'apposer des boîtes englobantes sur l'objet en question allait servir à augmenter les performances de HOG face à SIFT pour l'identification des objets requêtes face à la base de données. Toutes les images requêtes seront considérés pour confirmer cette hypothèse.

TABLE 5.3-1 TABLEAU DES RÉSULTATS DE L'HYPOTHÈSE 3

| Image requête | Classement Top 3 pour SIFT | | | Classement Top 3 pour HOG | | |
|---|---|---|---|--|--|--|
|  | 1 Distance: 256.61  | 2 Distance: 259.0  | 3 Distance: 260.24  | 1 Distance: 49.29  | 2 Distance: 49.37  | 3 Distance: 49.62  |
|  | 1 Distance: 229.33  | 2 Distance: 236.7  | 3 Distance: 238.19  | 1 Distance: 48.09  | 2 Distance: 48.34  | 3 Distance: 48.89  |
|  | 1 Distance: 278.08  | 2 Distance: 278.18  | 3 Distance: 294.28  | 1 Distance: 42.98  | 2 Distance: 43.32  | 3 Distance: 44.57  |
|  | 1 Distance: 301.91  | 2 Distance: 308.05  | 3 Distance: 308.69  | 1 Distance: 43.98  | 2 Distance: 43.99  | 3 Distance: 44.7  |
|  | 1 Distance: 251.98  | 2 Distance: 263.06  | 3 Distance: 268.66  | 1 Distance: 46.19  | 2 Distance: 46.36  | 3 Distance: 47.85  |



Taux de performance pour SIFT :0.41

Taux de performance pour HOG :0.56

5.4 Aparté sur le temps d'exécution des méthodes

Comme base de comparaison des approches, le tableau suivant sert à modéliser les performances de temps d'exécution des deux méthodes pour l'obtention des descripteurs (avec les hyperparamètres définis plus haut):

TABEAU 5.4-1 TABLEAU DE TEMPS D'EXÉCUTION ENTRE SIFT ET HOG POUR L'OBTENTION DE LEUR DESCRIPTEUR

| Caractéristique | Performances pour SIFT | Performance pour HOG |
|--------------------|------------------------|----------------------|
| Répétition du test | 40 | 40 |
| Moyenne de temps | 0.098712 | 0.091345 |
| Écart type | 0.021929 | 0.021918 |

| | | |
|------------------------------|----------|----------|
| Temps le plus petit | 0.082196 | 0.076999 |
| Temps au 1er quartile | 0.084969 | 0.081014 |
| Temps médian | 0.090202 | 0.083589 |
| Temps au 3e quartile | 0.099174 | 0.086225 |
| Temps le plus grand | 0.172752 | 0.191575 |

Pour compléter sur le temps d'exécution, le temps requis pour la comparaison entre deux descripteurs de l'image est représenté dans le tableau suivant (avec les hyperparamètres définis plus haut) :

TABEAU 5.4-2 TABLEAU DE TEMPS D'EXÉCUTION ENTRE SIFT ET HOG POUR UNE COMPARAISON DE LEUR DESCRIPTEUR

| Caractéristique | Performances pour SIFT | Performance pour HOG |
|------------------------------|-------------------------------|-----------------------------|
| Répétition du test | 40 | 40 |
| Moyenne de temps | 0.003100 | 0.002756 |
| Écart type | 0.002610 | 0.001388 |
| Temps le plus petit | 0.000000 | 0.001034 |
| Temps au 1er quartile | 0.001379 | 0.001985 |
| Temps médian | 0.002252 | 0.002030 |
| Temps au 3e quartile | 0.003684 | 0.003861 |
| Temps le plus grand | 0.013002 | 0.008173 |

6 Discussion des résultats:

Dans cette section, un retour sur les résultats des hypothèses est fait. Entre autres, une discussion présente si l'hypothèse est confirmée indéterminée ou erronée. Dans chaque cas, une explication résultante des observations encourues lors des manipulations et de la théorie est montrée. Finalement, une sous-section sur des pistes d'amélioration possible est fournie.

6.1 Discussion Hypothèse 1

Pour rappel, la première hypothèse stipulait que SIFT est plus robuste que HOG pour l'identification d'un objet par rapport à des rotations en plan et hors plan de l'image.

Face aux résultats de performance obtenus par rapport aux images de la balle, du lotus, du dauphin et du pot de cornichons, on observe par notre métrique un taux de performance de 0.75 pour SIFT contre 0.5 pour HOG. Ceci nous permet de conclure sur la validité de la première hypothèse.

Entre autres, parmi toutes les images requêtes testées pour cette hypothèse, la seule catégorie d'image où SIFT a moins bien performé que HOG est le pot de cornichons. L'explication trouvée à ce phénomène est que la texture du plancher en arrière-plan du bocal aide la performance de HOG. Les trois images de pot de cornichons retournées par HOG possèdent toutes une grande région avec le même type de plancher, HOG identifie alors plus dans ce cas l'arrière-plan que l'objet en lui-même. Pour SIFT, il est

possible dans cette situation que les points clés du plancher soient filtrés par les hyperparamètres de détection de contraste. De plus, de larges variations de rotation hors plan du bocal de cornichon sont présentes entre les images de la base de données. On en conclut que même si SIFT se veut tolérant à ce type de rotation, la méthode possède certaines limites d'angle pour reconnaître un objet.

On remarque d'un autre côté que HOG pâlit sur les images de la balle et du lotus. Ces objets prennent une grande proportion sur la taille totale de l'image. Pour des rotations de ces objets, HOG a des risques de devenir sensible, car la technique se base sur un histogramme d'orientation des gradients sur des régions de l'image et non sur des régions de l'objet. Donc, de bouger une grande partie des textures de l'image à la possibilité de beaucoup changer l'histogramme d'orientation des blocs et donc de nuire à l'identification.

6.2 Discussion Hypothèse 2

Un rappel de la deuxième hypothèse était que HOG donnerait de meilleurs résultats que SIFT pour des objets aux formes caractéristiques.

L'échantillonnage d'intérêt pour la vérification de l'hypothèse est représenté par la voiture, le visage et l'avion. Le taux de performance observé face à notre métrique dénotait un taux de 0.89 pour HOG contre 0.78 pour SIFT. L'hypothèse est alors confirmée.

Même si SIFT est une approche plus robuste, il n'en reste pas moins qu'elle est extrêmement précise sur son identification. De ce fait, elle perd en performance sur l'identification d'une classe générale d'objet: par exemple, pour un cas qui se rapproche du nôtre, si on souhaite identifier une voiture sans vraiment faire attention à la marque ou au modèle de celle-ci. On en conclue que SIFT est alors plus spécialisé pour retrouver exactement la même instance d'un objet caché dans une scène ou sur des plans plus variés. Entre autres, ceci expliquerait pourquoi elle est restée robuste avec l'image du visage. D'un autre côté, HOG se concentre sur une direction générale de gradients concentrés dans des blocs de l'image et laisse du coup plus de lousse à un modèle changeant de l'objet à identifier.

6.3 Discussion Hypothèse 3

L'hypothèse 3 stipulait que le fait d'utiliser des boîtes englobantes allait aider la performance de HOG et diminuer la performance de SIFT.

L'échantillonnage d'intérêt comprenait l'intégralité des images requêtes et le taux de performance obtenue par notre métrique était de 0.41 pour SIFT contre 0.55 pour HOG. On remarque alors que SIFT est beaucoup plus sensible à cette opération, mais que HOG n'a pas nécessairement un gain de performance conséquent. L'hypothèse est alors indéterminée.

Pour SIFT, on remarque que l'identification du visage et l'identification du lotus sont les seules images requêtes n'obtenant pas une baisse de performance liée à l'utilisation de boîtes englobantes. Ces images requêtes sont celle où l'objet prend la majeure partie des pixels de l'image. Il est alors à penser que dans un cas général, augmenter de beaucoup le ratio entre l'objet et le fond des images dans la base de données est problématique. SIFT se veut robuste à un changement d'échelle de l'objet dans l'image en trouvant ces points clés entre plusieurs octaves. Cependant, face à notre méthode de procédé, parce qu'on remet toutes les images sur la même échelle, on rend plus gros la représentation de l'objet identifié dans une boîte englobante par rapport à l'image requête. Un grossissement de l'objet n'est alors pas un facteur caractéristique invariant pour SIFT, d'où la baisse de performance.

Pour HOG, si on compare l'utilisation des boîtes englobantes entre ces résultats pour les hypothèses du point 2 avec les résultats du point 3, seul l'avion possède une baisse de performance avec l'application de la métrique de l'expérience. Les autres images requêtes possèdent le même résultat et donc un taux

de performance semblable. La tendance porte alors à croire que l'utilisation de boîte englobante n'est pas un facteur impactant la performance de détection. Certains comportements peuvent cependant être considérés comme différents sur l'identification des images. Entre autres, les images ressorties de la base de données sont beaucoup plus en lien avec la théorie voulue pour HOG. Effectivement, l'arrière-plan de l'image n'impactant plus la performance, on retrouve des images correctement identifiées possédant à la fois, peu de rotation en plan de l'image, peu de rotation hors plan de l'image, peu de changement de proportion de l'objet ou peu d'occlusion de l'objet dans l'image.

6.4 Discussion sur le temps d'exécution des méthodes.

Sur nos observations concernant le temps d'exécution pour obtenir les résultats des méthodes, on possède, comme le voudrait la théorie, une direction globale de rapidité en faveur de HOG. Cependant, le résultat n'est pas uniforme sur toutes les expériences et la différence de temps entre HOG et SIFT se rapproche beaucoup trop pour être concluant.

Il serait à penser que nos choix d'hyperparamètre pour SIFT seraient la cause de ce phénomène. Pour rappel, l'optimisation nous donne un nombre d'octaves de 1 à considérer. Cette considération diminue significativement le nombre d'itérations d'acquisition de descripteur pour SIFT. Entre autres, on calcule un moins grand nombre de DOG et on possède un moins grand nombre de points clés à considérer. Il serait alors, dans ce cas spécifique et pour des résolutions d'image moindre, possible de croire que SIFT serait presque équivalent à HOG. Pour montrer la relation entre le nombre d'octaves et le temps d'exécution pour SIFT, un graphe est laissé [en annexe](#): on remarque que plus le nombre d'octaves augmente, plus le temps d'exécution augmente.

6.5 Amélioration générale des tests exercés

Un retour sur la fiabilité des résultats est à reconsidérer pour cette expérience. Plusieurs mesures ont été testées lors de ces expériences. Cependant celles-ci se basent sur des optimisations d'hyperparamètres par rapport à une petite quantité de données. Ainsi, une piste d'amélioration serait d'augmenter le nombre d'échantillons de la base de données. Par exemple, il est possible d'utiliser la méthode SinGAN (cf. ressource [11] [en annexe](#)) pour obtenir une base de données plus diversifiée pour notre optimisation et espérer augmenter la fiabilité des résultats obtenus.

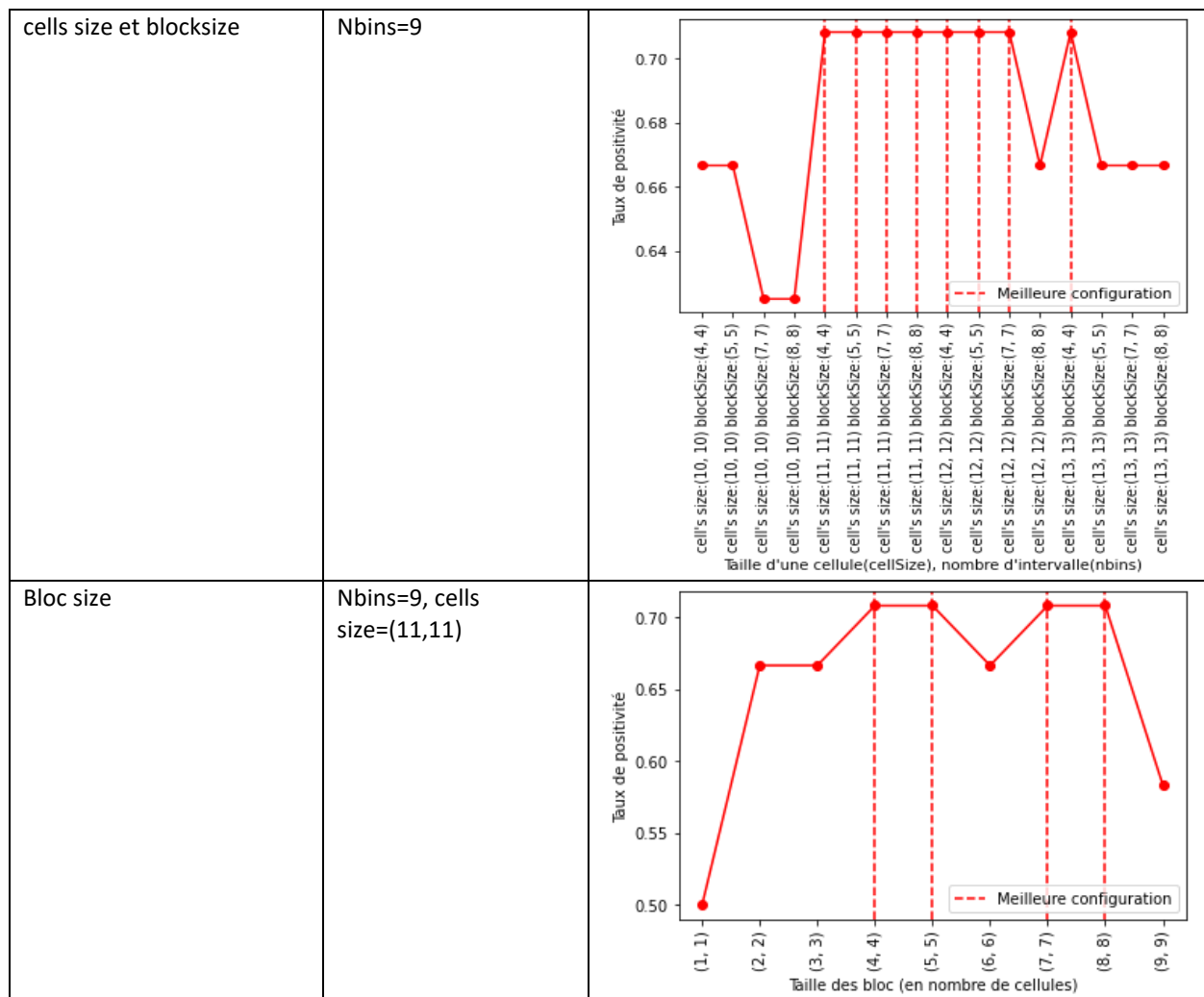
Par la suite, il aurait été intéressant d'obtenir des images avec un comportement caractéristique d'identifications plus ciblé. Par exemple, avoir des images ne variant qu'en rotation ou qu'en translation. Ceci ferait en sorte d'avoir des observations permettant de mieux conclure sur les hypothèses énumérées.

Un autre aspect d'amélioration serait d'avoir en main un nombre équivalent d'images requêtes ayant des images similaires dans la base de données que d'images requête n'ayant des images similaires dans la base de données (comme `strawberry_query.jpg`). Comme expliqué dans la méthodologie d'identification des hyperparamètres, avec seulement l'image de la fraise, pour laquelle aucune image de la base de données devrait être proposée par l'algorithme, l'optimisation du seuil de rejet nous indique qu'il ne faudrait tout simplement pas en avoir pour conserver la performance. Avec plus d'images requêtes comme celle de la fraise, il serait possible d'avoir une meilleure représentation pour définir un seuil de rejet plus significatif.

Annexe A – Graphe d’optimisation des hyperparamètres

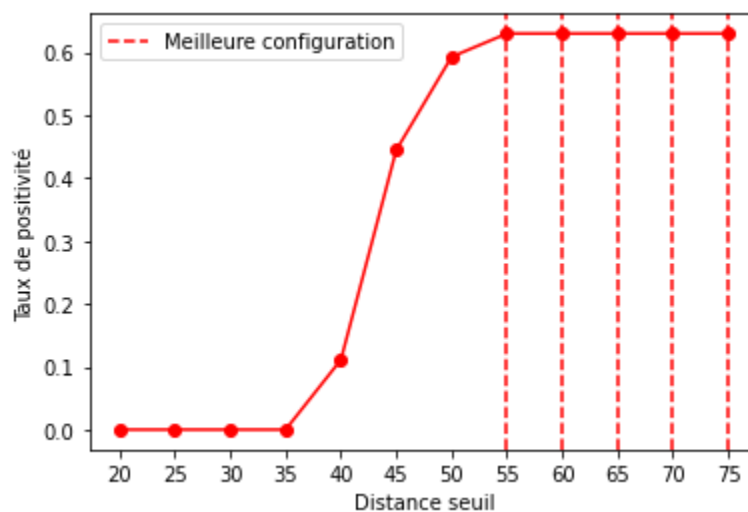
Hyperparamètres pour HOG:

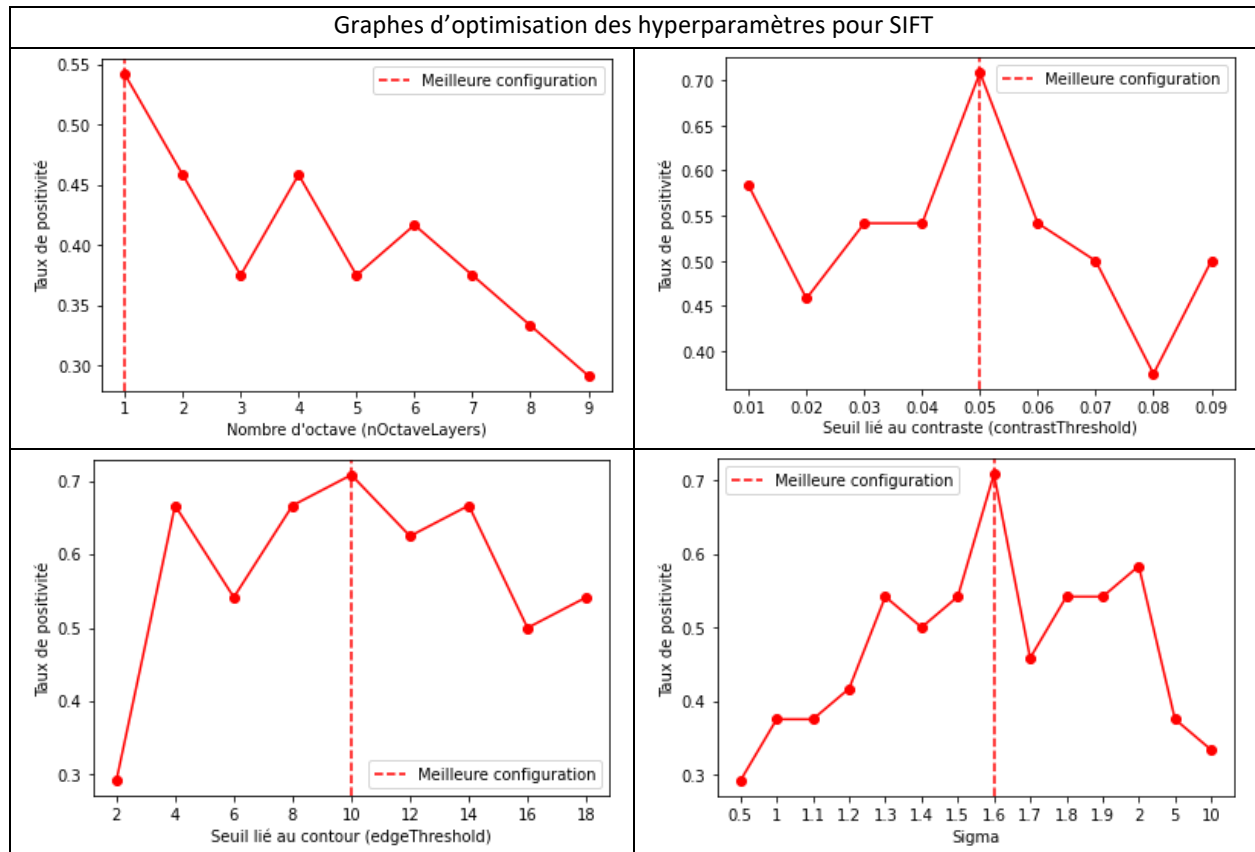
| Paramètre optimiser | Paramètre fixer précédemment | Graphe d'optimisation pour HOG | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|------------------------------|---|---|--------------------|--------|------|--------|------|--------|------|--------|------|--------|------|--------|------|--------|------|--------|------|--------|------|--------|------|--------|------|--------|------|--------|------|--------|------|--------|------|---------|------|---------|------|---------|------|---------|------|---------|------|---------|------|---|--------------------|--------|------|---------|------|---------|------|---------|------|----------|------|----------|------|---------|------|----------|------|----------|------|---------|------|----------|------|----------|------|---------|------|----------|------|----------|------|----------|------|----------|------|
| Nbins (nombre d'intervalle) | - | <p>Taux de positivité</p> <p>Nombre d'intervalles (nbins)</p> <p>Meilleure configuration</p> <table><thead><tr><th>Nombre d'intervalles (nbins)</th><th>Taux de positivité</th></tr></thead><tbody><tr><td>2</td><td>0.29</td></tr><tr><td>3</td><td>0.25</td></tr><tr><td>4</td><td>0.54</td></tr><tr><td>5</td><td>0.42</td></tr><tr><td>6</td><td>0.58</td></tr><tr><td>7</td><td>0.46</td></tr><tr><td>8</td><td>0.58</td></tr><tr><td>9</td><td>0.46</td></tr><tr><td>10</td><td>0.58</td></tr><tr><td>11</td><td>0.54</td></tr><tr><td>12</td><td>0.50</td></tr><tr><td>13</td><td>0.54</td></tr><tr><td>14</td><td>0.46</td></tr></tbody></table> | Nombre d'intervalles (nbins) | Taux de positivité | 2 | 0.29 | 3 | 0.25 | 4 | 0.54 | 5 | 0.42 | 6 | 0.58 | 7 | 0.46 | 8 | 0.58 | 9 | 0.46 | 10 | 0.58 | 11 | 0.54 | 12 | 0.50 | 13 | 0.54 | 14 | 0.46 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Nombre d'intervalles (nbins) | Taux de positivité | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 0.29 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | 0.25 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | 0.54 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | 0.42 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | 0.58 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | 0.46 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | 0.58 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | 0.46 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10 | 0.58 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 11 | 0.54 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 12 | 0.50 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 13 | 0.54 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 14 | 0.46 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Nbins et cells size | - | <p>Taux de positivité</p> <p>Taille d'une cellule(cellSize), nombre d'intervalle(nbins)</p> <p>Meilleure configuration</p> <table><thead><tr><th>cell's size (cellSize), nombre d'intervalle (nbins)</th><th>Taux de positivité</th></tr></thead><tbody><tr><td>(1, 1)</td><td>0.21</td></tr><tr><td>(1, 3)</td><td>0.21</td></tr><tr><td>(1, 5)</td><td>0.18</td></tr><tr><td>(3, 1)</td><td>0.58</td></tr><tr><td>(3, 3)</td><td>0.58</td></tr><tr><td>(3, 5)</td><td>0.46</td></tr><tr><td>(5, 1)</td><td>0.54</td></tr><tr><td>(5, 3)</td><td>0.58</td></tr><tr><td>(5, 5)</td><td>0.54</td></tr><tr><td>(7, 1)</td><td>0.58</td></tr><tr><td>(7, 3)</td><td>0.54</td></tr><tr><td>(7, 5)</td><td>0.58</td></tr><tr><td>(9, 1)</td><td>0.54</td></tr><tr><td>(9, 3)</td><td>0.58</td></tr><tr><td>(9, 5)</td><td>0.50</td></tr><tr><td>(11, 1)</td><td>0.62</td></tr><tr><td>(11, 3)</td><td>0.54</td></tr><tr><td>(11, 5)</td><td>0.62</td></tr><tr><td>(13, 1)</td><td>0.54</td></tr><tr><td>(13, 3)</td><td>0.62</td></tr><tr><td>(13, 5)</td><td>0.62</td></tr></tbody></table> <p>Taux de positivité</p> <p>Taille d'une cellule(cellSize), nombre d'intervalle(nbins)</p> <p>Meilleure configuration</p> <table><thead><tr><th>cell's size (cellSize), nombre d'intervalle (nbins)</th><th>Taux de positivité</th></tr></thead><tbody><tr><td>(9, 9)</td><td>0.54</td></tr><tr><td>(9, 10)</td><td>0.58</td></tr><tr><td>(9, 11)</td><td>0.62</td></tr><tr><td>(10, 9)</td><td>0.54</td></tr><tr><td>(10, 10)</td><td>0.58</td></tr><tr><td>(10, 11)</td><td>0.62</td></tr><tr><td>(11, 9)</td><td>0.50</td></tr><tr><td>(11, 10)</td><td>0.58</td></tr><tr><td>(11, 11)</td><td>0.62</td></tr><tr><td>(12, 9)</td><td>0.54</td></tr><tr><td>(12, 10)</td><td>0.58</td></tr><tr><td>(12, 11)</td><td>0.62</td></tr><tr><td>(13, 9)</td><td>0.54</td></tr><tr><td>(13, 10)</td><td>0.58</td></tr><tr><td>(13, 11)</td><td>0.62</td></tr><tr><td>(13, 12)</td><td>0.54</td></tr><tr><td>(13, 13)</td><td>0.62</td></tr></tbody></table> | cell's size (cellSize), nombre d'intervalle (nbins) | Taux de positivité | (1, 1) | 0.21 | (1, 3) | 0.21 | (1, 5) | 0.18 | (3, 1) | 0.58 | (3, 3) | 0.58 | (3, 5) | 0.46 | (5, 1) | 0.54 | (5, 3) | 0.58 | (5, 5) | 0.54 | (7, 1) | 0.58 | (7, 3) | 0.54 | (7, 5) | 0.58 | (9, 1) | 0.54 | (9, 3) | 0.58 | (9, 5) | 0.50 | (11, 1) | 0.62 | (11, 3) | 0.54 | (11, 5) | 0.62 | (13, 1) | 0.54 | (13, 3) | 0.62 | (13, 5) | 0.62 | cell's size (cellSize), nombre d'intervalle (nbins) | Taux de positivité | (9, 9) | 0.54 | (9, 10) | 0.58 | (9, 11) | 0.62 | (10, 9) | 0.54 | (10, 10) | 0.58 | (10, 11) | 0.62 | (11, 9) | 0.50 | (11, 10) | 0.58 | (11, 11) | 0.62 | (12, 9) | 0.54 | (12, 10) | 0.58 | (12, 11) | 0.62 | (13, 9) | 0.54 | (13, 10) | 0.58 | (13, 11) | 0.62 | (13, 12) | 0.54 | (13, 13) | 0.62 |
| cell's size (cellSize), nombre d'intervalle (nbins) | Taux de positivité | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (1, 1) | 0.21 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (1, 3) | 0.21 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (1, 5) | 0.18 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (3, 1) | 0.58 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (3, 3) | 0.58 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (3, 5) | 0.46 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (5, 1) | 0.54 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (5, 3) | 0.58 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (5, 5) | 0.54 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (7, 1) | 0.58 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (7, 3) | 0.54 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (7, 5) | 0.58 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (9, 1) | 0.54 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (9, 3) | 0.58 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (9, 5) | 0.50 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (11, 1) | 0.62 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (11, 3) | 0.54 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (11, 5) | 0.62 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (13, 1) | 0.54 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (13, 3) | 0.62 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (13, 5) | 0.62 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cell's size (cellSize), nombre d'intervalle (nbins) | Taux de positivité | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (9, 9) | 0.54 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (9, 10) | 0.58 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (9, 11) | 0.62 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (10, 9) | 0.54 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (10, 10) | 0.58 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (10, 11) | 0.62 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (11, 9) | 0.50 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (11, 10) | 0.58 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (11, 11) | 0.62 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (12, 9) | 0.54 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (12, 10) | 0.58 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (12, 11) | 0.62 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (13, 9) | 0.54 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (13, 10) | 0.58 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (13, 11) | 0.62 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (13, 12) | 0.54 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (13, 13) | 0.62 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |



Hyperparamètre: Nbins=9, cells size=(11,11), blockSize=(7,7)

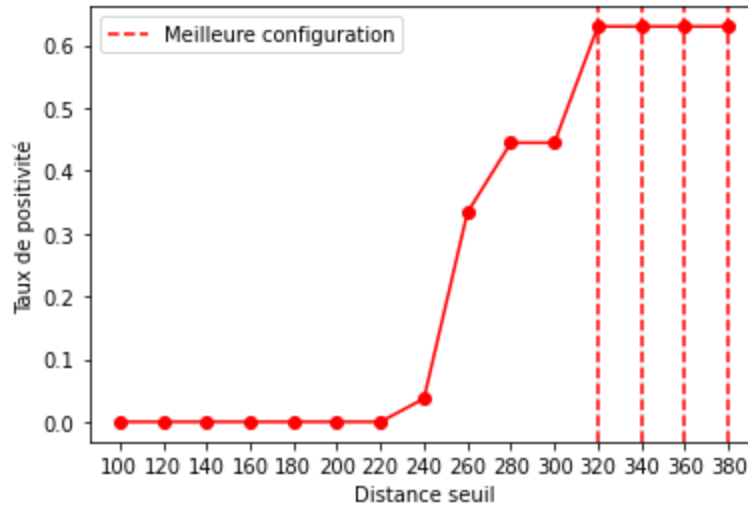
Recherche de la distance seuil au-delà de laquelle deux images ne sont plus considéré comme similaire (qui n'a finalement pas été fixé pour éviter de baisser les performances):



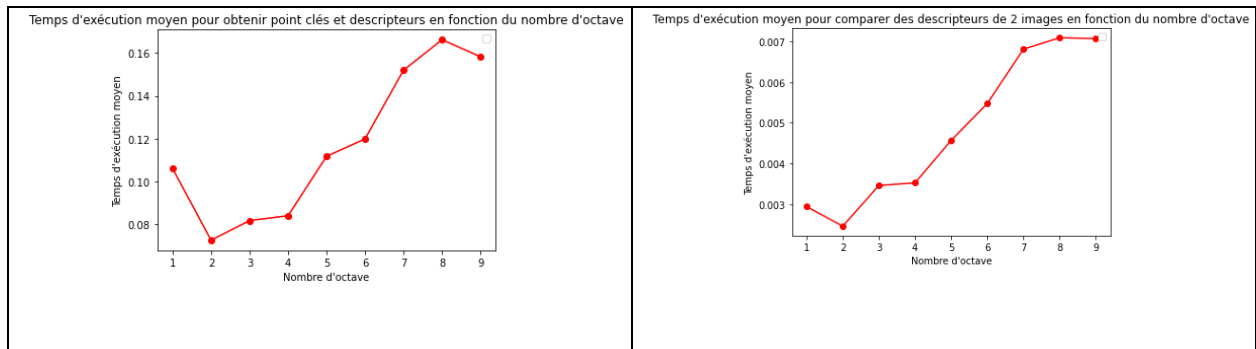


Hyperparamètre: nOctaveLayers=1, constratThreshold=0.05, edgeThreshold=10, Sigma= 1.6

Recherche de la distance seuil au-delà de laquelle deux images ne sont plus considéré comme similaire (qui n'a finalement pas été fixé pour éviter de baisser les performances):



Annexe B – Graphe du temps d’exécution par rapport au nombre d’octave considéré pour SIFT :



Annexe C – Bibliographie

Méthode HOG:

- [1] Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. *Http : //Lear.Inrialpes.Fr*. <https://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf>
- [2] *OpenCV : cv : : HOGDescriptor Struct Reference*. (s. d.). https://docs.opencv.org/4.x/d5/d33/structcv_1_1HOGDescriptor.html
- [3] Wikipedia contributors. (2020, 6 novembre). *Histogramme de gradient orienté*. https://fr.wikipedia.org/wiki/Histogramme_de_gradient_orient%C3%A9
- [4] *Histogram of Oriented Gradients — skimage v0.19.2 docs*. (s. d.). https://scikit-image.org/docs/stable/auto_examples/features_detection/plot_hog.html
- [5] *Module : feature — skimage v0.19.2 docs*. (s. d.). <https://scikit-image.org/docs/stable/api/skimimage.feature.html>

Quelle page internet aussi visitée:

<https://stackoverflow.com/questions/28390614/opencv-hogdescriptor-python>
<https://stackoverflow.com/questions/6090399/get-hog-image-features-from-opencv-python>
<https://stackoverflow.com/questions/44972099/opencv-hog-features-explanation>

Méthode SIFT:

- [6] *OpenCV : Introduction to SIFT (Scale-Invariant Feature Transform)*. (s. d.). https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html
- [7] *Moodle : Polytechnique Montréal : Se connecter sur le site*. (s. d.). <https://moodle.polymtl.ca/login/index.php>
- [8] Wikipedia contributors. (2022, 30 novembre). *Scale-invariant feature transform*. https://fr.wikipedia.org/wiki/Scale-invariant_feature_transform
- [9] Tyagi, D. (2021, 9 décembre). *Introduction to SIFT(Scale Invariant Feature Transform)*. Medium. <https://medium.com/data-breach/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40>
- [10] Singh, A. (2020, 23 décembre). *A Detailed Guide to the Powerful SIFT Technique for Image Matching (with Python code)*. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2019/10/detailed-guide-powerful-sift-technique-image-matching-python/>

Autre:

- [11] Tamar Rott Shaham, Tali Dekel, Tomer Michaeli (2019): *SinGAN: Learning a Generative Model from a Single Natural Image*

https://openaccess.thecvf.com/content_ICCV_2019/papers/Shaham_SinGAN_Learning_a_Generative_Model_From_a_Single_Natural_Image_ICCV_2019_paper.pdf