COMPTE-RENDU
TP DEEP LEARNING
Description d'images par SIFT et Bag of Words

Rédigé par :
Elodie Difonzo
Roqyun Ko

Objectif du TP

Lors de ces premières séances de TP nous avons codé un algorithme permettant de **classifier** un grand nombre d'images **en 15 catégories** de scènes d'intérieur et d'extérieur.
Le TP s'est divisé en 3 parties :

- Description d'images par **SIFT** (Scale-invariant feature transform)
- Constitution d'un **dictionnaire visuel**
- Utilisation de la technique du **BoW** (Bag of Words)

Nous reviendrons plus en détails sur chacune de ces parties dans ce compte rendu.

I. Description d'images par SIFT

Pour classifier l'ensemble des images fournie, il est nécessaire de pouvoir détecter les éléments semblables se retrouvant sur plusieurs de ces images. Or l'orientation et l'échelle des images ne sont pas toujours identiques.
Pour résoudre cette problématique, nous allons utiliser l'algorithme SIFT que l'on définit comme étant un **descripteur local**. Il permet de transformer un patch d'image en un vecteur qui le représente numériquement.
L'objectif de cet algorithme est de connaître les éléments caractéristiques d'une image donnée.

Le traitement décrit ci-dessous s'applique sur des patchs de l'image de taille 16 x 16 pixels.

La première étape consiste à calculer le gradient de chaque pixel du patch. En effet, le gradient permet de déterminer la direction des variations d'intensité au voisinage de chaque pixel.
Dans la pratique, on simplifie ce calcul par une approximation des dérivées partielles en utilisant les masques de Sobel 3 x 3.
L'application de ce masque permet la détection des contours de l'image. L'intérêt de séparer le filtre de convolution est de **détecter les contours** horizontaux et verticaux de façon distincte.

La matrice d'orientation des gradients est ensuite discrétisée pour obtenir une nouvelle matrice contenant des valeurs comprises entre 0 et 8 selon l'orientation du gradient. La matrice de norme des gradients est quant à elle pondérée par l'application d'un masque gaussien. Cette étape permet de réduire le bruit.

On divise maintenant le patch en plusieurs régions de taille 4 x 4 pour calculer l'histogramme des orientations des gradients sur chacune d'elle.

Pour finir, on concatène les 16 vecteurs obtenus pour former le descripteur.

Questions - Partie 1

1. Montrer que les masques M_x et M_y sont séparables, c'est à dire qu'ils peuvent s'écrire:

$$M_x = h_y \times h_x^T$$
$$M_y = h_x \times h_y^T$$

avec h_x et h_y deux vecteurs colonne de taille 3 à déterminer.

$$h_x = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

$$h_y = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

---

2. Quel est l'intérêt de séparer le filtre de convolution ?

Soit *I* une matrice représentant une image de taille quelconque, I_{3x3} est un patch de la taille 3x3 de l'image.
Le filtre de convolution pondère le patch par masque de la même taille et fait la somme des éléments de la matrice obtenue afin d'obtenir un Pix_{conv} :

Soit :

$$Patch_{masked} = I_{3x3} \times M_x = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

Alors, la convolution donne le résultat:

$$Pix_{conv} = I_{3x3} * M_x = a + b + c + d + e + f + g + h + i$$

Dans l'équation ci-dessus, on effectue 9 fois la multiplication pour obtenir $Patch_{masked}$ mais les calculs peuvent être simplifiés en utilisant les vecteurs h_x et h_y:

$$Patch_{masked} = I_{3x3} \times h_y \times h_x^T \quad où \quad h_y \times h_x^T = M_x$$

Donc,

$IM_y = I_{3x3} \times h_y$ fait une matrice de la taille 3x1 ***en faisant 3 multiplications***

$IM_y \times h_x^T$ fait la matrice $Patch_{masked}$ de la taille 3x3 ***en faisant 3 multiplications***

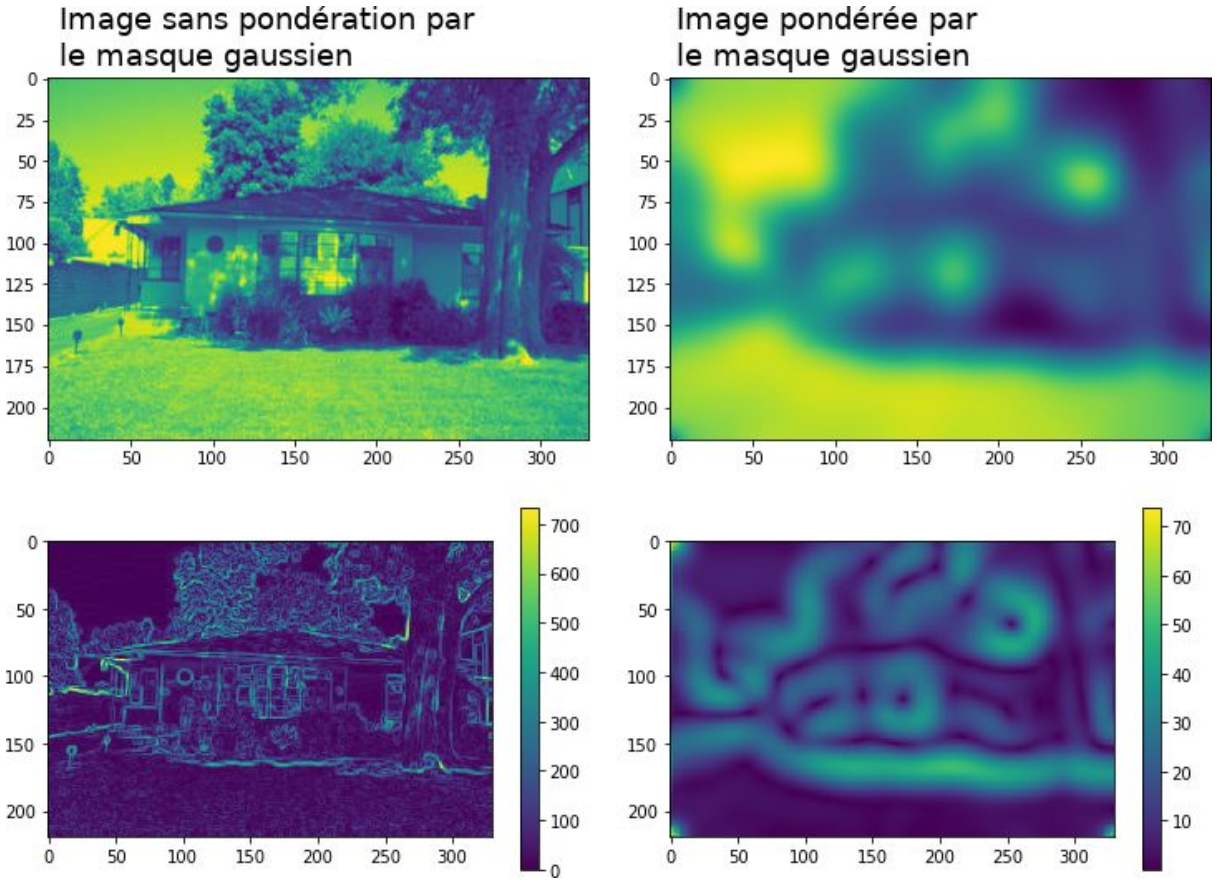
Alors qu'on obtient le même résultat, on peut réduire le nombre de multiplications à $2 \cdot log_N(N^2) = 2 \cdot N$ *de* N^2 *(Les masques sont de la taille NxN)*

L'intérêt de séparer le filtre de convolution est de réduire le nombre de calculs.

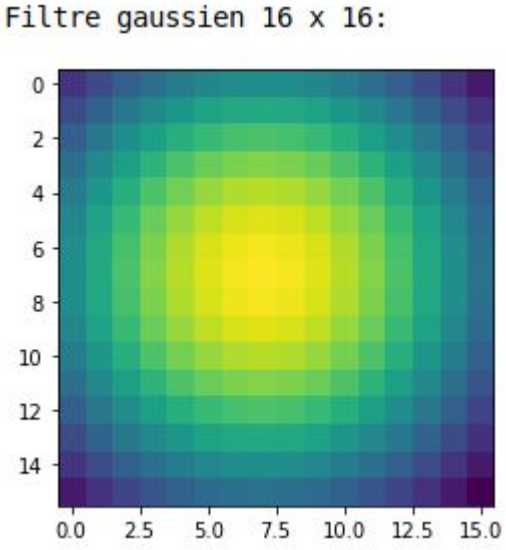
---

3. Quel est le rôle de la pondération par masque gaussien ?

Le filtre gaussien sert à brouiller des images afin de ne capter que les contours importantes.



La visualisation du masque :



Comme le masque de Sobel, on applique la convolution pour obtenir l'effet désiré (brouiller).

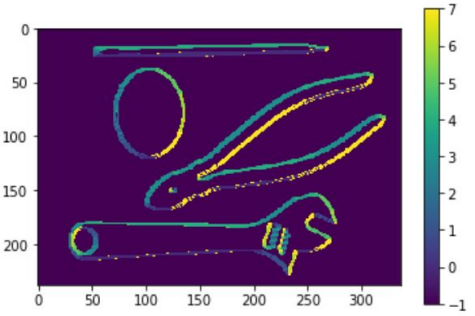
4. Expliquez le rôle de la discrétisation des orientations

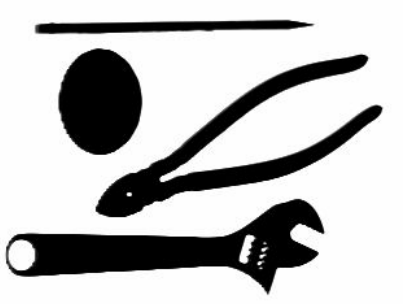
La discrétisation des orientations sert à identifier l'orientation du bord dans un patch. On peut vérifier cela en comparant l'image originale et l'image dont les orientations sont discrétisées :

```
In [12]: def compute_grad_mod_ori(I):
             # TODO
             Ix, Iy = compute_grad(I)
             Gm = np.sqrt(Ix*Ix + Iy*Iy)
             Go = compute_grad_ori(Ix,Iy,Gm)
             return Gm, Go

         Gm,Go = compute_grad_mod_ori(I)
         plt.imshow(Go)
         plt.colorbar()

Out[12]: <matplotlib.colorbar.Colorbar at 0x7fdd745
```





Les bords orientés vers le haut sont en bleu et les bords orientés vers le bas sont en jaune.

---

5. Justifiez l'intérêt des différents post-processings appliqués au SIFT

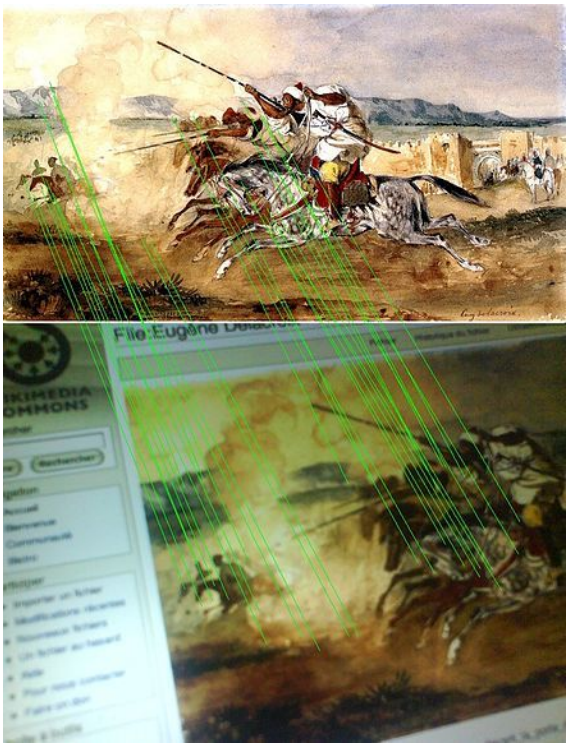
Si la norme 2 du descripteur P_{enc} est inférieure à un seuil de 0.5, la région où l'on dérive le SIFT du patch est vide et elle n'a aucune signification.

Seuiller les valeurs supérieures à 0,2 après la normalisation du vecteur P_{enc} sert à supprimer l'effet d'illumination dans les images dont on a obtenu les SIFTs. On obtient alors les SIFTs qui sont indépendant de l'intensité de la lumière.

Source:
https://medium.com/software-incubator/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40

---

6. SIFT est un nom abrégé de **S**cale **I**nvariant **F**eature **T**ransform. La méthode SIFT a des avantages certains dans la détection des éléments caractéristiques d'une image. En effet, elle reste efficace en cas d'illumination différente (post-processing), de changement d'échelle ou de rotation de l'image.



Source : https://fr.wikipedia.org/wiki/Scale-invariant_feature_transform

---

7. Interprétez les résultats que vous avez obtenus dans cette partie.

Tout d'abord, nous avons examiné 2 types de filtres : le filtre de sobel et le filtre gaussien.
Le premier sert à détecter les bords dans les images.
Le dernier sert à brouiller les images pour supprimer les détails (bords) inutiles dans les images.
Ensuite, nous avons identifié et discrétisé les orientations des contours obtenu après l'application du filtre de Sobel.
Enfin, après diviser l'image à analyser en plusieurs patchs, on calcule l'histogramme de chaque patch en comparant l'intensité du contours et leurs orientations.
En transformant l'histogramme obtenu en vecteur unidimensionnel, on obtient le vecteur SIFT.

II. Constitution d'un dictionnaire visuel

Après avoir extrait l'ensemble des descripteurs de la base d'images fournie, on souhaite maintenant constituer un dictionnaire visuel. L'objectif de cette partie est de regrouper les descripteurs SIFT en différentes classes possédant des similitudes.

Pour ce faire, nous allons nous servir de l'algorithme K-Means. Il s'agit d'une méthode d'apprentissage non supervisée, c'est à dire qu'on ne connaît pas à l'avance le nombre de classes optimal.

Cette méthode consiste à initialiser aléatoirement des centres parmi l'ensemble des points et assigner chaque point à son centre le plus proche. Les centres sont ensuite recalculés et on répète l'assignation jusqu'à la convergence.

Cet algorithme possède un certain nombre d'avantages. Tout d'abord, il est simple à implémenter et s'adapte facilement aux changements de données. Il est efficace même lorsque les données sont très nombreuses et fournit des résultats simples d'interprétation. Cependant, les centres initiaux étant tirés aléatoirement, cela peut mener à une incohérence. En effet, en l'exécutant plusieurs fois l'algorithme pourra aboutir à des résultats différents.

Questions - Partie 2

8. Justifiez la nécessité du dictionnaire dans le processus général de reconnaissance d'image que nous sommes en train de mettre en place.

Après avoir effectué l'analyse avec l'algorithme de SIFT, on peut obtenir les différents barycentres pour les différents types d'images en implémentant l'algorithme de K-Means. Autrement dit, les images qui appartiennent à la même catégorie vont avoir le même barycentre. L'ensemble de ces partitions représentent un dictionnaire visuel.

En conclusion, la création d'un dictionnaire visuel a pour objectif de classer les images selon leurs patterns. Donc elle est nécessaire pour le processus général de reconnaissance d'images que nous sommes en train de mettre en place.

---

9. Considérant les points $\{x_i\}_{i=1..n}$ assignés à un cluster c, montrer que le centre du cluster qui minimise la dispersion est bien le barycentre (moyenne) des points x_i : $\min_c \sum_i \| x_i - c \|_2^2$

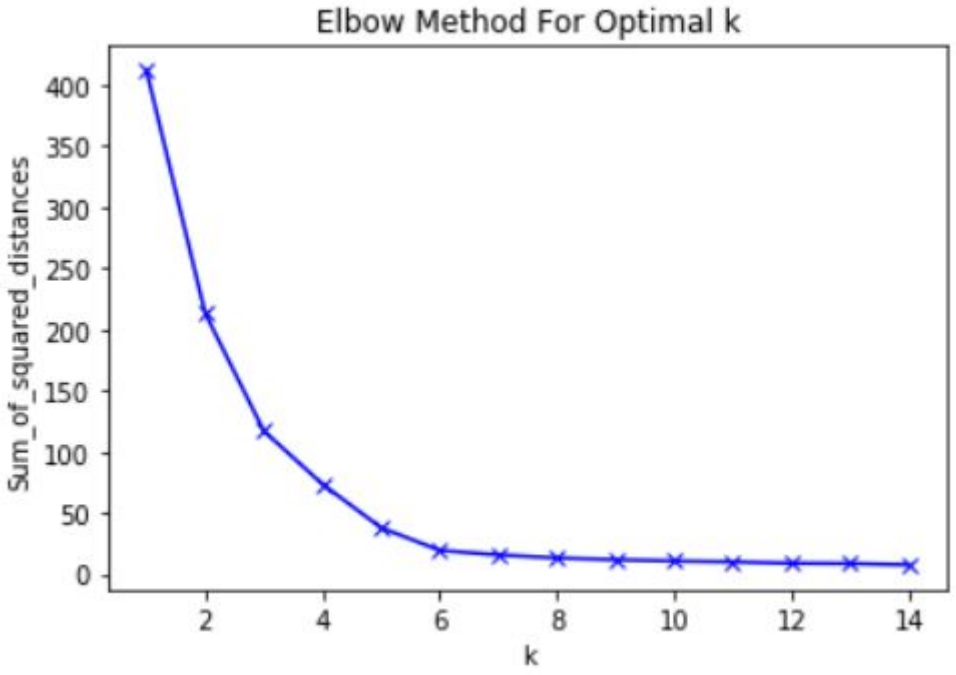
L'équation donnée est l'équation de l'erreur quadratique (Mean Squared Error). En choisissant la valeur de c telle que l'erreur soit minimisée, on sait que c est bien le barycentre où c est le moins écarté de tous les points.

On peut appliquer la dérivée à cette fonction d'erreur et trouver la valeur de C où la dérivée est égale à 0.

---

10. En pratique, comment choisir le nombre de clusters “idéal” ?

On augmente le nombre de clusters et on trace les “inertias” (les sommes des distances puissance de 2). On cherche le point où une des inerties commence à diminuer rapidement. C’est le nombre idéal de clusters (Elbow Method).



Par exemple, dans la figure ci-dessus, le nombre idéal de cluster est égale à 5.
(source: https://blog.cambridgespark.com/how-to-determine-the-optimal-number-of-clusters-for-k-means-clustering-14f27070048f)

---

11. Pourquoi l’analyse des éléments du dictionnaire doit se faire à travers des exemples de patchs et pas directement ?

C’est parce qu’on veut retenir les descripteur locaux. Une image peut contenir des images de maison, de voiture, de fenêtre, etc. On construit un dictionnaire visuel pour classer de nombreuses informations identifiables dans les images.

12. Commentez les résultats que vous aurez obtenus

En théorie, le nombre de clusters est parfait lorsque la relation ci-dessous est satisfaite :

$$\min_{c} \sum_{i} \| x_i - c \|_2^2 = 0$$

Mais cette condition est facile à satisfaire : il faut assigner le même nombre de clusters que les points x_i.
En réalité, cela peut signifier le surapprentissage. Donc, il faut trouver un compromis et l’obtention du nombre de cluster “idéal” est difficile.

De plus, on constate que l'algorithme de k-means est intensif au niveau de la mémoire et cela nous imposer d'utiliser le nombre plus petit des SIFT pour lancer k-means. Cela diminue alors la précision de l'apprentissage.

III. Utilisation de la technique du BoW

Cette dernière partie a pour objectif de créer une représentation numérique des images. Lors des étapes précédentes, nous avons obtenu des descripteurs locaux pour chaque image et un dictionnaire visuel des descripteurs les plus présents.
Maintenant on souhaite obtenir un descripteur global de chaque image à partir des descripteurs locaux connus.

Cet algorithme s'implémente en deux étapes : le coding et le pooling.
L'étape de coding consiste à coder chaque descripteur d'une image dans un nouveau vecteur par "projection" sur le dictionnaire visuel.
Le pooling permet ensuite à partir des descripteurs locaux d'obtenir un vecteur de description globale de l'image.

Il existe plusieurs pooling :

- Max pooling : Il diminue la dimension des données en utilisant que l'entrée maximum dans une région fixe de la couche convolutive.
- Sum pooling : Il considère la somme des entrées à la place du maximum.

L'algorithme est simple à implémenter et s'adapte aux changements d'échelle ou de rotation des images.

Questions - Partie 3

13. Finalement, que représente concrètement notre vecteur z pour une image ?

14. Montrez et discutez les résultats visuels obtenus

15. Quelle est l'intérêt du codage au plus proche voisin ? Quel(s) autre codage pourrait-on utiliser ?

Le plus proche voisin, Nearest Neighbors , il s'agit d'une méthode d'apprentissage non supervisé. Le principe des méthodes du plus proche voisin est de trouver un nombre prédéfini d'échantillons d'entraînement proches du nouveau point et d'en prévoir le libellé. (Source : https://scikit-learn.org/stable/modules/neighbors.html)

---

16. Quelle est l’intérêt du pooling somme ? Quel(s) autre pooling pourrait-on utiliser ?

L'intérêt du pooling somme est de prendre la valeur moyenne d’un patch sans pondérer et réduire la dimension des images. Le pooling somme est alors forcément proportionnel au pooling moyen, qui prend la moyenne pondérée. Il existe aussi le pooling max, qui prend la caractéristique la plus importante d’un patch.

---

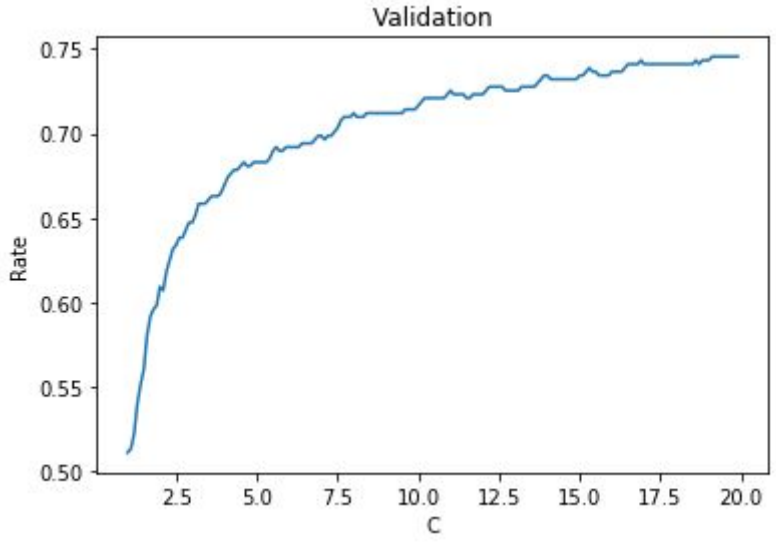
17. Quelle est l’intérêt de la normalisation L2 ? Pourrait-on utiliser une autre normalisation ?

L'intérêt de la normalisation L2 est de pénaliser la fonction de coût et enfin éviter le surapprentissage. Il existe aussi la normalisation L1.

IV. Classification d'images par SVM

Questions - Partie 4

1. Discutez les résultats obtenus, tracez une courbe de performance en fonction de C.




```
In [15]: C = 5.80
         clf = svm.SVC(C=C, gamma='scale')
         clf.fit(train_data, train_label)
         print(clf.score(test_data,test_label))

         0.688195991091314
```

```
In [16]: C = 7
         clf = svm.SVC(C=C, gamma='scale')
         clf.fit(train_data, train_label)
         print(clf.score(test_data,test_label))

         0.6937639198218263
```

```
In [17]: C = 9
         clf = svm.SVC(C=C, gamma='scale')
         clf.fit(train_data, train_label)
         print(clf.score(test_data,test_label))

         0.6937639198218263
```

```
In [18]: C = 19.9
         clf = svm.SVC(C=C, gamma='scale')
         clf.fit(train_data, train_label)
         print(clf.score(test_data,test_label))

         0.7360801781737194
```

On fait la validation croisée avec l'ensemble de tests mais on ne constate pas de surapprentissage.

2. Pourquoi l'ensemble de validation est-il nécessaire en plus de l'ensemble de test ?

On fait varier la valeur de C en fonction du taux de validation à partir de l'ensemble de validation. On risque donc de sur-apprendre les données d'entraînement et de validation. On fait un ensemble de tests indépendants qui sert purement à valider la classification.

---

3. Supposons que l'on souhaite utiliser ce classifieur sur de nouvelles images. À partir d'une image noir et blanc fournie, décrivez la chaîne de traitement qui permet d'obtenir la prédiction du type de scène de l'image.

De même, on va reconstruire les SIFTs de ces nouvelles images. Tout d'abord, on commence par appliquer le filtre de Sobel pour obtenir les contours dans les images. Ensuite, on discrétise les contours en 8 orientations et fait la somme des contours de même orientation.
Cela nous permet de construire un histogramme d'un patch. On le transforme en un tableau uni-dimensionnel. C'est le SIFT. On itère ce processus jusqu'à ce que l'on obtienne les SIFTs pour tous les patchs des images.

La prochaine étape est de classer les patchs de manière non supervisée en utilisant la méthode de K-Means. On garde le même nombre de clusters. A partir du résultat de K-Means, on reconstruit le dictionnaire visuel et puis le bag of word en utilisant hard-cording et pooling somme.

---

4. Comment pourrions nous améliorer notre chaîne de traitement complète ? (En particulier, le processus d'extraction de features (TME1-2) et celui de classification (TME3)

Pendant le processus de classification, le nombre de clusters n'est pas judicieusement choisi. Donc, on peut utiliser la "Elbow Méthode" pour mieux classifier.
