Valentyn Gerushta – 3703905
Léane Salais – 16003610
Rapport n°1 – 30 octobre

# rdfia

## sift & bag of words

Pour exploiter une base de données visuelles, il est tentant de commencer par une classification sémantique de ce qui s'y trouve. Or, quelle est la meilleure manière de représenter le contenu d'une image ? Les decripteurs SIFT (*Scale-Invariant Feature Transform*) sont de bons candidats.
La technique SIFT se fonde sur la caractérisation de zones d'intérêt à partir du gradient. Elle repère des variations locales fortes dans les éléments visuels, qui nous donnent une idée de leur structure – et par extension de leur nature.
Comme le suggère leur nom, les descripteurs SIFT sont invariants à un certain nombre de paramètres, ce qui assure une forte robustesse à tout modèle interprétatif dont ils seraient la base. Grâce à eux, on peut espérer trouver de fortes similarités entre des images qui contiennent les mêmes objets, et ce malgré des conditions de prise de vue différentes.
On peut extraire un dictionnaire général depuis les SIFTs calculés sur une base et y comparer ceux de chaque image pour en établir le contenu. Cet étiquetage serait ultérieurement utile pour découper une base en classes de similarité. L'objectif de ce rapport est de détailler le déroulement de cette technique et d'en démontrer les avantages et inconvénients.

## Descripteurs SIFT

La première étape vers la classification est donc de calculer les descripteurs liés à chaque image. On commence par quantifier son gradient horizontal et vertical à chaque pixel grâce à des filtres détecteurs de contours. Pour des raisons pratiques, on utilise des filtres séparables, ici des filtres de Sobel :

$$M_x = \frac{1}{4}\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix} \qquad M_y = \frac{1}{4}\begin{bmatrix}-1 & -2 & -1\\ 0 & 0 & 0\\ 1 & 2 & 1\end{bmatrix}$$

**1 Montrez que $M_x$ et $M_y$ sont séparables.**

Ils s'écrivent $M_x = h_y \cdot h_x^T$ et $M_y = h_x \cdot h_y^T$ où $h_x$ et $h_y$ sont des vecteurs colonne de taille 3 :

$$M_x = \frac{1}{4}\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix} = \frac{1}{2}\begin{bmatrix}1\\2\\1\end{bmatrix} \cdot \frac{1}{2}\begin{bmatrix}-1 & 0 & 1\end{bmatrix} = h_y \cdot h_x^T$$

$$M_y = \frac{1}{4}\begin{bmatrix}-1 & -2 & -1\\ 0 & 0 & 0\\ 1 & 2 & 1\end{bmatrix} = \frac{1}{2}\begin{bmatrix}-1\\0\\1\end{bmatrix} \cdot \frac{1}{2}\begin{bmatrix}1 & 2 & 1\end{bmatrix} = h_x \cdot h_y^T$$

donc $M_x$ et $M_y$ sont bien séparables, et s'écrivent tous les deux en fonction de $h_y = \frac{1}{2}\begin{bmatrix}1\\2\\1\end{bmatrix}$ et de $h_x = \frac{1}{2}\begin{bmatrix}-1\\0\\1\end{bmatrix}$.

**2 Quel est l'intérêt de séparer les filtres de convolution ?**

La séparabilité permet d'optimiser certains types d'implémentation en diminuant le nombre d'opérations lors du calcul. De fait, des multiplications successives (3+3 = 6 opérations par pixel) coûtent moins que des calculs matriciels (3*3 = 9 opérations par pixel).

L'image est ensuite divisée en patchs, avec une grille de pas fixé qui permet les recouvrements. A chacun sera associé un SIFT. Notons que la description d'une image sera plus précise mais aussi plus lourde à stocker si les patchs sont nombreux et se recouvrent ; on cherchera un compromis.

Une fois les éléments du gradient obtenus grâce aux filtres de Sobel, leurs composantes horizontales et verticales sont fusionnées et résumées via un calcul de norme et d'orientation. Les patchs ne sont donc pas prélevés tels quels sur l'image – au lieu des valeurs RGB, ils contiennent ces données de norme et d'orientation du gradient en chaque pixel. Ces informations vont subir de nouvelles modifications pour améliorer la robustesse de la représentation, nommément les suivantes :

**Quel est le rôle de la pondération par masque gaussien ?**

Extraites au niveau de chaque patch, les normes des gradients sont pondérées par une gaussienne. Pour chacun, elle diminue les valeurs les plus éloignées du point d'intérêt (le centre) jusqu'à les annuler. Ceci permet d'accorder moins d'importance aux bords d'un patch, qui correspondent aux centres de plusieurs autres dès lors qu'il y a recouvrement ; le calcul se concentre ainsi exclusivement sur des caractéristiques vraiment propres à chaque zone.

*A ne pas confondre avec un filtrage gaussien, qui aurait un effet lissant. Dans la technique SIFT commune, lisser aussi l'image avant de la décrire rend le système plus robuste. Des détails qui s'apparentent à du bruit disparaissent et ne laissent que des basses fréquences – des objets et contours marqués.*

**Expliquez le rôle de la discrétisation des orientations.**

Les orientations trouvées subissent elles aussi des transformations, en particulier une discrétisation qui les sépare en un nombre fini de classes. De ce fait, les contenus des patchs seront plus tard comparables classe à classe : ces classes d'orientation auront divers degrés de présence dans chacun, et on étudiera facilement leur distribution.

Un modèle ainsi traité est aussi plus robuste aux rotations. Tant que l'angle d'une rotation (appliquée ou inhérente à l'image) ne dépasse pas une certaine valeur, le résultat reste le même que pour l'original. Des images similaires auront donc des orientations de gradient semblables – ce qui est voulu – là où une description au degré près n'aurait pas permis pas de les rapprocher. La tolérance sera logiquement plus grande si les classes sont peu nombreuses. Selon le contexte, on cherchera un équilibre entre la précision des orientations mémorisées et le degré de robustesse du modèle.

Tout est maintenant prêt pour le calcul des descripteurs SIFT. Les patchs sont divisés en sous-patchs, sur lesquels il suffit de calculer l'histogramme des orientations. Le SIFT final sera fait d'une juxtaposition de ces histogrammes, moyennant un dernier faisceau de transformations :

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

Le seuillage et la normalisation de chaque descripteur – en deux temps – assurent sa stabilité face à la *vitesse* des variations de contraste. Le descripteur ne doit plus réagir au bruit seul (1). Sa valeur ne doit plus dépendre non plus du degré de netteté de l'image (on rappelle que le gradient d'un contour a une norme plus ou moins grande selon son acuité…) puisqu'on seuille tout à une valeur limite (2).

(1) On commence par annuler les descripteurs dont la norme euclidienne totale est inférieure à un certain seuil : les écarts entre leurs valeurs sont trop faibles. Interprétativement, ils correspondent à des zones unies plus ou moins bruitées mais inintéressantes par défaut de contraste. On normalise les autres.

(2) Un seuillage et une seconde normalisation désensibilisent ensuite les descripteurs aux changements extrêmes. On impose une valeur maximale pour chaque classe, le descripteur sera donc le même quoi qu'on la dépasse. De cette manière, on ne conserve que l'information « moyenne », à savoir qu'*il y a* un changement très significatif du gradient en tel point – indépendamment de sa valeur exacte.

Le descripteur à ce stade reflète donc la distribution des orientations du gradient dans chaque patch, et ce de façon assez indépendante de la rotation, des translations, du bruit et du degré de netteté.

Ajoutons qu'il était déjà invariant à la luminosité globale, car le fait de prendre un gradient comme outil de travail invisibilise tout changement additif (et uniforme).

**6 Expliquez en quoi le principe du SIFT est une façon raisonnable de décrire numériquement un patch d'image pour faire de l'analyse.**

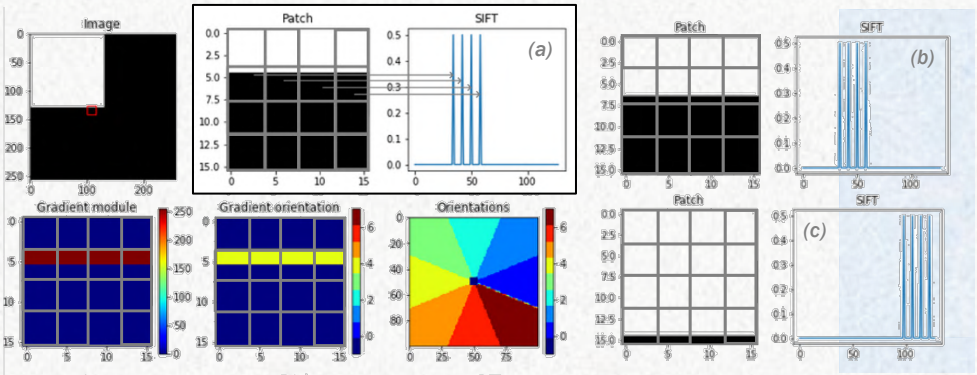
Pour ce qui est de « décrire », le modèle SIFT permet de caractériser le contenu visuel d'une image, le plus indépendamment possible du cadrage, du bruit et de l'exposition. Nous avons justifié sa robustesse dans les questions précédentes. Il est donc un indicateur fiable des éléments qui y sont mis en scène, en cela qu'il représente particulièrement bien leur *structure*.

Pour ce qui est du « raisonnable », on se rend compte que la représentation d'une base à l'aide de SIFTs peut être considérablement réduite en termes de taille. Le fait de travailler en valeurs de gradient fait que les vecteurs mémorisés sont des condensés d'information à propos des images. Seules les zones suffisamment variables (donc représentatives) comptent, ce qui donne des résumés plus *sparse*.
Cette façon de faire floute la forme et la répartition spatiale des objets au sein des sous-patchs. Cela dit, le but étant l'analyse et non la reconstruction, ce qu'on garde est suffisant et ce choix se justifie pleinement.

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

On souhaite expliquer la lecture des descripteurs SIFT, et montrer comme ils varient en fonction des conditions de prise de vue et des paramètres de découpage.

Sur un descripteur, la taille des pics de l'histogramme dépend du module du gradient (sa « quantité ») lié à chaque classe d'orientation, et ce dans chaque sous-patch. Ces pics sont donc d'aspect similaire et régulier lorsque ce module et cette orientation sont constants – comme dans la Figure 1. En ajoutant une grille de visualisation, on constate bien (encadré noir) la correspondance entre entre eux. En (b) et (c), on a également tenté de montrer la réaction du descripteur aux translations resp. légères ou plus marquées :




Il est invariant aux translations légères tant qu'elles ne dépassent pas la taille d'un sous-patch. On aurait pu également montrer sa résistance aux variations de lumière ou aux rotations.

*Figure 1 – Comparaison entre des SIFTs basiques. Impact d'un déplacement léger (b) ou plus net (c) du patch.*

Dernière chose : du fait du seuillage de l'histogramme, l'effet du masque gaussien (Q3) ne paraît que sur des pics intermédiaires (ci-contre). La réponse est moins forte lorsque le masque est appliqué. Mais ces images sont trop simples pour en apprécier pleinement l'effet.

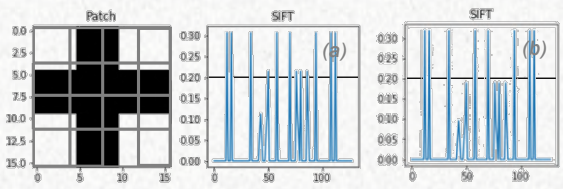



*Figure 2 - Impact du masque gaussien : sans (a) et avec (b).*

# Dictionnaire visuel

Une fois les descripteurs SIFT locaux extraits de chaque image de la base, ils sont regroupés dans K classes de similarité grâce à un algorithme de clustering (comme K-means). On génère ainsi un dictionnaire de mots visuels de taille arbitraire qui sera notre seule mémoire des étapes précédentes.

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

Même si nous avons discrétisé les orientations de gradient au cœur des patchs, il reste quand même de nombreuses possibilités de combinaison par patch. Le dictionnaire nous permet de créer un vocabulaire réduit, de taille fixe, qui résume ces possibilités. Cette réduction est très utile pour des questions de dimension et de complexité des calculs ultérieurs ; mais montrons que ce vocabulaire réduit est adéquat.

Le cœur (centroïde) de chaque cluster de similarité représente maintenant un *type* de pattern appris à partir de la base, et ce sont ces abstractions des patterns récurrents qui formeront le vocabulaire. – Nous sommes certains qu'ils sont *récurrents* du fait de la nature du processus d'apprentissage : les patterns très courants se présentent en grand nombre, donc avec davantage de poids. Ils auront donc tendance à influencer fortement les abstractions et à servir de points d'ancrage pour les K mots du dictionnaire.

Dans une étape ultérieure, chaque image sera décrite par un histogramme à K bins. Les mots visuels de l'image seront mis en correspondance avec le dictionnaire selon leur distance aux centroïdes, classés selon le mot connu le plus proche, puis comptés par classe. A terme, ce décompte sera interprété pour classer l'image à partir de ses parties, en reconnaissant que tel ou tel mot connu y est majoritaire.
Le dictionnaire visuel aura donc joué un rôle central : en listant les éléments bien connus sur la base, il aura permis une indexation efficace des descripteurs locaux pertinents dans les images.

**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 $x_i$.**

La dispersion de points x autour d'un centre c est une forme d'erreur. Un algorithme de clustering annule le gradient de cette erreur. Minimiser la dispersion, c'est donc apprendre une moyenne :

$$\hat{c} = \underset{c}{\operatorname{argmin}} \sum_{i}^{n} ||x_i - c||_2^2 \quad \equiv \quad -2 \sum_{i}^{n} \left|\left|x_i - \hat{c}\right|\right| = 0 \quad \equiv \quad \hat{c} = \frac{1}{n} \sum_{i}^{n} x_i$$

où les $x_i$ sont les n éléments du cluster.

Pour construire le dictionnaire, le nombre de clusters K peut être choisi à l'envi. Il existe cependant un compromis entre la précision des mots visuels appris (qui servent de centre à des classes d'équivalence plus ou moins étendues entre les SIFTs) et la taille du dictionnaire à mémoriser. Un grand dictionnaire prend de la place, est plus difficile à manier, mais fait peu d'amalgame entre les descripteurs SIFT : on multiplie les cœurs de clusters et donc la finesse des motifs reconnaissables. Un petit dictionnaire permet de gagner de la place, mais donne des mesures de similarité moins fines en réunissant des descripteurs SIFT vaguement apparentés sous le même centroïde. Comme pour la discrétisation des orientations à la Q4, le choix dépendra là encore du contexte d'utilisation.

**10 En pratique, comment choisir le nombre de clusters "idéal" ?**

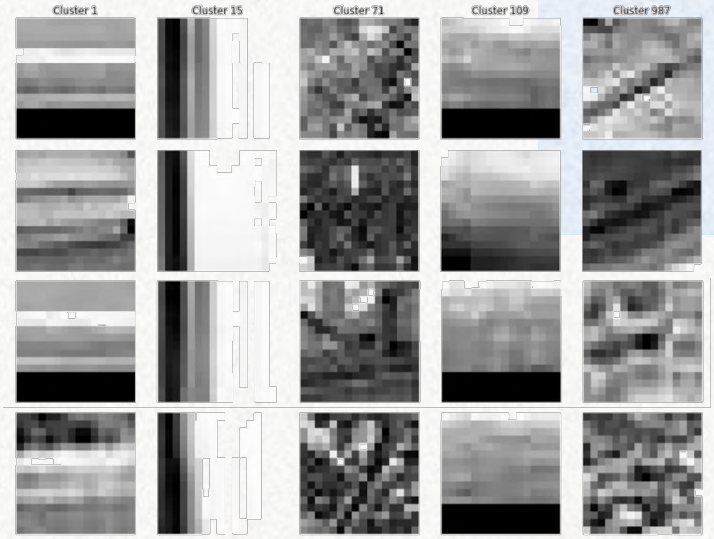
Pour un même jeu de données, il n'existe pas un unique clustering possible et aucun procédé infaillible pour trouver le bon. Si le nombre de mots K du dictionnaire n'est pas choisi à l'avance, la méthode usuelle est de lancer K-Means avec des valeurs croissantes de K et de vérifier la variance et la taille des clusters – c'est un GridSearch. On recherche à la fois une variance intraclasses faible et une taille raisonnable pour toutes (un cluster trop peu peuplé est inutile pour établir une typologie de la base).

Cependant, il existe comme on l'a dit un compromis entre la précision du résultat et le coût computationnel. Générer et manier un plus grand nombre de clusters prend du temps. En pratique, le meilleur K est celui pour lequel la variance intraclasses ne se réduit plus *significativement* ; on sait qu'elle décroît de plus en plus lentement avec K, et on se satisfera d'un découpage légèrement sous-optimal afin de ne pas trop alourdir nos calculs.

**Pourquoi l'analyse du dictionnaire doit-elle se faire à travers des exemples, et non directement ?**

Les éléments du dictionnaire n'existent pas dans les images. Ils sont une abstraction. Ils représentent une moyenne d'un grand nombre d'objets qui se ressemblent, et donc quelque chose de flou. Pour l'interprétation humaine, on préfère donc afficher les patchs dont les représentations sont les plus proches des centroïdes (suivant la logique de Q9, ceux qui y pèsent le plus puisqu'ils ont déjà largement servi à les créer) pour obtenir une visualisation plus claire.

**Commentez les résultats que vous aurez obtenus.**




Ci-contre, nous affichons quatre patchs à distance croissante (au 1er, 2nd, 5ème et 15ème degré) du centre de cinq clusters. Ces résultats montrent que le dictionnaire contient des clusters homogènes, avec des signes distinctifs évidents et plutôt faciles à interpréter. Les cinq affichés ici présentent des caractéristiques diverses :
1) lignes horizontales claires,
15) bloc vertical sombre à gauche,
71) texture rugueuse (végétale ?) sombre,
109) bloc horizontal sombre en bas,
987) structure vaguement diagonale…
Le résultat est différent mais toujours parlant à chaque exécution.

Parler de zones *claires* et *sombres* est inadéquat, puisque les descripteurs sont résilients aux variations de luminosité (à la suite des calculs et normalisations effectués). Cependant, le contraste général demeure, et il est facile d'exprimer ainsi nos observations. A vue de nez, ce dictionnaire nous semble satisfaisant, il permet bien de caractériser des structures diverses sous des archétypes bien définis.

# Bag-of-Words (BoW)

A l'issue de la première phase, chaque image était representée par l'ensemble de ses SIFTs. Il est maintenant possible de ramener cette représentation à un ensemble de mots du dictionnaire. Cette logique correspond exactement à du traitement du langage : connaissant les éléments de sens, voici leur similarité aux composantes du corpus ; quels sont celles qui les décrivent au mieux ?..
Notons qu'en passant au modèle sac-de-mots, toute information spatiale est définitivement perdue. Il n'y a aucun moyen de réarranger les mots listés pour récupérer l'image originale.

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

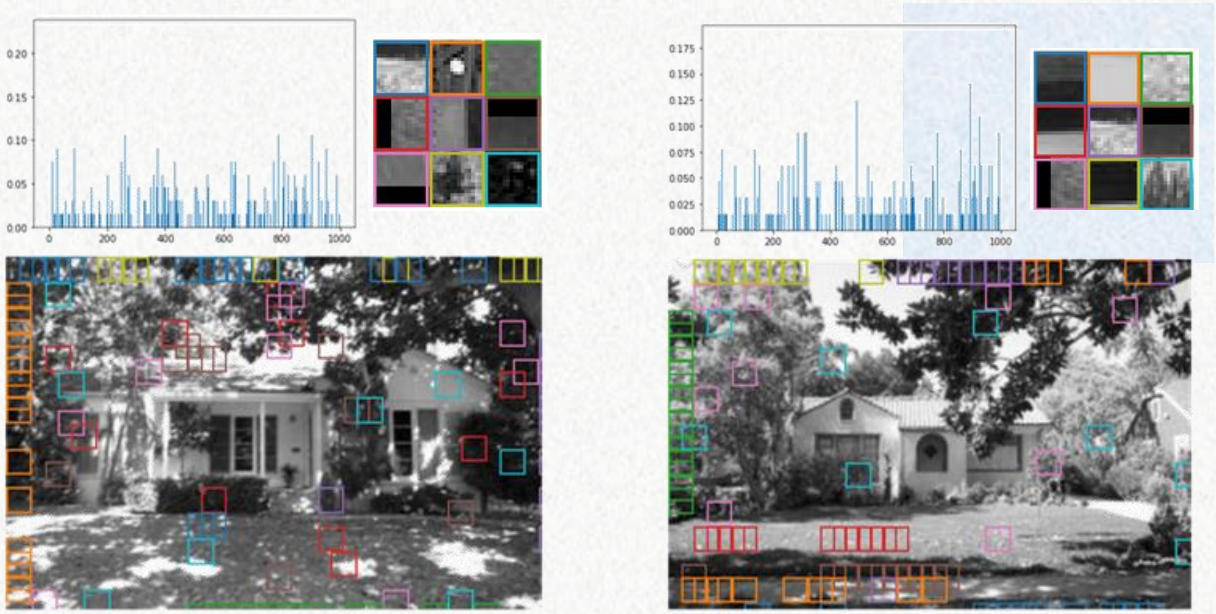
Le vecteur z est un tableau de comptage qui donne le nombre d'apparitions des 1001 mots du dictionnaire visuel dans une image I. C'est une représentation fréquentielle complète de l'image. Sous la forme d'un histogramme non-normalisé, elle a effacé l'information spatiale ; cela dit, l'objectif étant la comparaison de contenu, cela n'a aucune importance.

[Parler d'« apparitions » est incorrect. Comme expliqué plus haut, aucun des mots du dictionnaire n'existe dans les images (dans la mesure où il s'agit de centres de clusters). Il s'agit donc plutôt d'un comptage des patchs qu'on a classifiés comme étant *proches* de l'un ou de l'autre des éléments du vocabulaire.]

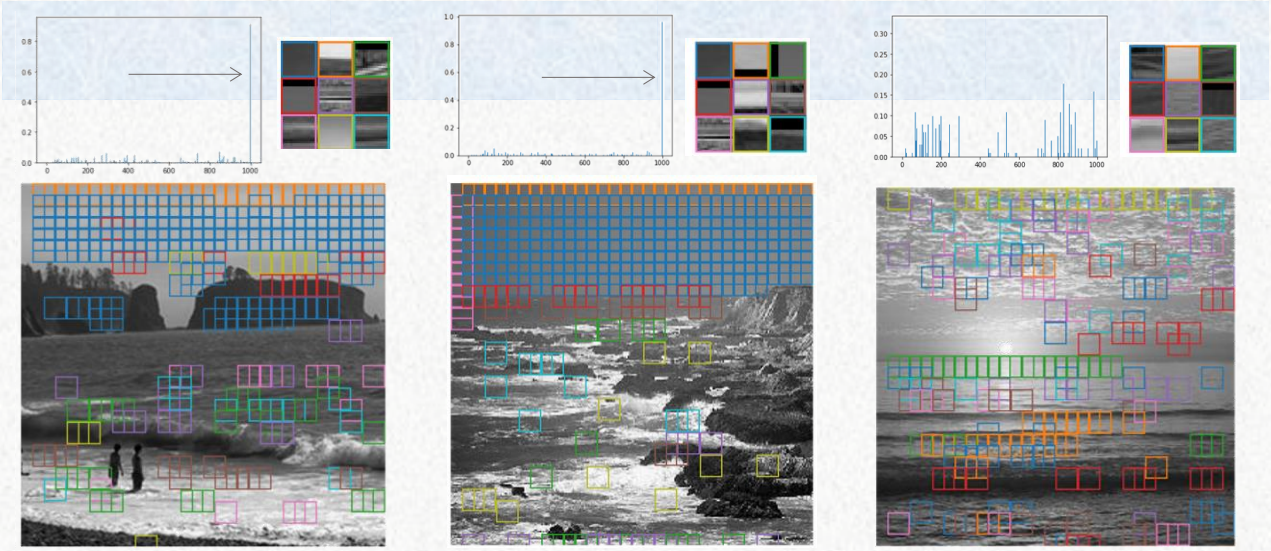
Dans cet histogramme, certains mots du vocabulaire peuvent ne pas apparaître du tout, d'autres très souvent. Sa dimension reste cependant la même dans tous les cas. On interprétera la taille de ses pics (ou leur absence) après normalisation pour qualifier les éléments représentés dans l'image.

**14 Montrez et discutez les résultats visuels obtenus.**

L'interprétativité du modèle depend de la taille des patchs conservés. Nous avons utilisé un découpage 16*16, ce qui reste trop petit pour reconnaître des objets entiers et déployer une sémantique à l'échelle humaine. Cependant, certains éléments restent commentables : voici quelques images, avec pour chacune les histogrammes BoW associés, les SIFTs courants repérés, et une légende parcellaire censée les résumer.



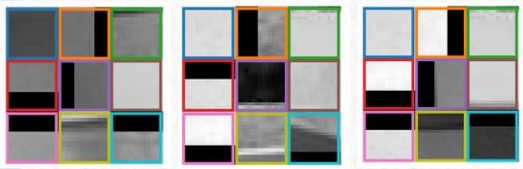
Sur ces images qui se ressemblent, les éléments sont variés, avec des bâtiments et des éléments naturels. Les patchs de la légende le sont donc aussi. Appuyant l'uniformité de l'histogramme, où toutes les classes sont représentées, les étiquettes colorées des patchs sont d'ailleurs bien réparties dans l'image : aucun mot visuel ne dépasse largement les autres en fréquence. Mais voyons maintenant un autre cas :



Sur ces paysages essentiellement composés d'eau, les textures à dominante horizontale sont les plus communes et ressortent bien dans les trois légendes résumées.
Notons aussi que quand le ciel est dégagé, les zones unies sont majoritaires, comme l'indique le pic à l'extrême de l'histogramme. Il est lié au mot artificiel n°1001 que nous avions ajouté au dictionnaire exprès.

Ce résultat est satisfaisant. La caractérisation réussie des zones unies et des objets corrobore la notion selon laquelle la représentation BoW peut servir de résumé sémantique pour toutes les images.

Cependant, nous avons une critique inattendue à faire au sujet de la qualité des patchs recueillis. On note une affinité systématique avec les bordures d'image. Les patchs affichés dans les légendes comportent souvent de larges zones noires : il s'agit du padding... Il suffit que les bords de l'image soient initialement unis pour faire du padding un mot visuel largement majoritaire. Pour éviter cette focalisation inutile, qui nous fait perdre un grand potentiel de description, il pouvait être plus intéressant d'utiliser un padding « miroir ».



*Apparitions exagérées du padding pour les BoW de plusieurs images.*

*Indépendamment de la question, nous avons une autre remarque. Les couleurs qui encadrent les patchs de la légende ne correspondent pas à celles qui sont les leurs dans les images. Nous l'avons signalé, sommes à peu près certains de notre code, et savons que cet effet était récurrent à travers tous les groupes de TME.*

Notre version de Bag of Words fonctionne : codage des SIFTs par leur plus proche voisin dans le dictionnaire, obtention des occurrences totales de ces mots, puis normalisation L2 du comptage. Mais elle admet des variantes.

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

Le codage au plus proche voisin permet de classer les patchs d'une image en les assimilant à un et un seul mot du dictionnaire visuel – le plus proche au sens de la norme. Cette méthode est simple, intuitive, rapide à mettre en place. Elle est aussi robuste à plusieurs transformations et décalages tant que le même mot visuel reste le voisin privilégié du patch : c'est une forme de one-hot-encoding.

Un soft-assignment est aussi envisageable, où la proximité à chaque mot du dictionnaire est quantifiée par un pourcentage. Cela permettrait d'améliorer la descriptivité de l'ensemble, car certains patchs sont un mélange quasi-uniforme de plusieurs mots ; mais aussi stocker davantage d'informations, alors que le codage utilisé ici est déjà suffisamment informatif pour la plupart des cas.

16 **Quel est l'intérêt du pooling somme ? Quel autre pooling pourrait-on utiliser ?**

Le pooling somme prend en compte le nombre d'occurrences d'un mot visuel dans une image. Il permet de créer un tableau de comptage, et donc de bien recueillir tous les mots visuels présents dans l'image tout en identifiant les mots majoritaires.

Utiliser un pooling somme, c'est donc s'orienter vers un calcul de fréquence (après normalisation). C'est déjà mieux qu'un pooling binaire par « présence ». Mais il serait encore plus informatif d'utiliser un pooling de type tf-idf (en suivant la métaphore d'un document) qui associe, à chaque mot visuel m d'une image i :

$$w_{m,i} = tf_{m,i} \log\left(\frac{N}{df_m}\right)$$

où $tf_{m,i}$ est le nombre d'occurrences du mot m dans l'image i (résultat du pooling somme), N le nombre total d'images dans la base, et $df_m$ le nombre d'images parmi elles qui contiennent ce mot.

L'usage d'un score tf-idf réduirait notre problème de dominance du padding : dans un tel codage, les scores associés aux mots omniprésents sur la base sont pénalisés. Les mots de score maximal seraient donc très présents dans l'image, mais aussi *spécifiques* à cette image ou à un sous-groupe d'images.

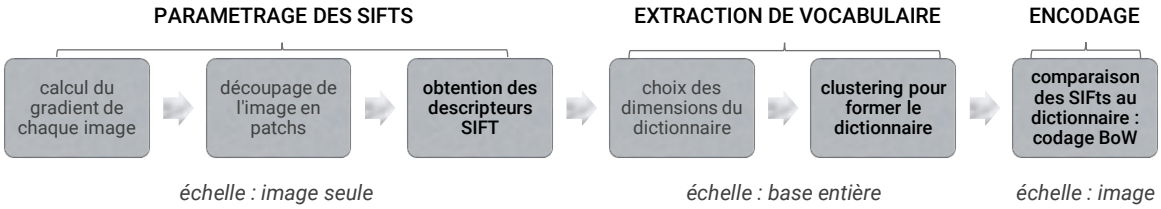
**17 Quel est l'intérêt de la normalisation L2 ? Pourrait-on utiliser une autre normalisation ?**

La normalisation du tableau de comptage permet d'obtenir une distribution des « apparitions » des mots. Elle est nécessaire pour mettre en place des systèmes de comparaison qui ne dépendent pas de la taille de chaque image (et donc du nombre total des mots qui s'y trouvent).

Pourquoi L2 plutôt que L1, ou autre chose ? Elle est le seul choix possible ici.
La normalisation L1 utilise une valeur absolue avec une grande variabilité des résultats. La normalisation L2 renvoie un vecteur dont la norme est toujours 1, avec néanmoins une certaine *répartition* des valeurs à l'intérieur. Or, nous cherchons bien à établir une *répartition* des mots du dictionnaire par image, et nous en voulons une représentation comparable et non-biaisée pour toutes. Nous avons donc *besoin* de l'aspect relatif qu'apporte L2 et de cette norme de 1 qui permet une visualisation *fréquentielle*.

## Conclusions et autres considérations

Nous avons bien trouvé un moyen d'encoder des images de taille variable via leurs SIFTs. Nous avons réduit et uniformisé les dimensions d'entrée, créé des BoWs, pour utiliser des classifieurs de type SVM sur ces images. Tout s'est fait selon le protocole suivant :




Il ne reste plus qu'à collecter ces BoWs et à laisser une SVM les organiser. Nous avons tenté cette classification au TME3, avec des pré-résultats satisfaisants.

Comment pourrions-nous améliorer notre protocole ? Nous avons signalé le problème du padding qui est omniprésent dans les représentations SIFT des images, et le complément possible au niveau du pooling dans le BoW : utiliser un pooling tf-idf au lieu de la simple fréquence des mots visuels.

Un autre élément attire notre attention. Il est parfois utile de détecter les zones unies (par exemple pour repérer des paysages), parfois non (notamment si les objets de la base se découpent sur un fond). Dans ce dernier cas, deux options s'offrent à nous : économiser le mot artificiel, ici le n°1001, qui y était dédié, ou bien détecter les zones d'intérêt avant de lancer le calcul des SIFTs exclusivement dessus. La deuxième option offre d'autres avantages, comme celui de proposer potentiellement des SIFTs plus interprétables – car mieux centrés sur les points d'intérêt.

*Quelques sources :*

https://www.kaggle.com/residentmario/l1-norms-versus-l2-norms
https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/
http://weitz.de/sift/