Rapport BOF et BOW

Ismat Benotsmane

October 2020

1 Introduction

Le but de ces travaux pratiques et de réaliser une série de tache essentielle pour préparer les données aux taches de classification.
On va dans un premier temps construire des descripteurs locaux en utilisant l'algorithme **Scale-Invariant Feature Transform(SIFT)**.
On construira ensuite un dictionnaire visuel qui representera les descripteurs-types des motifs les plus fréquents dans nos images. Avec ce dictionnaire là, on pourra construire un descripteur d'image avec la technique de **BoW (Bag of Words)**.

2 SIFT

2.1 Calcul du gradient d'une image

(Q1) On a les masques M_x et M_y :

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

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

Nous cherchons à montrer que ces 2 masques sont séparables, c'est-à-dire qu'ils peuvent s'écrirent comme : $M_x = h_y h_x^T$ et $M_y = h_x h_y^T$.
On doit donc résoudre le systéme d'équation :

— $M_x = (y_1, y_2, y_3).(x_1, x_2, x_3)^T$
— $M_y = (x_1, x_2, x_3).(y_1, y_2, y_3)^T$

On obtient alors :

$M_x = \frac{1}{4}(1, 2, 1).(-1, 0, 1)^T$
$M_y = \frac{1}{4}(-1, 0, 1).(1, 2, 1)^T$

On a donc :

$$h_x = \frac{1}{4}(-1, 0, 1)$$

$$h_y = \frac{1}{4}(1, 2, 1)$$

(Q2) L'interet de separer le filtre de convolution est tout simplement pour des raisons de rapidité de calcul, car il est plus simple et plus rapide de multiplier par des vecteurs que par une matrice.
Dans notre cas, si on a avait multiplier par la matrice M_x on aurait fait $3x3$ opérations. Mais avec la séparation, on ne fait plus que $3 + 3$ opérations.

2.2 Calcul de la représentation SIFT d'un patch

(Q3) L'application du masque gaussien à nos données permet de données plus d'importance au pixel proche du centre de la distribution, et par consécant d'éviter d'avoir des patches bruités par certains pixels. Cela permettra de générer des descripteurs sifts plus robuste.

(Q4) Tout d'abord on discrétise les orientations pour des raisons computationnelles, car il est impossible de créer des sifts de toute les directions possibles. Et meme si on considére par exemple chaque degré comme une orientation et qu'on discrétise là-dessus, pour un patch on aura alors un sift de taille $16 \times 360 = 5760$

L'autre raison, c'est d'avoir des sifts robuste à la rotation. C'est à dire que si on applique une rotation qui reste dans le même pas de discrétisation sur une image, son descripteur sift restera inchanger.

(Q5) Lors du post-processing appliqué au SIFT, on renvoie un descripteur nul si la norme-2 du SIFT en question est inferieur à 0.5 car on considére qu'il n'indique aucune information intéressante puisqu'il concerne une zone où il y a peu de changement.
De ce fait, on gagnera du temps à ne pas consider ces SIFT's.

La normalisation et le passage des valeurs superieurs à 0.5 à 0.2 permet de rendre le descripteur insensible à la luminosité.

(Q6) Le principe du SIFT est une façon raisonnable pour décrire un patch d'image, car ça permet de traiter chaque pixel du patch en fonction des pixels voisinants. En calculant le gradient par exemple, cela nous permettra d'evaluer si ce patch est interessant ou s'il concerne une zone neutre avec aucun changement. Cela permet également d'avoir un descripteur tolérant face à une rotation et moins sensible à la luminosité.

On aurait pu faire une classification en considérant uniquement pixel par pixel mais dans ce cas-ci on aurait classifier l'image en fonction de la couleur de ses pixels.

(Q7) Notre patch est découpé en 16 region de taille 4×4. Les 4 premières

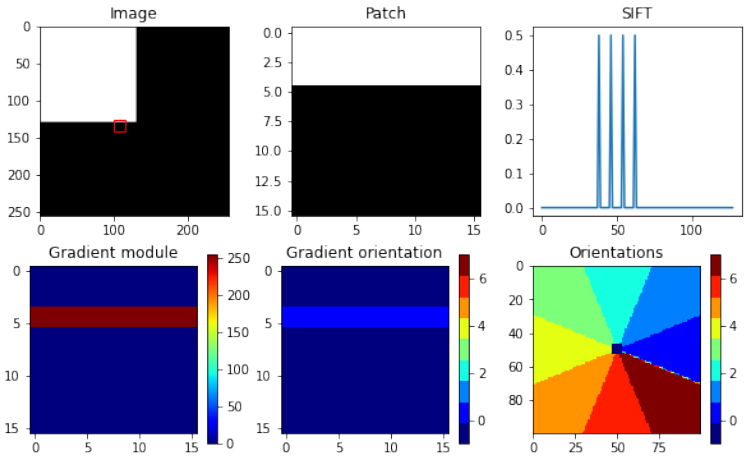

FIGURE 1 – Descripteur SIFT d'un patch

regions se situe dans une zone blanche où il y a rien d'intéressent.
Les 4 regions suivante se retrouve sur un bord, donc fort changement. Ce qu'on retrouve bien sur le SIFT à partir du 4×8éme élément du graph où l'on reçoit

un signal par region.
La norme de gradient est sans surprise élévé sur la zone où le bord se situe du fait de la valeur des gradients des pixels qui sont élevé.
Quant à l'orientation du gradient, elle suit le sens du bord comme l'indique le graph *Gradient orientation*.

3 Dictionnaire visuel

(Q8) Chaque mot du dictionnaire est un centroide du kmeans qu'on a appliqué sur l'ensemble des SIFT de notre base de données. On a choisi arbitrairement 1001 clusters ce qui nous fait un dictionnaire de 1001 mots.
Les mots du dictionnaire représente donc chacun un SIFT-type qui fera référence à un pattern fréquent qui a été observer dans nos images.

Le but est de se servir de ce dictionnaire pour comprendre une image, c'est-à-dire savoir quels patterns la compose et combien. Cette étape est essentielle pour le processus de la classification. Et pour faire ça, on va decomposer notre image en un ensemble de SIFT, et pour chacun d'eux on va mesurer la distance au SIFT-type du dictionnaire, et on compte ces SIFT-type proches.

(Q9)

$$\min_c f = \sum_{i=1}^{n} ||x_i - c||^2$$

Pour trouver le c optimal :

$$c^* = \frac{\delta f}{\delta c} = 0 \iff \sum_{i=1}^{n} (2c - 2x_i) = 0$$

$$c^* = \frac{1}{n} \sum_{i=1}^{n} x_i$$

(Q10) Dans notre cas, on a fixé le nombre de cluster à 1000 mais evidemment il se peut que ce nombre ne soit pas optimal pour le dataset qu'on posséde.
Pour trouver le nombre de clusters optimaux, on utilise la méthode du Elbow qui consiste à mesurer la distortion en fonction du nombre de clusters. Le but est de choisir le nombre de cluster à partir duquel la variation de la distortion ne se réduit plus de façon significative.
La courbe a une forme de coude, le nombre de clusteurs optimaux se situe donc au niveau du coude.

(Q11) Les éléments du dictionnaire représente des SIFT moyen obtenu par clustering, ils sont donc pas interprétables visuellement si on les affiche. On ne peut pas également reconstruire fidélement un patch à partir d'un SIFT.
Il est donc interessant d'utiliser des patches existant pour exprimer un élément du dictionnaire en trouvant le patch qui lui est proche.

(Q12) Aprés avoir calculer notre dictionnaire en appliquant un kmeans avec 1000 clusters en ajoutant un cluster pour les descripteurs nuls. On a séléctionné aleatoirement 30 images, pour lesquelles on a calculer leur SIFT.
On a pu ensuite interpréter notre dictionnaire en s'appuyant sur les SIFT calculer précédemment pour trouver le celui le plus proche de chaque cluster.
Dans les clusters, on a pu remarquer une fenetre ou encore une roue.

Plus on va avoir un sample d'image diversifié en terme d'image(trés differentes les une des autres) plus notre dictionnaire sera exprimer par des patches trés éloignés.

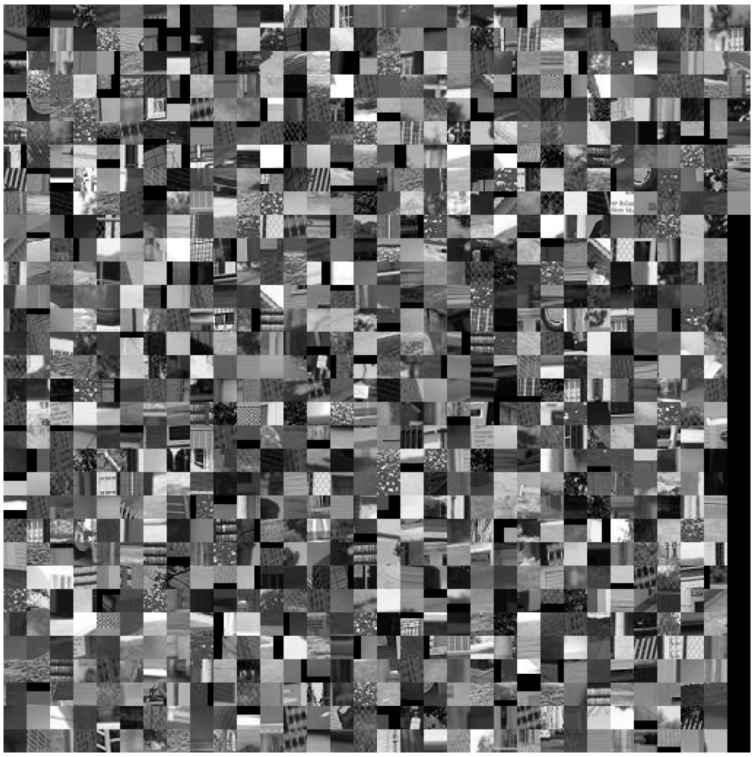

FIGURE 2 – Interprétation du dictionnaire par des patches

4 Bag of Words

(Q13) Le descripteur BOW permet de définir un vecteur z qui va caractérisé une image. Ce vecteur sera la représentation globale d'une image contrairement au SIFT qui eux été une representation locale(région) d'une image.

Cette représentation permet d'interpréter une image et ainsi pouvoir effectuer d'autres taches derriere tel que de la classification.

(Q14) La figure 3 est le vecteur z qui indique la proportion de chaque mot du dictionnaire dans l'image. Ce vecteur là sera le descripteur de notre image. Comme on l'observe sur cette figure, 90% des régions de cette image correspondent au cluster nul. Il fait référence au cluser des régions dans lesquelles la variation est trés faible.

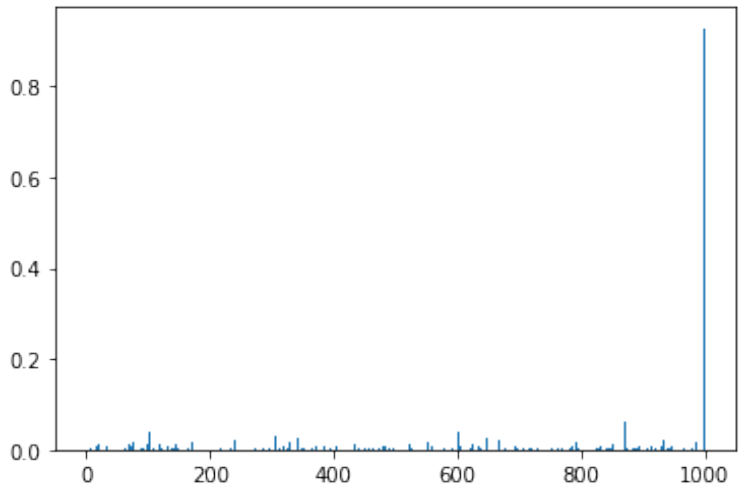

FIGURE 3 – Vecteur z representant l'image 4

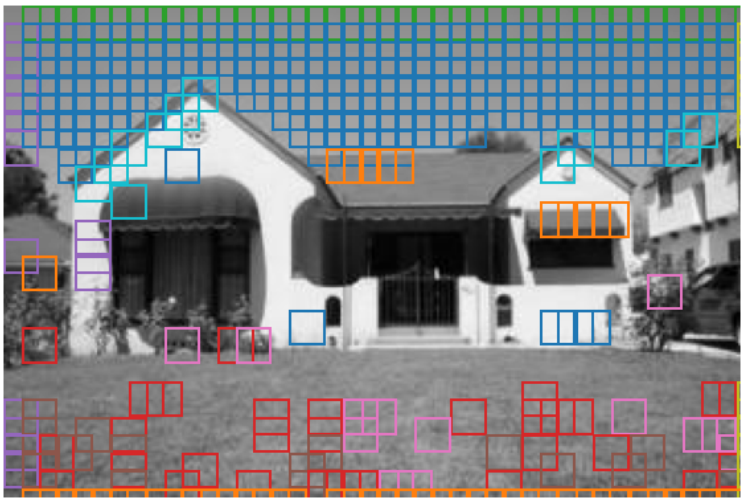

FIGURE 4 – L'image 4 d'une maison

La figure 4 est une image d'une maison avec les patches des 9 mots les plus important qui l'a décrive.
On remarque que les patches de couleur bleu, sont en nombre les plus important. De plus, il couvre une région(ciel clair) où il n'y a pas de variation ce qui concorde avec la figure 3.
On peut également relever que les les patches bleu clair suivent les bords du toit en diagonale.

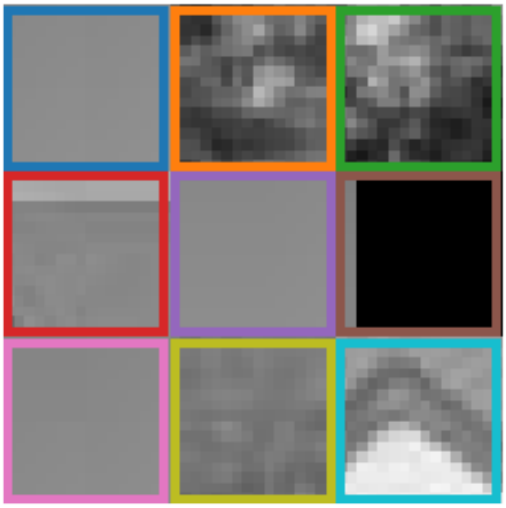

FIGURE 5 – Prototype des mots

La figure 5 représente le prototype des 9 mots les plus présents dans l'image.

(Q15) L'interet du codage plus proche voisin est d'affecter un patch seulement à un seul cluster. Pour cela on utilise un encodage one-hot.
Le but ici est d'associer un seul mot du dictionnaire au patch. C'est ce qu'on appelle du "hard assignement".

On pourrait utiliser une methode de "soft assignement", où un patch est définie par les probabilités d'appartenance à chaque mot du dictionnaire. On pourrait utilisé la fonction softmax pour cela.

(Q16)
Le pooling somme est le fait de compter pour chaque cluster le nombre de regions de l'image qui lui sont affectées parce qu'elles sont considérées proche.
Il existe également d'autre pooling comme le max pooling qui consiste à prendre uniquement le cluster qui apparait le plus de fois dans une image.

(Q17) La normalisation L2 permet d'avoir un vecteur z standardisé pour n'importe qu'elle image, où sa somme est égale à 1.
Cette normalisation permet aussi de comparer deux images entre-elle en analysant l'importance de chaque cluster indépendemment de leur de patch/SIFT, car une image avec un nombre de patch élevé ne sera comparé de la meme maniere qu'une image avec un nombre de SIFT faible.