

UNIVERSITÉ NATIONALE DU VIETNAM À HANOÏ
INSTITUT DE LA FRANCOPHONIE POUR
L'INNOVATION

Option : Systèmes Intelligents et Multimédia (SIM)

Promotion : 23

Année académique : 2019-2020

Module : VISION PAR ORDINATEUR

**Reconnaissance d'objets avec le
descripteur SIFT**

TP1

Présenté par
SOUMANA HAMADOU ABDOURAHMANE
OUEDRAOGO ABDOUL-FATAO
Email :abdoulsoumana1201@gmail.com

Prof :Dr NGUYEN Thi Oanh

0.1 Introduction

De nos jours avec l'avancé de l'internet et des technologies modernes, notre société est devenue un monde d'image. En effet aujourd'hui n'importe qui peut utiliser l'appareil photo de son smartphone pour prendre une photo ou une vidéo et la partager sur le web ou sur les réseaux sociaux. Toutes ces images qui inondent internet regorgent des données qui pourraient être précieuses pour les entreprises. Cependant, pour pouvoir collecter, analyser, traiter et comprendre ces données, il est nécessaire que les ordinateurs soient en mesure de "voir" une image et de comprendre son contenu. Tel est l'objectif de la vision par ordinateur.

Pour mieux comprendre son fonctionnement il nous a été demandé dans le cadre de notre Tp de faire la « reconnaissance d'objets avec le descripteur SIFT ». L'objectif de ce Tp est de pouvoir implémenter un programme qui sera capable de :

- Détecter de points d'intérêt SIFT et Descripteur SIFT.
- Reconnaissance d'objet avec SIFT.
- Calcul de descripteurs SIFT et leurs mise en correspondance (Matching) des points d'intérêt et entre des images.

0.2 Fonctionnement

0.2.1 Détection de points d'intérêt et descripteurs

La détection de points d'intérêts (ou coins) est, au même titre que la détection de contours, une étape préliminaire à de nombreux processus de vision par ordinateur. Les points d'intérêts, dans une image, correspondent à des doubles discontinuités de la fonction d'intensités. Celles-ci peuvent être provoquées, comme pour les contours, par des discontinuités de la fonction de réflectance ou des discontinuités de profondeur. L'étape fondamentale de la méthode proposée par Lowe consiste à calculer ce que l'on appelle les « descripteurs SIFT » des images à étudier.

Il s'agit d'informations numériques dérivées de l'analyse locale d'une image et qui caractérisent le contenu visuel de cette image de la façon la plus indépendante possible de l'échelle (« zoom » et résolution du capteur), du cadrage, de l'angle d'observation et de l'exposition (luminosité). Ainsi, deux photographies d'un même objet auront toutes les chances d'avoir des descripteurs SIFT similaires, et ceci d'autant plus si les instants de prise de vue et les angles de vue sont proches. D'un autre côté, deux photographies de sujets très différents produiront selon toute vraisemblance des descripteurs SIFT très différents eux aussi (pouvoir discriminant). Cette robustesse, vérifiée dans la pratique, est une exigence fondamentale de la plupart des applications et explique en grande partie la popularité de la méthode SIFT.

0.2.2 Exécution du programme

Pour tester notre programme nous avons mis en place une base d'images dataset qui contient en son sein un ensemble de classe d'objet(10 classes) dans lesquels se trouve une base train d'entraînement et une base test qui sert de test pour le modèle.

Pour exécuter le programme on se place dans le dossier qui contient le programme en question avec la commande `cd`. On exécute la commande `pythonx(x` étant la version de python installer sur votre machine) et suivant le différent programmes que nous voulons effectuer on exécute les commandes suivantes :

- Pour détecter les points d'intérêts et descripteurs SIFT on tape la commande **P_interet_Descripteurs.py**
- Pour la correspondance(matching) entre deux images on utilise la commande **match_features.py**
- et **train.py** pour observer le taux de prédiction de notre apprentissage.

0.3 Description de la phase d'implémentation

La phase d'implémentation de notre programme est constitué de trois partie :

- Création de la base images
- Processus de détection des points d'intérêt dans une image
- Le Processus de matching entre différente image.

0.3.1 Création de la base images

Pour la constitution de notre base d'image,nous avons utilisé le lien donné dans l'épreuve de notre Tp, c'est un ensemble d'image provenant de Columbia University Image Library (COIL-100) qui nous a permis de spécifier les images couleur bien distincte. Et cela a été faite en vue de respecter les consignes donnés par le TP dans le but d'avoir des bons résultats. Dans notre dataset,nous avons 10 classes que nous avons nommer lot. Dans chaque lot nous avons des données de **train** et celles de **test**. Nous avons pris 70% des données pour le training et 30% pour le test.

0.3.2 Processus de détection des points d'intérêt dans une image

Selon wikipedia " la détection de zones d'intérêt d'une image numérique (feature détection en anglais) consiste à mettre en évidence des zones de cette image jugées « intéressantes » pour l'analyse, c'est-à-dire présentant des propriétés locales remarquables. De telles zones peuvent apparaître, selon la méthode utilisée, sous la forme de points, de courbes continues, ou encore de régions connexes rectangulaires ou non et qui constituent le résultat de la détection " Les points d'intérêts présentent plusieurs avantages à savoir :

- Sources d'informations plus fiable que les contours car plus de contraintes sur la fonction d'intensité.
- Robuste aux occultations (soit occulté complètement, soit visible).
- Pas d'opérations de chaînage (contours).
- Présents dans une grande majorité d'images

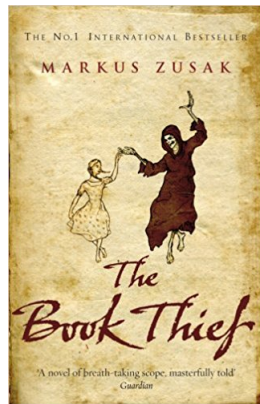


FIGURE 1 – image Originale

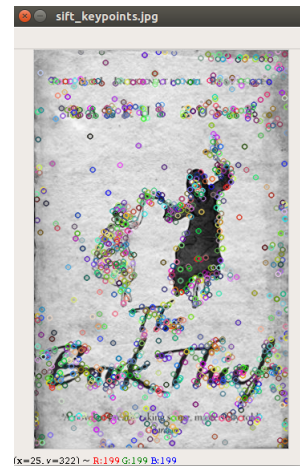


FIGURE 2 – points d'intérêt détectés

Ici nous avons une image hors base sur laquelle, nous avons appliqué une détection des points d'intérêts.

Pour cette image, nous allons générer les fonctionnalités SIFT. Nous devons d'abord construire un objet SIFT, puis utiliser la fonction **detectAndCompute** pour obtenir les points-clés.

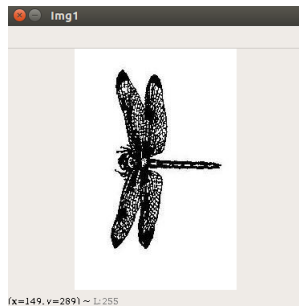


FIGURE 3 – Image_base



FIGURE 4 – Image_base

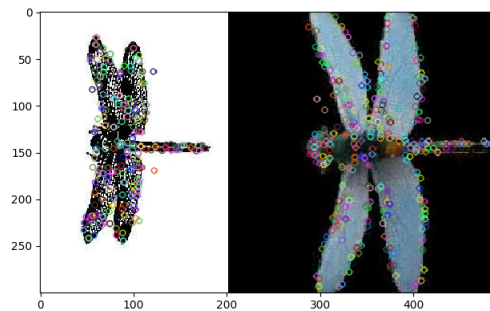


FIGURE 5 – points d'intérêt détectés_deux_images

Maintenant, pour ces deux images, nous allons générer les fonctionnalités SIFT. Nous devons d'abord construire un objet SIFT, puis utiliser la fonction **detectAndCompute** pour obtenir les points-clés. Il renverra deux valeurs : les points-clés et les descripteurs. Déterminons les points clés et imprimons le nombre total de points clés trouvés dans chaque image : en utilisant `len(Kp1)`, `len(Kp2)`.

Ces dernières sont nos images de notre base de lot1, nous constatons que les coins, les points de forte variations sont tous détectés grâce à l'algorithme de sift.

0.3.3 Mise en correspondance (matching) de points d'intérêts

La mise en correspondance est très utilisée dans le domaine de la vision par ordinateur. En effet, c'est l'une des étapes les plus importantes lors de la reconstruction 3D à partir d'une paire stéréo ou d'une séquence d'images ou lors de la détermination de la structure à partir d'un mouvement. L'étape de matching (appariement) est un processus de comparaison qui consiste à calculer les distances entre les caractéristiques.

les figures ci-dessous montre la mise en correspondance entre les images de la base lot1 (image0045) (image0009). Pour faire la correspondance des caractéristiques de l'image (image0045) avec les caractéristiques de l'image (image0009). Nous allons utiliser la fonction `match()` du module `BFMatcher` (correspondance de force brute). En outre, nous allons tracer des lignes entre les caractéristiques qui correspondent dans les deux images. Cela peut être fait en utilisant la fonction `drawMatches` dans `OpenCV`.

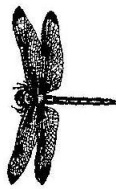
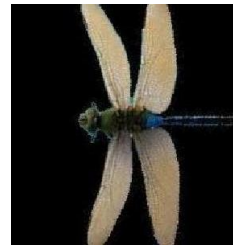
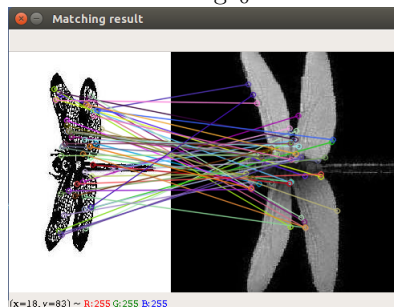
FIGURE 6 – image₀₀₀₉FIGURE 7 – image₀₀₄₅

FIGURE 8 – matching entre deux images du lot1

Ces images nous indiquent la correspondance de matching de point d'intérêt, bien qu'elles soient différentes par leurs couleurs. Ces dernières ont donné une correspondance importante, mais n'empêche-on constate quelques défauts sur les correspondances.

0.3.4 MISE EN CORRESPONDANCE DE DEUX(2) IMAGES HORS BASE

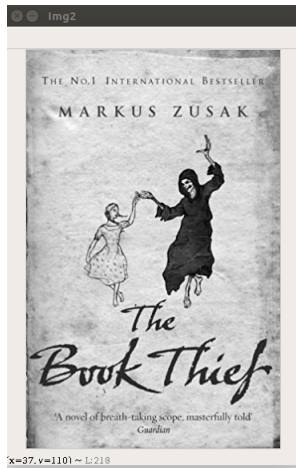


FIGURE 9 – Image originale

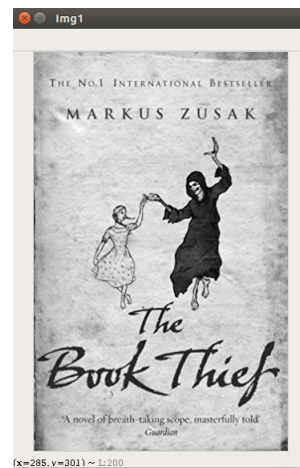


FIGURE 10 – Image originale



FIGURE 11 – matching entre deux images hors base

Sur ces deux même images, nous avons eu une correspondance parfaite car les deux images sont identiques de part leur taille, forme et couleur : d'où cette parfaite correspondance.

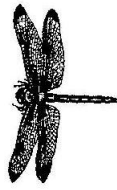


FIGURE 12 – Image originale



FIGURE 13 – Image originale

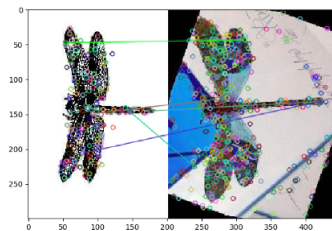


FIGURE 14 – matching entre deux images

Sur ces deux images sont les mêmes mais présente une différence de taille et de couleur d'où la correspondance de matching n'est pas parfaite.

0.4 Matrice de confusion

Une matrice de confusion est un résumé des résultats de prédictions sur un problème de classification. Les prédictions correctes et incorrectes sont mises en lumière et réparties par classe. Les résultats sont ainsi comparés avec les valeurs réelles.

Elle permet de comprendre de quelle façon le modèle de classification est confus lorsqu'il effectue des prédictions. Ceci permet non seulement de savoir quelles sont les erreurs commises, mais surtout le type d'erreurs commises. Les utilisateurs peuvent les analyser pour déterminer quels résultats indiquent comment les erreurs sont commises.

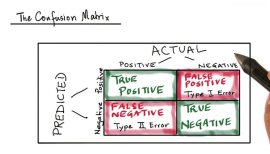


FIGURE 15 – Matrice de confusion

0.4.1 Calcul et Interprétation de la matrice de confusion sur les données de la base

Les deux figures (figure 16 et 17) ci-dessous nous montre les valeurs prédites sur chaque classe (lot) en colonne, et également la somme de toutes les valeurs prédites sur l'ensemble de classe (lot) .

En effet, nous avons pris dix classes (lots) qui contiennent au total 761 images. Sur les 761 images de notre dataset, nous avons 228 images bien prédites (figure 17) et parmi l'ensemble des classes, la classe (lot2) est l'originale

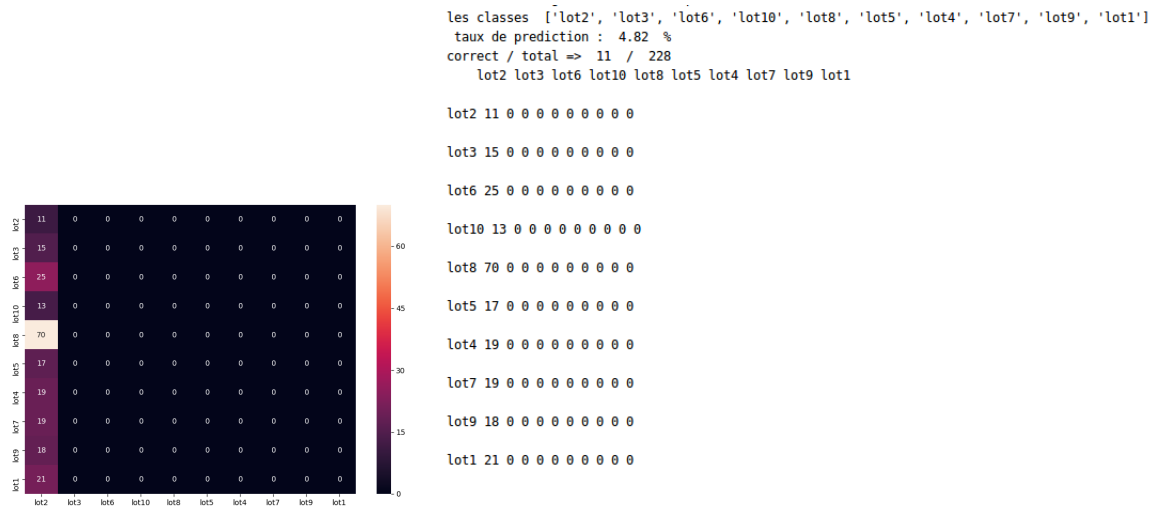


FIGURE 16 – Matrix_Confusion

les classes ['lot2', 'lot3', 'lot6', 'lot10', 'lot8', 'lot5', 'lot4', 'lot7', 'lot9', 'lot1']
 taux de prediction : 4.82 %
 correct / total => 11 / 228

lot2 lot3 lot6 lot10 lot8 lot5 lot4 lot7 lot9 lot1

lot2 11 0 0 0 0 0 0 0 0 0
 lot3 15 0 0 0 0 0 0 0 0 0
 lot6 25 0 0 0 0 0 0 0 0 0
 lot10 13 0 0 0 0 0 0 0 0 0
 lot8 70 0 0 0 0 0 0 0 0 0
 lot5 17 0 0 0 0 0 0 0 0 0
 lot4 19 0 0 0 0 0 0 0 0 0
 lot7 19 0 0 0 0 0 0 0 0 0
 lot9 18 0 0 0 0 0 0 0 0 0
 lot1 21 0 0 0 0 0 0 0 0 0

FIGURE 17 – Matrix _Confusion_lot and Taux_precision

Les images ci-dessous montrent le pourcentage de prédiction des différentes images des classes (lots).

classe originale lot2 classe prédite lot2
 taux de prediction : 100.0 %
 lot2

FIGURE 18 – prédiction lot2_lot 2

classe originale lot3 classe prédite lot2
 taux de prediction : 91.67 %

FIGURE 20 – prédiction lot3_lot 2

classe originale lot1 classe prédite lot2
 taux de prediction : 5.21 %

FIGURE 19 – prédiction lot1_lot 2

classe originale lot4 classe predite lot2
taux de prediction : 7.1 %

FIGURE 21 – prédiction lot4_lot 2

classe originale lot6 classe predite lot2
taux de prediction : 40.74 %

FIGURE 23 – prédiction lot6_lot 2

classe originale lot5 classe predite lot2
taux de prediction : 8.15 %

FIGURE 22 – prédiction lot5_lot 2

classe originale lot7 classe predite lot2
taux de prediction : 6.4 %

FIGURE 24 – prédiction lot7_lot 2

classe originale lot9 classe predite lot2
taux de prediction : 5.56 %

FIGURE 26 – prédiction lot9_lot 2

classe originale lot8 classe predite lot2
taux de prediction : 16.92 %

FIGURE 25 – prédiction lot8_lot 2

classe originale lot10 classe predite lot2
taux de prediction : 21.15 %

FIGURE 27 – prédiction lot10_lot 2

Les figures ci-dessus montre la précision de chaque image des différentes classes (lots) par rapport à la **classe (lot2) prédite**. On remarque que la matrice de confusion entre le lot2 (lot originale) et lui même nous avons un taux de prédiction 100 % qui s'explique par le fait les images sont identiques dans ces lots.

On constate aussi que la matrice de confusion au niveau du lot6-lot2, lot3-lot2 on obtient un pourcentage de 40,74% et 74.67% respectivement qui montre que la correspondance est beaucoup plus précise sur ses images.

0.5 Conclusion

La vision par ordinateur est un domaine de l'intelligence artificielle (IA) qui permet aux ordinateurs et aux systèmes de tirer des informations significatives à partir d'images numériques, de vidéos et d'autres entrées visuelles ; et de prendre des mesures ou de formuler des recommandations en fonction de ces informations.

Dans le cadre de ce Tp nous avons implémenter un programme qui permet de détecter les points d'intérêt et scripteurs d'une image, et la mise en correspondance (matching). Nous avons utiliser une base d'image provenant de Colombia University Image Library(COIL-100) qui contient plusieurs images bien distinctes. Après les analyses des résultats obtenu sur l'ensemble des test test effectuer nous pouvons dire que notre programme fonctionne bien car il arrive a bien détecté les points d'intérêts et la correspondance qui existe entre deux images.

0.6 Référence

[1]<https://towardsdatascience.com/computer-vision-an-introduction-bbc81743a2f7>

[2]<https://www.ibm.com/topics/computer-vision>

<https://www.analyticsvidhya.com/blog/2019/10/detailed-guide-powerful-sift-technique-image-matching-python/> [4]https://fr.wikipedia.org/wiki/Matrice_de_confusion

.