

Matière : Vison par Ordinateur

TP1 :Reconnaissance d'objets avec le descripteur SIFT

Auteur : Oumarou Altine Mohamadou Aminou
Ousseini Hamza Abdoul-Djalil

Enseignante : NGUYEN Thi Oanh

1 Introduction

Le présent rapport fait la description du TP1 réalisé dans le cadre du module de Vision par Ordinateur. Notre objectif ici est de pouvoir implémenté un programme capable de faire la reconnaissance d'objet avec le descripteur **SIFT**. Pour ce faire nous allons utilisé les point d'intérêt pour calculer cette méthode. Ce rapport a pour but donc d'expliquer les différents travaux réalisé ,les analyses ainsi que les commentaires et les différentes améliorations possible.

QUELLE EST LA DÉMARCHE A SUIVRE ?

De prime abord pour atteindre cet objectif notre travail consiste à faire la reconnaissance d'objets à l'aide du descripteur SIFT ou nous allons commencé par extraire les points SIFT d'images en référence c'est à dire apprentissage qui représente plus ou moins plusieurs classe d'objets. Puis a partir d'une image qui représente un objet inconnue ,nous allons extraire les différents points SIFT et Subsidiairement, comparer avec les classes déjà connues pour pouvoir identifier l'objet.

2 Fonctionnement

2.1 Détection de points d'intérêt et descripteurs

Les points d'intérêt d'une image sont en général les coins d'un objet ou les contours marquant un changement de texture, par exemple la séparation d'un arbre et du ciel.La méthode la plus connue pour détecter les points d'intérêt est celle de Harris. Quelques temps après, D.Lowe, chercheur canadien, a amélioré l'idée d'Harris. La méthode de Lowe permet d'avoir une plus grande précision car elle détecte en plus des points d'intérêt, les régions d'intérêt. L'algorithme qu'a conçu D.Lowe s'intitule Scale-invariant feature transform (SIFT). Cet algorithme permet aussi la mise en correspondance des points contrairement à celui de Harris, c'est-à-dire que sa méthode sait trouver un même point dans deux images différentes représentant le même objet.

Dans la première étape de la méthode SIFT, on cherche d'abord à trouver les points clés. Ensuite, pour pouvoir les apparier, on stocke différentes informations telles que la luminosité ou l'angle d'observation. Ces informations doivent être indépendantes de l'échelle de l'objet. On les appelle les descripteurs SIFT.

2.2 Comment exécute-t-on le 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(15 classes) dans lesquels se trouve une base **train** d'entraînement et une base **test** qui sert de test pour le model. *Il est également à noter que chacun des différents*

répertoires contient plusieurs images.

NB : A considérer également pour la subdivision des bases d'image nous avons optez pour le système 70% pour la base d'entraînement et 30% pour le test. Vu la nature conséquente de la base d'origine ,et en vue d'une exécution rapide du programme nous avons pris cent (100) images par classe d'objet. L'exécution du programme. Le programme contient trois fichiers *.py* à savoir : un fichier nommé `point_d_interets_descripteurs.py` qui detecte les points d'intérêt sur une image (detection de SIFT), il existe ce qu'on appelle descripteur de points d'intérêt (comme par exemple descripteur SIFT) pour qu'à la suite si on veut comparer deux images, on compare leurs descripteurs de points d'intérêt, l on , `feature_matching.py` et `train.py`

- on se place dans le dossier qui contient le programme en question avec la commande **cd**
- puis exécuter la commande **pythonx**(x étant la version de python installer sur votre machine) suivi **train.py**
- Ensuite exécuter le programme au moyen de la commande : **pythonx point_d_interes_et_descripteurs** pour observé les point d'intérêt ainsi que les descripteurs des images mis en paramètres.
- et en fin **pythonx match_features.py** pour les différentes correspondance entre image par le processus de matching.

3 Description de la phase d'implementation

Pour la phase d'implementation du programme nous avons subdivisé ce dernier en quelques partie essentielle :

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

3.1 Création de la base images

Dans l'optique de pouvoir constituer la base d'image,nous avons télécharger une base d'image provenant de **Columbia University Image Library (COIL-100)** qui nous a permis à spécifier les images des différentes base. En effet ces images sont formés de plusieurs objet différents avec chacune une couleur bien distincte. Et cela a été faites en vue de respecter les consignes donnés par le TP dans le but d'avoir des bons résultats.

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

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. Ce sont par exemple : les coins, les jonctions en T ou les points de fortes variations de texture.

L'algorithme SIFT (Scale Invariant Feature Transform) a été proposé par David Lowe, université de British Columbia, en 1999 pour détecter et décrire des zones d'intérêts (local features) dans une image. A noter qu'il s'agit ici non seulement de détecter mais aussi de caractériser, par des valeurs, pour pouvoir reconnaître (mettre en correspondance) par la suite ces zones ou points d'intérêts dans d'autres images de la même scène.



FIGURE 1 – image originale

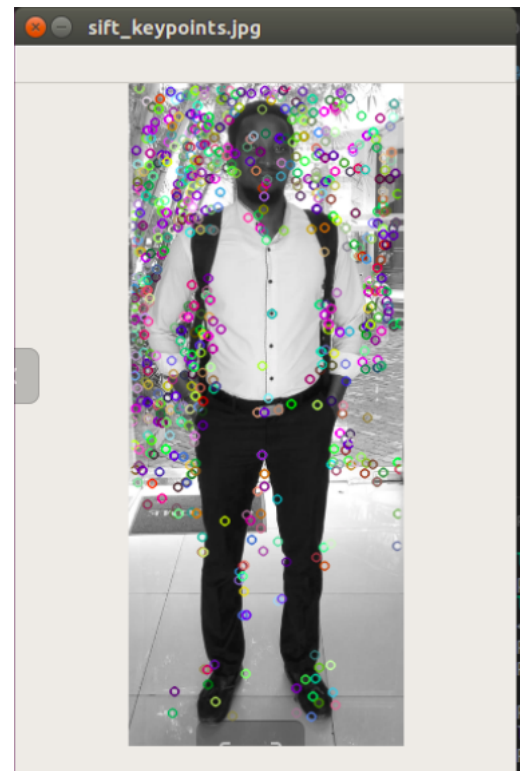


FIGURE 2 – point d'intérêt détecter



FIGURE 3 – image originale



FIGURE 4 – point d'intérêt détecter

Dans cette partie nous observons deux images avec différentes couleurs et positions différentes. Nous constatons également l'apparition des points d'intérêt qui sont dus à l'algorithme utilisé. Nous re-

marquons également que l'algorithme de détection des points d'intérêt s'est focalisé généralement sur les points particuliers des différents contours qui sont sélectionnés suivant un critère bien défini. Ainsi, les coins sont les points de l'image où le contour change brutalement de direction, comme par exemple aux quatre sommets d'un rectangle. Il s'agit de points particulièrement stables, et donc intéressants pour la répétabilité comme le montre les images à droite.

3.3 Mise en correspondance (**matching**) de points d'intérêt

Le matching de 2 images (Figure 21) intervient après la détection des points d'intérêt et l'extraction des caractéristiques du voisinage de ces points. L'étape de matching (appariement) est un processus de comparaison qui consiste à calculer les distances entre les caractéristiques. les figure ci-dessous montre la mise en correspondance entre l'objet1_15 et l'objet1_65.



FIGURE 5 – image objet1-15 normale



FIGURE 6 – image objet1-65 normale

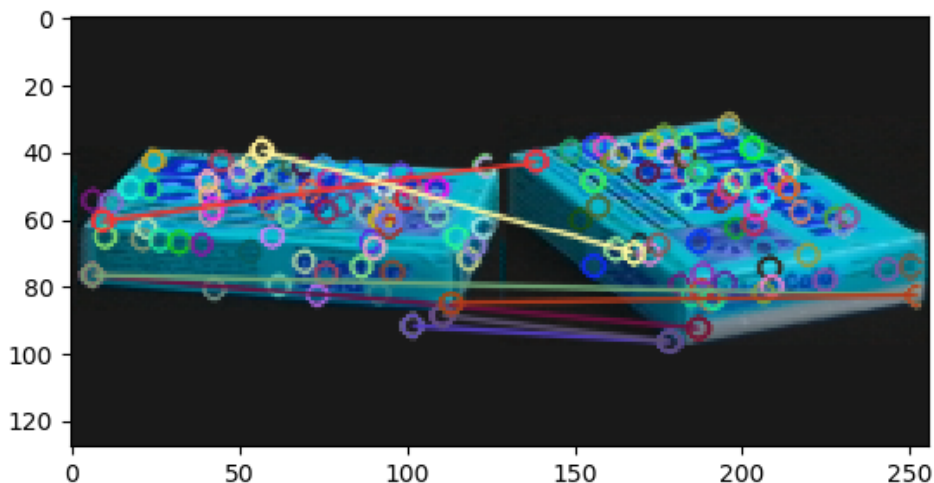


FIGURE 7 – matching entre deux images

MISE EN CORRESPONDANCE DE DEUX(2) IMAGES HORS BABSE

Bien que les deux images se distinguent de leur position et la couleur, la correspondance a belle et bien été faite sur ces dernières.



FIGURE 8 – image objet1-15 normale



FIGURE 9 – image objet1-65 normale

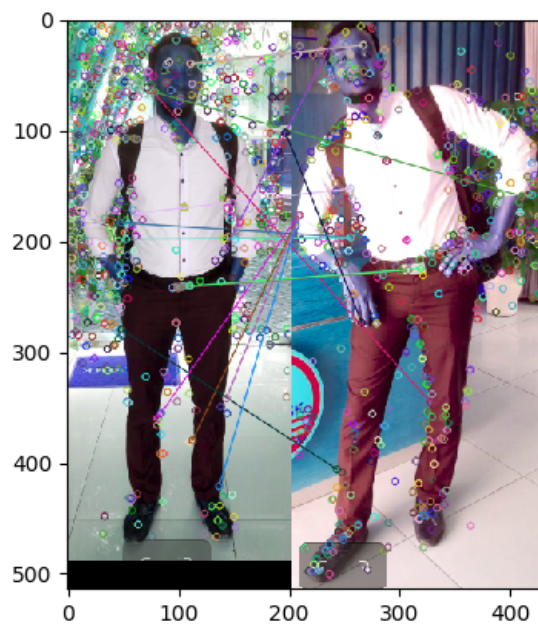


FIGURE 10 – matching entre deux images hors base

4 La matrice de confusion

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

Une matrice de confusion est un tableau souvent utilisé pour décrire les performances d'un modèle de classification (ou "classificateur") sur un ensemble de données de test pour lesquelles les valeurs vraies sont connues. La matrice de confusion elle-même est relativement simple à comprendre, mais la terminologie associée peut prêter à confusion.

		Réalité	
		Positifs	Négatifs
Prédictions	Positifs	Vrais positifs (VP)	Faux positifs (FP)
	Négatifs	Faux Négatifs (FN)	Vrais Négatifs (VN)

FIGURE 11 – tableau d'une matrice de confusion

Dans le but d'avoir des bons résultats nous avons calculer la matrice de confusion avec plusieurs valeurs différentes. Ci-dessous les différentes captures lustrant ce expliqué haut.

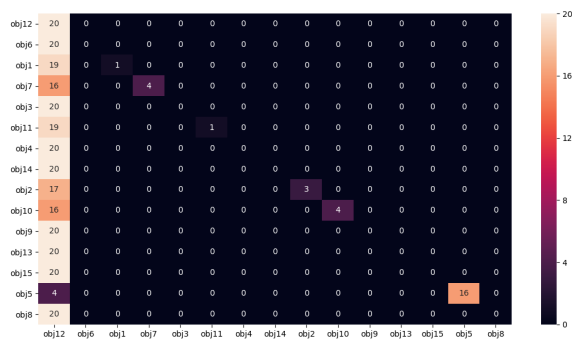


FIGURE 12 – matrice de confusion avec 0.1

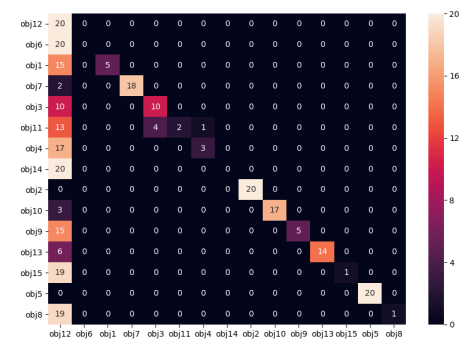


FIGURE 13 – matrice de confusion avec 0.2

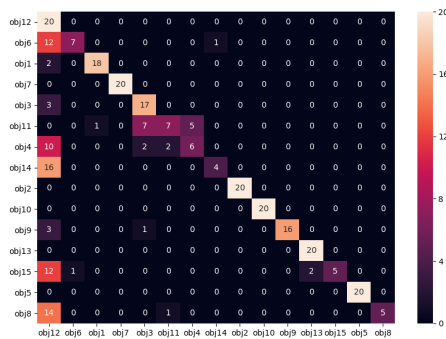


FIGURE 14 – matrice de confusion avec 0.3

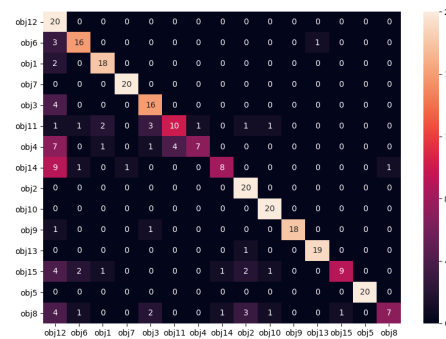


FIGURE 15 – matrice de confusion avec 0.4

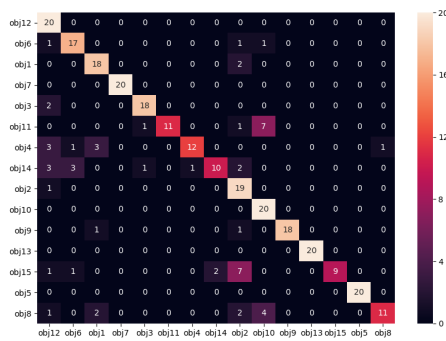


FIGURE 16 – matrice de confusion avec 0.5

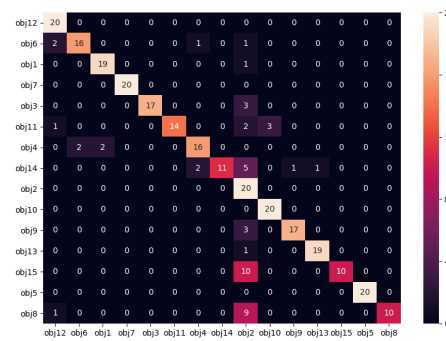


FIGURE 17 – matrice de confusion avec 0.6

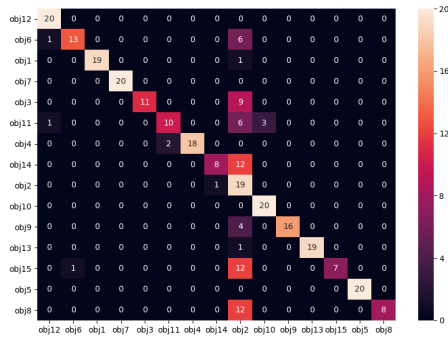


FIGURE 18 – matrice de confusion avec 0.7

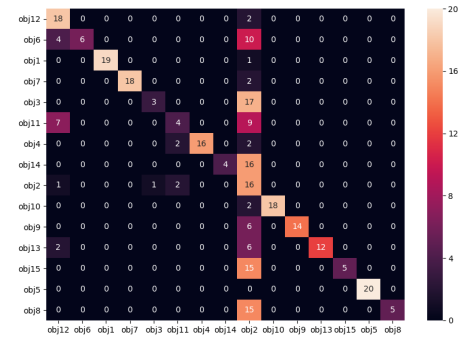


FIGURE 19 – matrice de confusion avec 0.8

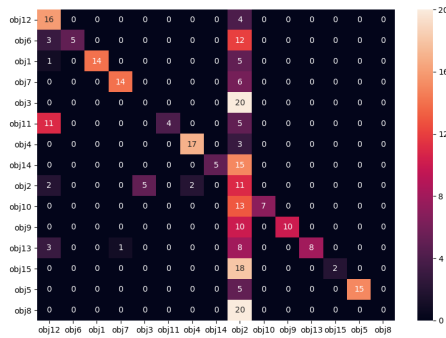


FIGURE 20 – matrice de confusion avec 0.9

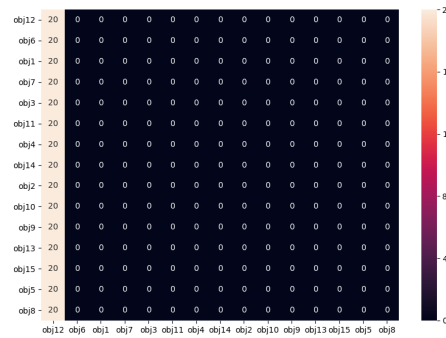


FIGURE 21 – matrice de confusion avec 1

Les images ci-dessous montre le pourcentage de prédiction des différents objet.

taux de prediction : 16.33 %
correct / total => 49 / 300

FIGURE 22 – % de prédiction avec 0.1

taux de prediction : 67.67 %
correct / total => 203 / 300

FIGURE 24 – % de prédiction avec 0.3

taux de prediction : 76.67 %
correct / total => 230 / 300

FIGURE 26 – % de prédiction avec 0.5

taux de prediction : 72.67 %
correct / total => 218 / 300

FIGURE 28 – % de prédiction avec 0.7

taux de prediction : 41.0 %
correct / total => 123 / 300

FIGURE 30 – % de prédiction avec 0.9

taux de prediction : 44.0 %
correct / total => 132 / 300

FIGURE 23 – % de prédiction avec 0.2

taux de prediction : 75.33 %
correct / total => 226 / 300

FIGURE 25 – % de prédiction avec 0.4

taux de prediction : 79.33 %
correct / total => 238 / 300

FIGURE 27 – % de prédiction avec 0.6

taux de prediction : 59.33 %
correct / total => 178 / 300

FIGURE 29 – % de prédiction avec 0.8

taux de prediction : 6.67 %
correct / total => 20 / 300

FIGURE 31 – % de prédiction avec 1

A cet effet, une matrice de confusion détaillée pour ce descripteur est calculée comme le montre les figures ci-dessus. Dans ces figures, les éléments des classes sont représentés par des pixels noirs sur chacune des lignes de la matrice. Le rang de la ligne correspond au numéro des objets de la base utilisé comme requête. L'information contenue dans ces matrices nous permet, y compris, d'identifier les symboles graphiques de la catégorie qui sont bien ou mal classés. Ainsi, les éléments des différentes diagonales (représenté par des petits rectangle) représentent le nombre d'images bien classées sur chacun des objets.

La présence de vrai positif et faux négatifs sur les différentes matrices de confusion observé explique au mieux la variation de la nature des différents objet de la base.c'est le cas notamment sur l'images ci-dessous :

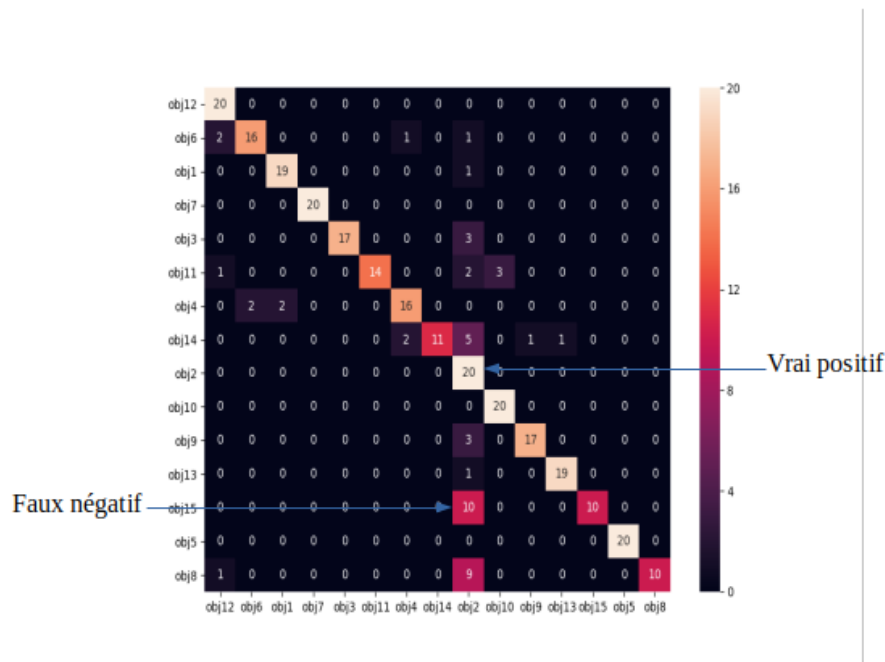


FIGURE 32 – Détection de vrai et faux positif

Sur l'image ci-dessus nous constatons un vrai positif qui est dû au fait la prédiction de l'élément a été bien prédit (objet2,objet2). Par contre il y'a un faux négatif avec l'objet2 qui prédit l'objet15 .C'est qui ne pas vrai.

En lisant la matrice de confusion, on peut aisément constater que l'insuffisance du taux de prédiction global (61,8%) résulte de la confusion entre certains objet comme sur la matrice de confusion avec la valeur 0.6 par défaut on constate l'objet2(20) et l'objet9(17) puis l'objet1(19) et l'objet11(14). A l'image des objets2 et objets10,objets7 puis en fin objets5, plusieurs images ont été bien discriminées, ce qui prouve que, visuellement, ces images sont bien différenciables.

5 Conclusion

La vision par ordinateur est un processus de traitement de l'information. Elle utilise des stratégies bien définies permettant d'atteindre ses buts. L'entrée d'un système de vision est constituée par une séquence d'images. Le système lui-même apporte un certain nombre de connaissances qui interviennent à tous les niveaux. La sortie est une description de l'entrée en termes d'objets et de relations entre ces objets.

Au cours de ce TP nous avons implémenté un programme permettant de détecter les points d'intérêt SIFT ainsi que les descripteurs dans une image. Nous avons créé pour ce faire une base de 1500 images différentes de par leurs natures, couleurs, luminosités et décors pour créer des images ne faisant que ressortir les points d'intérêt et la correspondance entre deux images. Ainsi après plusieurs expérimentations réalisées nous retenons que notre programme fonctionne bien car il détecte bien les points d'intérêt des images quel qu'en soit leurs couleurs même si nous constatons aussi la présence de certaines anomalies ou des pixels n'ayant aucun trait avec les images.

6 Référence

1. Camillerapp, J. (2012, March). Utilisation des points d'interet pour rechercher des mots imprimés ou manuscrits dans des documents anciens. In CIFED 2012-colloque international sur l'écrit et le document.
2. [https ://stackoverflow.com/fr/q/1858241](https://stackoverflow.com/fr/q/1858241)
3. [https ://stackoverflow.com/fr/q/7957169](https://stackoverflow.com/fr/q/7957169)
4. [https ://fr.wikipedia.org/wiki/Matrice_de_confusion](https://fr.wikipedia.org/wiki/Matrice_de_confusion)
5. Thuilier, J. (2012). Contraintes préférentielles et ordre des mots en français (Doctoral dissertation, Université Paris-Diderot-Paris VII).