



INSTITUT
FRANCOPHONE
INTERNATIONAL

Vision par Ordinateur

Promotion 24 – S.I.M.

RAPPORT TP 2

Reconnaissance d'Objets avec le
descripteur SIFT

ACHILLE Peterson
&
TELEMAQUE Astrel

Encadrant : Dr. NGUYEN Thi Oanh

Date de Remise : *Janvier 2020*



Since 1906
VNU
ĐẠI HỌC QUỐC GIA HÀ NỘI
Vietnam National University, Hanoi

Table des matières

Introduction	3
2.- Objectif.....	4
3.- Fonctionnement Générale du Système	4
4.- Exécution du programme.....	4
5.- Implémentation du programme.....	5
5.1.- Préparation du donnée	5
5.2.- Bibliothèques utilisées	5
5.3.- Importation des données	5
6.- Détection de points d'intérêt et descripteurs	6
7.- Mise en correspondance	8
7.1.- Matching	8
7.2.- Mise en correspondance de deux images différentes	9
8.- Matrice de confusion	10
Conclusion et Perspective.....	12
Références Scientifiques	12

Introduction

De nos jours avec l'avancé de l'internet et des technologies modernes, notre société est devenue un monde d'image. La vision par ordinateur est un domaine interdisciplinaire qui traite de la façon dont les ordinateurs peuvent être conçus pour acquérir une compréhension de haut niveau à partir d'images ou de vidéos numériques. Du point de vue de l'ingénierie, il cherche à automatiser les tâches que le système visuel humain peut effectuer. "La vision par ordinateur concerne l'extraction, l'analyse et la compréhension automatiques des informations utiles d'une image unique ou d'une séquence d'images. Elle implique le développement d'une base théorique et algorithmique permettant la compréhension visuelle automatique." En tant que discipline scientifique, la vision par ordinateur s'intéresse à la théorie sous-jacente aux systèmes artificiels qui extraient des informations à partir d'images. Les données d'image peuvent prendre de nombreuses formes, telles que des séquences vidéo, des vues de plusieurs caméras ou des données multidimensionnelles d'un scanner médical. En tant que discipline technologique, la vision par ordinateur cherche à appliquer ses théories et ses modèles à la construction de systèmes de vision par ordinateur.

Pour mieux comprendre son fonctionnement Madame 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.

2.- Objectif

L'objectif de ce TP consiste à faire la reconnaissance d'objets à l'aide du descripteur SIFT.

Dans un premier temps, nous allons extraire les points SIFT d'images de référence (d'apprentissage) représentant plusieurs classes d'objets. Ensuite, à partir d'une image représentant un objet inconnu, on extrait les points SIFT et on compare avec les classes connues pour identifier l'objet.

3.- Fonctionnement Générale du Système

Dans cette section, nous allons décrire le fonctionnement général de notre système qui permet de faire la Reconnaissance d'objets avec le descripteur SIFT. La reconnaissance d'objets, nous permet de détecter la présence d'une instance ou d'une classe d'objets, dans une image ou une scène naturelle. Selon Neisser, notre capacité à reconnaître un objet consiste en deux _étapes : un processus de sélection pour extraire les informations les plus pertinents, et une chaîne complexe des processus pour identifier l'objet. En effet, Pour ce faire avec notre système, nous avons utilisé une base d'images qu'on a divisée en deux grandes parties, le training (5768 images) et le test (1443 images). Quant au training, elle est utilisée pour extraire les descripteurs des images d'entraînement. Ensuite, dans la partition test, on choisit une image comme image d'entrée dans le système, son descripteur est ensuite extrait pour déterminer les descripteurs avec lesquels il correspond le mieux parmi les descripteurs d'entraînement. Ainsi, le système évalue le pourcentage de correspondance en fonction de sa catégorie.

4.- Exécution du programme


Nous avons travaillé sur deux fichiers de codes, l'un pour la manipulation effectuée avec SIFT sur une image (main.py) et la deuxième partie pour utiliser le dataset Coil-100 (TP2-VisionParOrdinateur.ipynb)

Pour exécuter le programme on se place dans le dossier qui contient le programme en question avec la commande `cd`. On exécute la commande `python(x étant la version de python installée sur votre machine)` et suivant les différents programmes que nous voulons effectuer on exécute la commande, Pour détecter les points d'intérêts et descripteurs SIFT et matching on tape la commande `main.py`

5.- Implémentation du programme

5.1.- Préparation du donnée

Nous avons choisi la base d'images coil-100. Après avoir télécharger cette base nous avons diviser en deux la base respectivement pour les données de test et d'entrainement, soit 80% Train et 20% Test.



The screenshot shows a Jupyter Notebook interface with the title 'TP-VisionParOrdinateur.ipynb'. The menu bar includes 'Fichier', 'Modifier', 'Affichage', 'Insérer', 'Exécution', 'Outils', 'Aide', and a status message 'Toutes les modifications ont été enregistrées'. The left sidebar has icons for a menu, search, expand, and a file explorer. The main area shows a code cell with the following Python code:

```
[8] # Division des images en train et test
X_train, X_test, y_train, y_test = train_test_split(list_image, list_label, test_size=0.2, random_state=100)
print("Nombre d'image d'entrainement : {}".format(X_train.shape[0]))
print("Nombre d'image de test : {}".format(X_test.shape[0]))
```

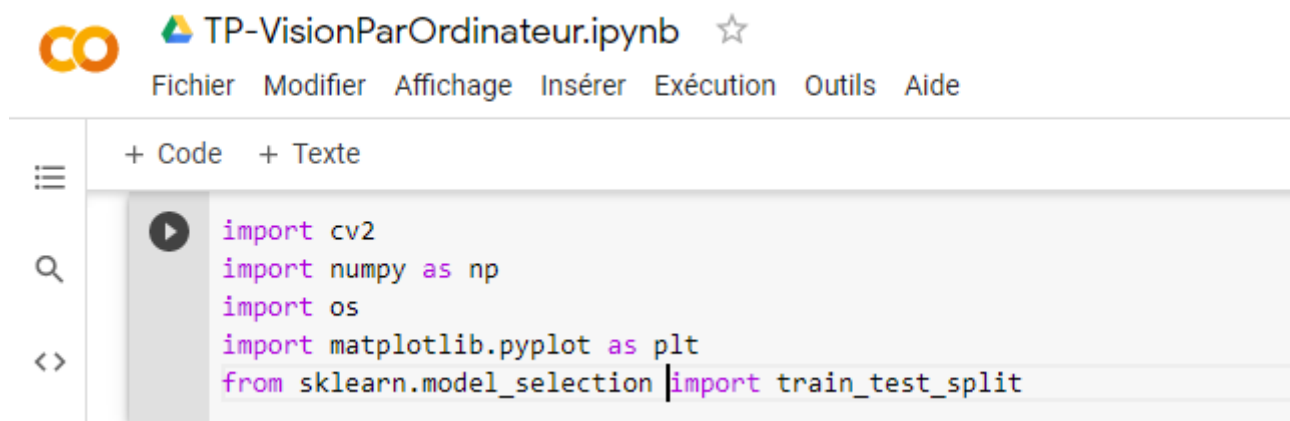
Below the code cell, the output is displayed:

```
Nombre d'image d'entrainement : 5768
Nombre d'image de test : 1443
```

Figure 1 : Préparation Donnée

5.2.- Bibliothèques utilisées

Nous avons utilisé les différentes bibliothèques comme OpenCV, Matplotlib, Numpy, Os, Panda, Scikit-learn, Scipy, Seaborn, ect..



The screenshot shows a Jupyter Notebook interface with the title 'TP-VisionParOrdinateur.ipynb'. The menu bar includes 'Fichier', 'Modifier', 'Affichage', 'Insérer', 'Exécution', 'Outils', 'Aide'. The left sidebar has icons for a menu, search, expand, and a file explorer. The main area shows a code cell with the following Python code:

```
import cv2
import numpy as np
import os
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
```

Figure 1: Bibliothèques

5.3.- Importation des données

Path_Images nous donne le chemin d'accès à notre dataset se trouvant sur le drive et liste_Images contiendra les différents objets. Nous avons crée deux tableaux : l'un pour contenir les objets (Images_C) et l'autre pour contenir les indexes ou labels pour les différents objets (Index_C). Au regard des résultats obtenus, nous avons au total 7211Objets et 7211 Indexes.



```
img_path = "/content/drive/MyDrive/TP2-VisionParOrdinateur/coil-100"

list_file = [f for f in os.listdir(img_path) if os.path.isfile(os.path.join(img_path, f))]

list_image = []
list_label = []

for file in list_file :
    if file.split(".")[1] == 'png':
        img = cv2.imread(img_path+"/"+file)
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        list_image.append(img)
        list_label.append(file.split('__')[0])

list_image = np.array(list_image)
print(list_image.shape)

list_label = np.array(list_label)
print(list_label.shape)
```

(7211, 128, 128, 3)
(7211,)

Figure 2 : Importation de données

6.- 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.

```
CO TP-VisionParOrdinateur.ipynb ☆
Fichier Modifier Affichage Insérer Exécution Outils Aide Toutes les modifications ont été enregistrées

+ Code + Texte

[9] sift = cv2.xfeatures2d.SIFT_create()

[10] # Créer les descripteurs et les points clés de toutes les images d'entraînement
X_key = []
X_descriptor = []

for img in X_train:
    img_gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
    kp, des = sift.detectAndCompute(img_gray, None)
    X_key.append(kp)
    X_descriptor.append(des)

X_key = np.array(X_key)
X_descriptor = np.array(X_descriptor)
print(X_key.shape)
print(X_descriptor.shape)
```

Figure 3 : Code d'implémentation descripteurs et les points clés

✚ Les points clés une image entraîne de la base donnée Coil-100

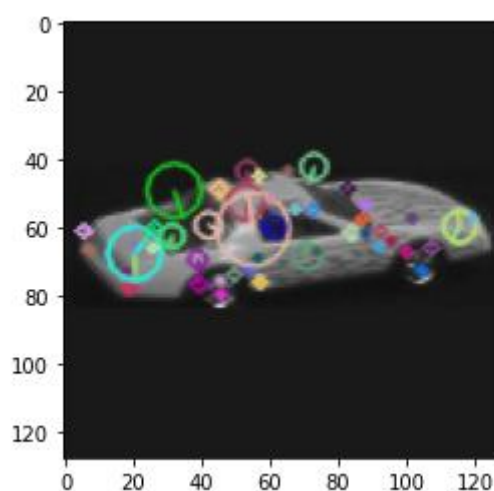


Figure 4.- Les descripteurs des points d'intérêts sur une image entraînée

✚ Les points clés une image quelconque avec le code python du descripteur SIFT



Figure 5.- les descripteurs des points d'intérêts sur une image quelconque

7.- Mise en correspondance

La mise en correspondance des points d'intérêt peut être utilisée pour estimer les paramètres de la transformation qui relie deux images (comme par exemple une homographie).

C'est également, comme nous allons le voir, une étape importante de la mise en correspondance à partir de germes.

7.1.- Matching

Considérons deux images 1 et 2 d'une scène, il s'agit ici de déterminer, pour un élément de l'image 1, l'élément qui lui correspond dans l'image 2 et éventuellement dans d'autres images. La mise en correspondance de primitives est un problème fondamental de la vision par ordinateur. C'est un processus intermédiaire entre les processus dit de hauts niveaux :

Reconstruction, reconnaissance, etc. et ceux de bas niveaux : extraction d'indices.



Figure 6.-Mise en correspondance d'une image par elle-même

Electivement, nous remarquons que le nombre des points détectés est le même et les points sont bien mis en correspondance. Donc les points d'intérêts sont bien détectés et la correspondance est respectée.

7.2.- Mise en correspondance de deux images différentes

Nous remarquons que certains points d'intérêts ne correspondent à aucun point dans l'image Correspondant, il est tout à fait logique puisque l'objet se trouve sous des angles différents, en effet, il y a certains points ne correspondent à aucun point (image au centre), certains correspondent à d'autres points (image à droite), c'est dû à la considération des points jonctions.

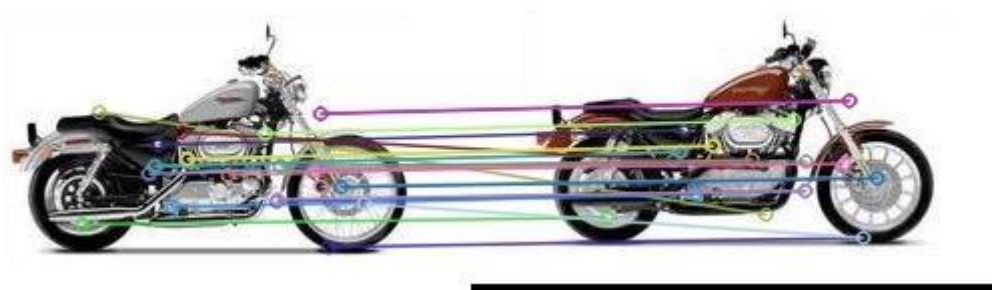


Figure 7- Mise en correspondance deux images différentes (1)



Figure 8- Mise en correspondance deux images différentes (2)

Nous remarquons qu'il y a un seul point de correspondance entre ces deux images, tout a fait logique, puisque ce sont des images différentes et qui n'appartient même pas à la même catégorie. le seul point correspond est une confidence. Par ailleurs la première image n'a que deux images par ce que c'est une image homogène donc il n'y a presque pas des points d'intérêts.

8.- 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.

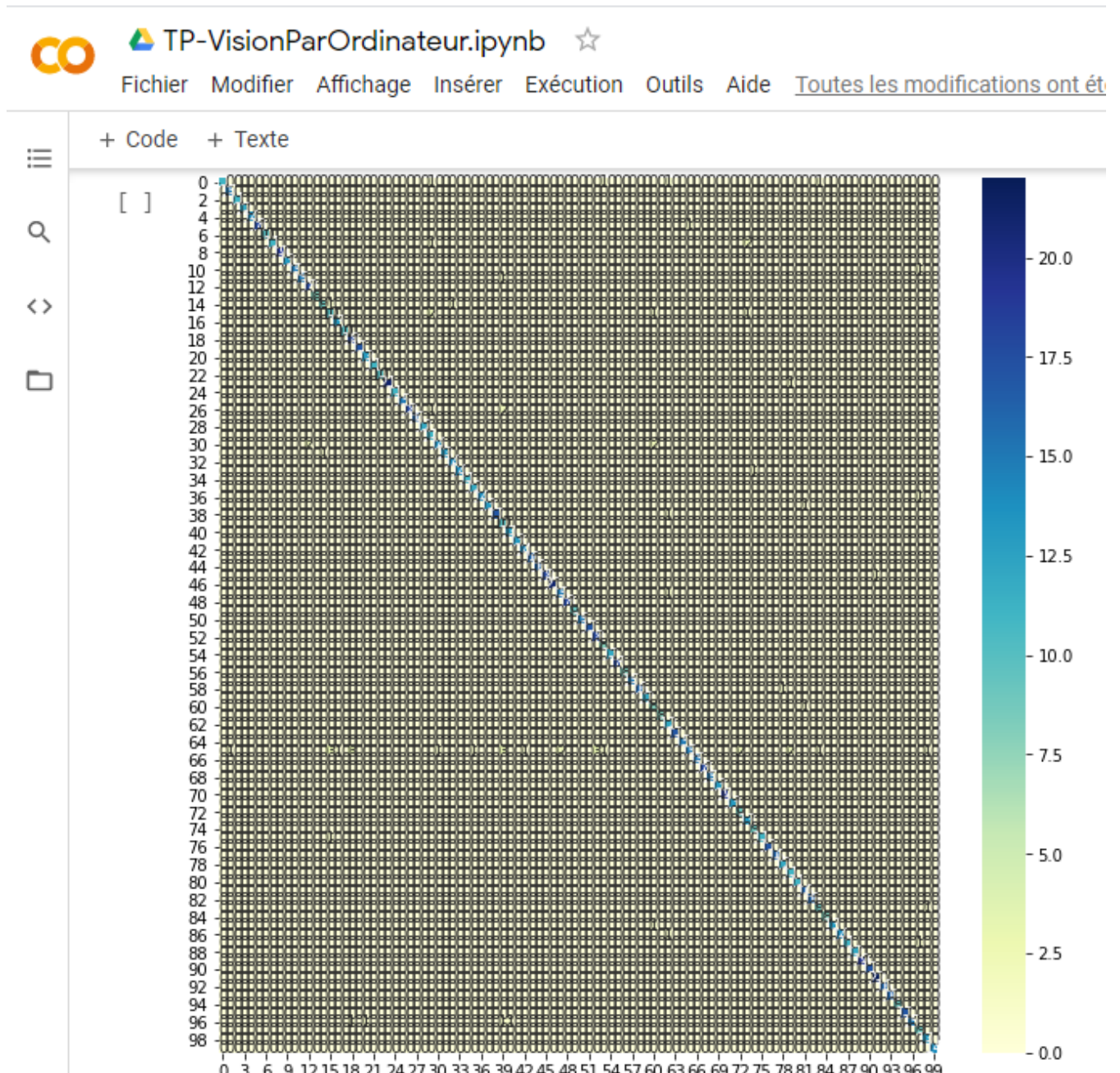


Figure 9 : Matrice de Confusion

Nous n'avons pas de mauvaises prédictions d'objets par rapport aux classes correspondantes.

Conclusion et Perspective

Au terme de ce TP, nous avons pu atteindre tous les objectifs qui nous avaient été assignés au départ qui consistait à faire la reconnaissance des objets à l'aide du descripteur SIFT.

A souligner que nos résultats sont beaucoup plus encourageant vis-à-vis du cahier de charges du TP à travers les voyants de la matrice de confusion.

Cependant, il est difficile de détecter les images disposant de très peu (nombre réduit) de descripteurs comme ce sont le cas des images lices.

Cependant, comme toute œuvre humaine, ce programme n'est pas parfait. En perspective, nous comptons améliorer notre programme avec l'évolution du cours tout en ajoutant ou bien essayant d'autres méthodes de reconnaissance des objets entre autres.

Références Scientifiques

- L. Younes, B. Romaniuk, E. Bittar, Comprendre et paramétrer l'algorithme SIFT (Scale Invariant Feature Transform). 17 novembre 2011.
- <https://www.cs.ubc.ca/~lowe/keypoints/>
- <https://towardsdatascience.com/computer-vision-an-introduction-bbc81743a2f7>
- <https://www.ibm.com/topics/computer-vision>
- <http://www.analyticsvidhya.com/blog/2019/10/detailed-guide-powerful-sift-techniqueimagematching-python/>
- https://fr.wikipedia.org/wiki/Matrice_de_confusion