



INSTITUT  
FRANCOPHONE  
INTERNATIONAL

**UNIVERSITÉ NATIONALE DU VIETNAM À HANOÏ  
INSTITUT FRANCOPHONE INTERNATIONALE**

---

OPTION : SYSTÈMES INTELLIGENTS ET MULTIMÉDIA (SIM)

PROMOTION : 25 ANNÉE ACADÉMIQUE 2021-2022

MODULE : VISION PAR ORDINATEUR

**RECONNAISSANCE D'OBJETS AVEC LE  
DESCRIPTEUR SIFT**

**SOLUTION PROPOSÉES PAR LE GROUPE 3  
CONSTITUÉ DE :**

**TANKEU NGUEKEU BAUDELAIRE ISMAËL  
TCHOTANEU NGATCHA Giresse**

**ENCADRÉE PAR :**

**NGUYEN THI OANH – IPH, [oanhnt@soict.hust.edu.vn](mailto:oanhnt@soict.hust.edu.vn)**

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Résultat attendu</b>	<b>3</b>
<b>3</b>	<b>Description du processus de travail</b>	<b>3</b>
3.1	L'entraînement . . . . .	3
3.2	Le test du modèle . . . . .	4
<b>4</b>	<b>Évaluation</b>	<b>5</b>
4.1	hyperparamètre seuil = 0.75 et k = 10 . . . . .	5
4.2	hyperparamètre seuil = 0.8 et k = 10 . . . . .	7
4.3	hyperparamètre seuil = 0.7 et k = 10 . . . . .	7
4.4	hyperparamètre seuil = 0.7 et k = 100 . . . . .	8
<b>5</b>	<b>Conclusion</b>	<b>8</b>

## Table des figures

1	Visualisation de quelques images de notre dataset . . . . .	4
2	Image reflétant le résultat attendu par notre modèle . . . . .	4
3	Image de même classe . . . . .	4
4	Matrice de confusion avec les hyper-paramètres seuil = 0.75 et k = 10 . . . . .	6
5	Score de notre modèle avec ou sans celles qui ont moins de 75% . . . . .	7
6	Classes ayant un score moins de 75% . . . . .	7
7	Score de notre modèle avec ou sans celles qui ont moins de 75% . . . . .	7
8	Nombre de classe à moins de 75% . . . . .	7
9	Score de notre modèle avec ou sans celles qui ont moins de 75% . . . . .	7
10	Classes ayant un score moins de 75% . . . . .	8
11	Score de notre modèle avec ou sans celles qui ont moins de 75% . . . . .	8
12	Nombre de classes ayant un score moins de 75% . . . . .	8

# 1 Introduction

Avec l'évolution de la technologie, nous avons assisté à une explosion de la quantité de donnée numériques. Elles peuvent être regroupés en plusieurs classes telle que les images, les contenus textes, etc. Ces données peuvent être exploitées afin de créer des modèles d'intelligence artificiel en particulier celle de la vision par ordinateur. C'est un domaine de l'intelligence artificielle qui essaye de reproduire les capacités visuelles humaines à partir des programmes informatiques. Le but sera de construire des modèles informatiques qui pourront détecter ou reconnaître les objets de manières non ambiguës dans une image ou une vidéo. Ainsi dans le cadre de notre Travail Pratique, il nous a demandé de créer un classificateur d'images sur un jeu de données de 100 classes en utilisant les descripteurs de **SIFT** (Scale-Invariant Feature Transform)

Dans le cadre de notre travail, il nous est demandé d'utiliser les descripteurs de points d'intérêts pour pouvoir créer notre modèle. Grâce aux travaux de (David Lowe et al.) [1] qui utilise le descripteur **SIFT** pour pouvoir reconnaître un objet image malgré la transformation qu'il aurait subie. Il sera question d'utiliser la même technique afin de construire un modèle de classification basé sur les étapes suivantes :

- Créer un modèle de nos descripteurs d'images d'entraînement ;
- A partir des descripteurs de nos images image test, on fera un matching avec nos descripteurs d'images d'entraînement afin de prédire leur classe d'appartenance ;
- Ainsi le modèle obtenu pourra être évalué à partir de la **matrice de confusion** ;

Comme outils de mise en place de notre modèle, nous avons utilisé : **Python3** et **Colab**. Pour le faire, nous avons structuré notre travail comme suite.

## 2 Résultat attendu

Il est question pour nous de construire un modèle qui se sert des descripteurs d'images pour prédire la classe d'appartenance à une image. Le modèle obtenu sera évalué par une série d'images tests et par la suite nous allons nous servir de la matrice de confusions afin d'observer les résultats de test dans le souci de trouver des stratégies pour améliorer les performances de notre modèle. Ces stratégies seront basées sur la modification des valeurs des hyper-paramètres qui sont le coefficient de similarité nommé **seuil** et nombre de voisin dans l'approche de **KNN** lors de la classification.

## 3 Description du processus de travail

### 3.1 L'entraînement

Nous avons utilisé le jeu de données Columbia University Image Library qui contient cent classes d'images donc chacune des classes contient soixante-douze images par classe donnant un total de 7200 images. Dans un premier temps, nous avons subdivisé de manière automatique et aléatoire de notre jeu de données en deux, représentant chacun la moitié de notre jeu de données c'est-à-dire 3600 pour le test et de même pour l'entraînement. Un affichage aléatoire permet de visualiser quelques images de ce jeu de données d'entraînement. Nous observons qu'il y a des images similaires, car il s'agit de la même image ayant subi une rotation de 5 degrés lors de la prise de vue. Ces images représentent des images de la même classe d'appartenance. Par la suite, nous avons calculé les descripteurs de toutes nos images d'entraînement que nous avons stocké individuellement dans un fichier. Ces différents descrip-



FIGURE 1 – Visualisation de quelques images de notre dataset

teurs d'images pourront être par la suite utilisés pour faire du matching avec les descripteurs des images tests afin de prédire le résultat de classification de la classe.

### 3.2 Le test du modèle

Par la suite, nous avons choisir une image de manière aléatoire afin de prédire la classe d'appartenance de cette image. Nous avons choisir l'image suivante :

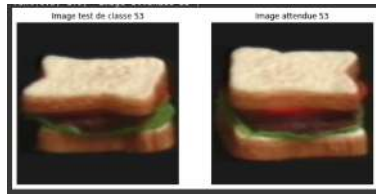


FIGURE 2 – Image reflétant le résultat attendu par notre modèle

Nous allons par la suite calculé le descripteur de l'image test. Le descripteur obtenu, sera matché avec les autres descripteurs de nos données d'entraînement. Pour le matching, nous avons utilisé le ratio de la distance entre deux vecteurs que nous avons dit qu'il doit être inférieure à la valeur seuil de **0.75**. Cette valeur seuil permet de filtrer pour avoir les bons matching c'est-à-dire un vecteur de descripteur test qui matche sur deux vecteurs de descripteurs des images d'entraînement.

Les résultats obtenus et l'exploitation de [2] vont nous permettre d'utiliser une approche pour la classification. Pour pouvoir déterminer la classe d'appartenance de l'image test, nous avons utilisé l'approche des KNN avec  $K = 10$ . Nous obtenons le résultat suivant :

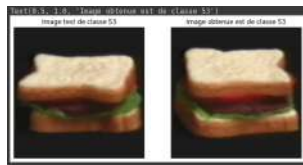


FIGURE 3 – Image de même classe

Par le résultat obtenu, nous pouvons par la suite évaluer notre modèle par le changement de nos hyper-paramètres cités plus haut pour pouvoir conclure sur le comportement de notre modèle.

## 4 Évaluation

### 4.1 hyperparamètre seuil = 0.75 et $k = 10$

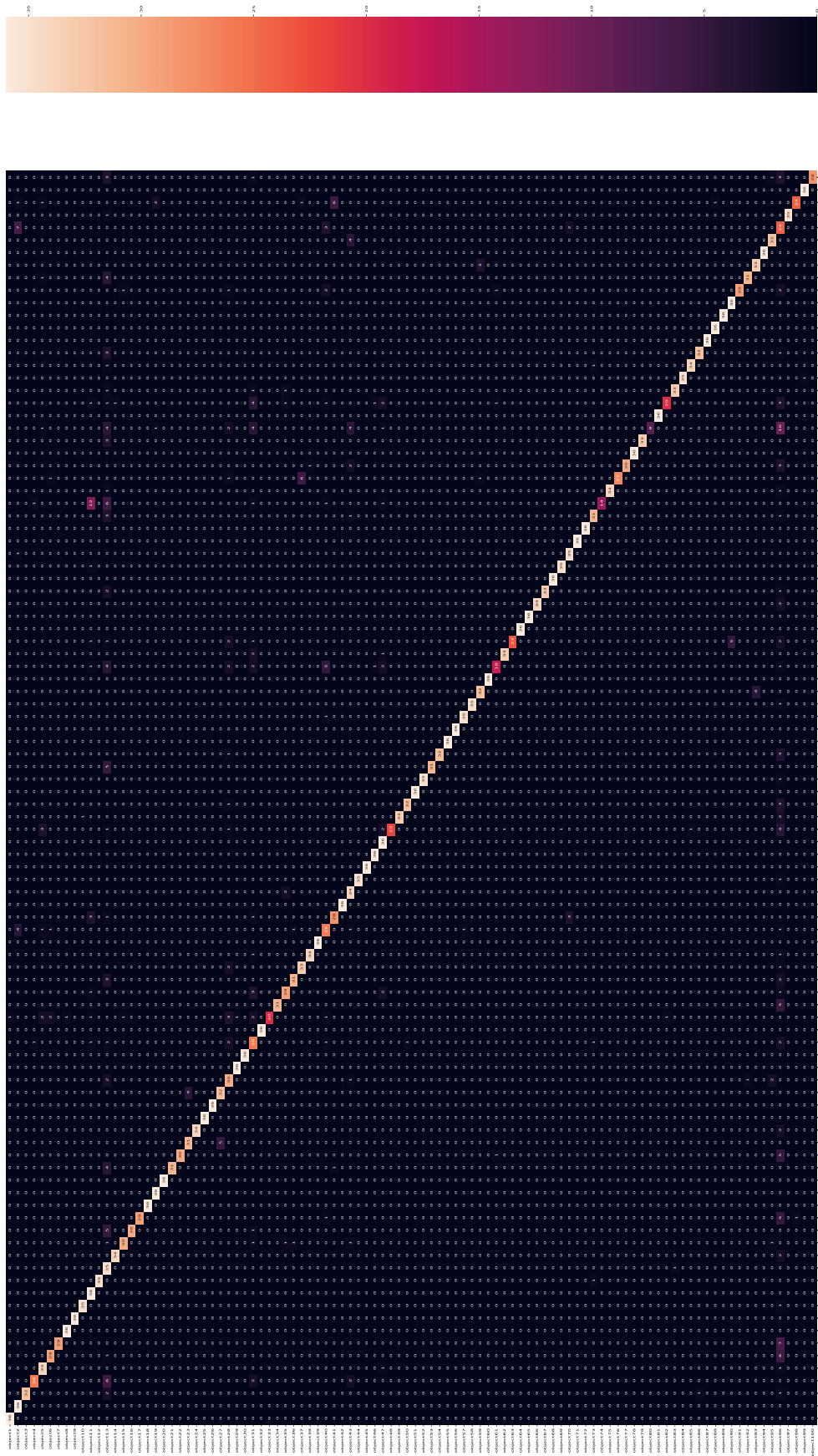


FIGURE 4 – Mtrice de confusion avec les hyper-paramètres seuil = 0.75 et  $k = 10$

Nous pouvons pas cette matrice de confusion évaluer notre modèle avec ces hyper-paramètres. Nous pouvons calculer le score de notre modèle de manière général et son score en éliminant les classes ayant un score inférieur à 75%. On obtient ainsi la figure ci-dessous :

```
Le score de notre modèle est de 89.38888888888889
Le score de notre modèle en éliminant les classe qui non pas un score de 75% est de :92.41666666666667
```

FIGURE 5 – Score de notre modèle avec ou sans celles qui ont moins de 75%

Car nous considérons que notre modèle est performant s'il affiche un score à plus de 75%. Nous pouvons ainsi afficher les classes ayant un score moins de 75%. Ce qui nous donne un total de 12 classes.

```
Classes ayant un score inférieur à 75% sont:
ject4': 26, 'object131': 26, 'object33': 26, 'object40': 26, 'object48': 22, 'object61': 18, 'object63': 23, 'object74': 14, 'object80': 8, 'object82': 26, 'object96': 24, 'object98': 24}
```

FIGURE 6 – Classes ayant un score moins de 75%

## 4.2 hyperparamètre seuil = 0.8 et k = 10

Nous pouvons ainsi nous demander comment améliorer notre modèle. Pour cela, nous allons augmenter le seuil à 0.8 à fin d'analyser. Nous remarquons une baisse du score de notre modèle, ce qui signifie qu'en matchant nos descripteurs en terme de distance ou en se rapprochant d'une valeur proche de la distance, on perd en terme de performance pour notre modèle.

```
Le score de notre modèle est de 85.94444444444447
Le score de notre modèle en éliminant les classe qui non pas un score de 75% est de :89.94444444444446
```

FIGURE 7 – Score de notre modèle avec ou sans celles qui ont moins de 75%

```
Le nombre de classes ayant un score inférieur à 75% est de :18
```

FIGURE 8 – Nombre de classe à moins de 75%

## 4.3 hyperparamètre seuil = 0.7 et k = 10

Nous allons essayer de modifier la valeur seuil tout en la diminuant à 0.7, on pourra voir les nouveaux changements. Nous observons une très grande augmentation du score de notre modèle.

```
Le score de notre modèle est de 91.47222222222224
Le score de notre modèle en éliminant les classe qui non pas un score de 75% est de :95.52777777777781
```

FIGURE 9 – Score de notre modèle avec ou sans celles qui ont moins de 75%

Ce qui montre qu'en utilisant un seuil de similarité par trop proche de la valeur une lors du matching de nos descripteurs, on n'améliore la qualité de performance de notre modèle car nous n'avons plus que 7 classes qui n'ont pas de bon score.



```
Les classes ayant un score inférieur à 75% sont:  
{'object4': 25, 'object33': 20, 'object48': 24, 'object61': 21, 'object74': 18, 'object80': 15, 'object82': 19}
```

FIGURE 10 – Classes ayant un score moins de 75%

#### 4.4 hyperparamètre seuil = 0.7 et k = 100

Nous pouvons aussi penser à l'hyper-paramètre k qui représente le nombre de k voisins. Nous allons l'augmenter exponentiellement c'est-à-dire le mettre à 100 afin d'évaluer le comportement de notre modèle.

```
Le score de notre modèle est de 75.4166666666667  
Le score de notre modèle en éliminant les classe qui non pas un score de 75% est de :80.25000000000001
```

FIGURE 11 – Score de notre modèle avec ou sans celles qui ont moins de 75%

```
Le nombre de classes ayant un score inférieur à 75% est de :40
```

FIGURE 12 – Nombre de classes ayant un score moins de 75%

Nous observons qu'avec un seuil de 0.7 et d'un k égale à 100, on constate un score totalement au rabais et le nombre de classe ayant un score inférieur est de 40. Ce qui montre que ces hyperparamètres rend notre modèle très moins performant.

## 5 Conclusion

Nous pouvons dire que ce travail nous permis de créer un modèle afin de pouvoir faire de la classification. L'évaluation de ce modèle permet de dire que les facteurs de performance dépendent du **seuil** et de **k**. Nous pouvons qu'avec le seuil compris entre **0.7 et 0.75** et k compris entre **5 et 10**, on pourrait avoir un modèle assez performant pour la classification.

L'accès au code source se fait en par un clic ici

## Références

- [1] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999.
- [2] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2) :91–110, 2004.