

UNIVERSITÉ NATIONALE DU VIETNAM À HANOÏ
INSTITUT FRANCOPHONE INTERNATIONAL



Option : Systèmes Intelligents et Multimédia (SIM)

Promotion : XXI

Reconnaissance des formes

Études et expérimentation de la classification des scènes naturelles

Rapport final

DIALLO Azise Oumar

KAFANDO Rodrique

RAKOTOARIVELO A. Nobby

Encadrant :

Dr HO Tuong Vinh, Professeur (VNU, IFI)

Année académique 2016-2017

Table des matières

1	Introduction générale	2
2	État de l’art	2
2.1	Les scènes naturelles et leurs caractéristiques	2
2.1.1	Définition des scènes naturelles	2
2.1.2	Caractéristiques des scènes naturelles	3
2.2	Les modèles de reconnaissance des scènes	3
2.2.1	Reconnaissance d’une scène en tant qu’ensemble d’objets	3
2.2.2	Reconnaissance d’une scène sur la base de ses caractéristiques propres . .	3
2.3	Présentation de quelques approches ou méthodes utilisées par les systèmes de reconnaissance des scènes naturelles	3
2.3.1	La modélisation hiérarchique bayésienne (naive baye)	4
2.3.2	Les méthodes de deep learning : CNN	4
2.3.3	Le codage par motifs fréquents pour la classification des scènes naturelles	5
2.3.4	La classification à partir de l’analyse en composantes indépendantes (ACI)	5
2.4	Conclusion	6
3	Solution proposée (Conception)	6
3.1	Problématique	7
3.2	Présentation de la base de données d’images	7
3.3	Proposition de solution	8
3.4	Outils de réalisation	10
3.5	Évaluation du système de reconnaissance	11
3.6	Conclusion	11
4	Implementation	11
4.1	Préparations des bases de données TRAIN et TEST	11
4.2	Implémentation du programme avec l’approche Bag-Of-Words	11
5	Expérimentations et analyse des résultats	12
5.1	Expérimentations	12
5.2	Analyse des résultats obtenus	16
6	Conclusion générale	17
6.1	Difficultés rencontrées	17
6.2	Perspectives	17
	Références bibliographiques	19

1 Introduction générale

Dans [1], la reconnaissance des formes est présentée comme une discipline qui regroupe toutes les activités liées à la reproduction ou à l'imitation de la perception humaine par un système artificiel, principalement en vue de l'automatiser.

De nos jours, plusieurs travaux de recherches essaient de reconnaître automatiquement les scènes naturelles. Cependant, cette tâche n'est pas toujours aisée, voir difficile à réaliser au regard de la nature des images des environnements naturels.

C'est dans cette optique que situe notre deuxième projet de reconnaissance de forme sous le thème « Études et expérimentation de la reconnaissance des scènes naturelles ».

Le présent document est le rapport final de notre projet. Il résume l'ensemble des travaux réalisés durant ce projet.

2 État de l'art

L'état de l'art est une phase importante dans toute étude. Dans cette partie, nous allons présenter la notion de scène naturelle et ses caractéristiques. Ensuite, nous évoquerons les méthodes ou approches de reconnaissance des scènes naturelles. Enfin, nous présenterons quelques travaux utilisant ces méthodes.

2.1 Les scènes naturelles et leurs caractéristiques

Notre environnement est composé de plusieurs objets. Ces objets peuvent être de formes variées, de taille différente, respectant certaines règles physiques. Cependant, en absence totale de lumière (exemple le soleil, les lampadaires...), leurs existences visuelles nous seraient inconnues. En effet, c'est uniquement grâce à la présence d'une source de lumière que nous percevons le monde qui nous entoure uniquement. A quoi renvoi cette notion de scènes naturelles et quelles sont ses caractéristiques ?

2.1.1 Définition des scènes naturelles

Notre environnement est la représentation la plus fidèle de ce que on pourrait décrire comme scène naturelle. Dans [2], il est établi qu'un seul exemplaire de scène naturelle équivaldrait à plus de 1000 mots. Lorsqu'on parle de scène naturelle, on pense directement à ce qui est dans ou appartient à la nature, à ce qui n'est pas le produit d'une pratique humaine (exemple les montagnes, la forêt...). Cependant, dans notre cas d'étude, la notion de « naturel » va au-delà de cet entendement. En effet, nous allons considérer une autre notion de la nature qui est « l'existence dans la réalité ». Ainsi, nous pouvons à travers cette définition prendre en compte tout ce qui est produit par l'homme (la rue, les chambres de nuit...).

2.1.2 Caractéristiques des scènes naturelles

En Introduction, nous avons vu que les scènes naturelles ne sont perceptibles que lorsqu'elles sont mises sous une source lumineuse. Ainsi, les images des scènes naturelles présentent certaines caractéristiques physiques liées à la prise photographique. Il s'agit notamment des couleurs, de la luminance, de la contraste (globale et locale). Il est important de noter que nous parlons ici de caractéristiques « perceptives » et non « sémantiques ». La sémantique décrit la nature et la fonction des objets contrairement à la perception qui correspond à la forme et à la géométrie.

2.2 Les modèles de reconnaissance des scènes

Une scène peut être définie comme un décor ou un cadre. Elle peut donc regrouper plusieurs objets ayant des caractéristiques données. Ainsi, il est possible de reconnaître une scène à partir des objets qui la compose ou à partir de ses caractéristiques propres.

2.2.1 Reconnaissance d'une scène en tant qu'ensemble d'objets

Plusieurs recherches [3, 4, 5] ont porté sur la reconnaissance visuelle des scènes en les décrivant comme une suite de traitements ascendants de plus en plus complexes des différents objets constituant la scène. De l'identité des objets découlerait la reconnaissance de la scène dans son ensemble, son contexte absolu, son environnement.

2.2.2 Reconnaissance d'une scène sur la base de ses caractéristiques propres

Si les premières recherches ont porté sur une analyse locale à travers les objets qui composent la scène, les travaux actuels sont plutôt orientés vers une analyse globale. Cette analyse est basée sur les caractéristiques physiques (bas-niveau). Voir notamment [6, 7, 9, 10].

2.3 Présentation de quelques approches ou méthodes utilisées par les systèmes de reconnaissance des scènes naturelles

Les systèmes de reconnaissance des formes en général ou la reconnaissance des scènes naturelles en particulier se basent sur plusieurs méthodes ou approches. En effet, ces systèmes ne font pas exception dans les méthodes d'apprentissage supervisé. Ils se basent donc sur les même algorithme d'apprentissage. Ainsi, les méthodes rencontrées lors de notre étude sont entre autre :

- La Modélisation hiérarchique bayésienne. Cette approche est utilisée dans les travaux présentés dans [11].
- L'analyse en composante principale et l'analyse en composante indépendante. L'ACI est utilisée dans [22, 23]
- Les méthodes deep learning à savoir les CNN. La reconnaissance de scène avec les CNN est implémentée dans [14].
- Le codage par motifs fréquents. Cette nouvelle approche est utilisée par A. Cornuéjols et al. dans [19].
- Les sacs de mots ou Bags of words (BOW). Cette approche est utilisée dans [25].

- La correspondance pyramidale spatiale ou spatial pyramid matching (SPM). C’est une approche similaire aux BOW. Cette technique est utilisée par Svetlana Lazebnik et al [8].
- Les méthodes hybrides qui font la combinaison de plusieurs algorithmes. Voir notamment [24].

Par la suite, nous allons faire la présentation de quelques travaux qui à notre connaissance offrent de meilleurs résultats dans les phases d’expérimentations.

2.3.1 La modélisation hiérarchique bayésienne (naive baye)

Dans [11], Li Fei-Fei et Pietro Perona proposent une nouvelle approche pour apprendre et reconnaître les catégories de scène naturelle. En effet, il propose un modèle hiérarchique bayésien pour apprendre et reconnaître les catégories de scène naturelle. Pour ce faire, les auteurs représentent l’image d’une scène par une collection de régions locales (reconnaissance à partir d’ensemble d’objets), désignées comme mots de code (bag of words) obtenus par un apprentissage non supervisé. Chaque région est représentée comme faisant partie d’un «thème». Il faut signaler que dans les travaux précédents, ces thèmes ont été appris à partir d’annotations manuscrites d’experts, alors que cette méthode apprend les distributions thématiques ainsi que la répartition des mots de code sur les thèmes sans supervision. Dans le cas de la reconnaissance des scènes naturelles, les acteurs ont utilisé la base « 13 Natural Scene Categories » (nous reviendrons plus en détails sur la base d’images dans la Section 2.1). De plus, pour l’apprentissage, chaque catégorie de scènes a été divisée de manière aléatoire en deux ensembles distincts d’images, 100 pour l’apprentissage (base TRAIN) et le reste pour les tests (base TEST). Le taux de performance est de l’ordre de 82,3%. Si les résultats sont assez acceptables pour la reconnaissance de scènes naturelles de l’extérieur, ce n’est pas le cas pour les scènes intérieures. Ainsi, on peut dire que le modèle n’est pas complet. C’est pour cela, les auteurs proposent l’utilisation de fonctions plus riches : en utilisant différentes astuces ainsi qu’une hiérarchie de mots-clés.

2.3.2 Les méthodes de deep learning : CNN

Les réseaux de neurones convolutifs (CNN) sont reconnus comme étant l’approche d’apprentissage en profondeur la plus réussie et largement utilisée (voir notamment [12]). De nos jours, cette méthode est devenue la plus dominante dans la majorité des tâches de reconnaissance et de détection [13, 15, 16, 17]. Le CNN est une architecture multi-couches à inspiration biologique composée de couches convolutives, de mise en commun et entièrement connectées. Elle peut être formée de façon supervisée. Il existe plusieurs architectures de CNN parmi lesquelles on peut citer : AlexNet, CaffeNet, VGGNet, VGG-VD Networks et PlacesNet. Dans [14], les auteurs ont utilisé deux bases d’images à savoir « UC Merced Land Use Dataset » et « WHU-RS Dataset ». Les résultats des performances sont présentés dans le tableau ci-dessous (tab1).

CNN	UCM(%)	WHU-RS(%)
AlexNet	94.37	93.81
CaffeNet	94.43	94.54
VGG-F	94.35	95.11
VGG-M	94.48	94.98
VGG-S	94.60	95.46
VGG-VD16	94.07	94.35
VGG-VD19	93.15	94.36
PlacesNet	91.44	91.73

Tab1 : Résultats des expérimentations avec les modèle CNNs

Bien vrai que cette méthode est performante, elle présente un inconvénient majeur qui la nécessité d'utiliser un grand nombre de données pour avoir une bonne performance. Ensuite, le temps de traitement peu etre très élevés.

2.3.3 Le codage par motifs fréquents pour la classification des scènes naturelles

Dans [19, 20], les acteurs proposent une méthode pour répondre au problème de description des données de grandes dimensions. En effet, ils présentent une nouvelle méthode de codage de données, le codage clairsemé par motifs fréquents (Frequent Item Sets for Independant Component Analysis : FISICA). Cette méthode est particulièrement adaptée à l'analyse de données en grandes dimensions. Pour illustrer son fonctionnement, deux domaines ont été choisis à savoir la reconnaissance de scènes naturelles et la reconnaissance de chiffres manuscrits. La méthode de classification se fait en deux phases à savoir : une étape de détermination d'une base de fonctions de base permettant de redécrire les données et phase d'emploi (utilisation) du système de codage ainsi obtenu pour classer de nouvelles formes. Pour cette dernière phase, les acteurs ont utilisé la méthode de classification de l'algorithme du plus proche voisin (kNN).

Dans le cas de la reconnaissance des scènes naturelles, les acteurs ont utilisé les images provenant de la base COREL¹. Les images sont redécrites par 128x128 pixels en 128 niveaux de gris. Pour ces expériences, la base utilisée comportait 1082 images dont la moitié a servi pour les fonctions de base et l'autre moitié pour les tests. Le taux de performance est de l'ordre de 80% en utilisant des fonctions de base de taux de couverture égale à 5% sous contrainte de connexité, en utilisant une formule d'appariement continu entre les images et les fonctions de base. Si les résultats sont acceptables pour la reconnaissance de scènes naturelles, ce n'est pas le cas pour les chiffres manuscrits. En effet, les résultats sont plus médiocres pour la reconnaissance des chiffres. Ainsi, une des limites de cette méthode est sa dépendance avec la nature des images traitées.

2.3.4 La classification à partir de l'analyse en composantes indépendantes (ACI)

L'ACI est une méthode statistique qui traite des observations vectorielles (multivariées) afin d'en extraire des composantes linéaires aussi indépendantes que possible. C'est une extension de

1. (http://www.corel.com/gallery_line/)

l'analyse en composantes principales (ACP). En fait, si l'ACP cherche à extraire des variables décorréliées, en se limitant à imposer la contrainte d'indépendance aux statistiques d'ordre deux (matrice de covariance), l'ACI en revanche, cherche l'indépendance des statistiques d'ordre supérieur à deux des variables mesurées [21]. Le but principal de l'ACI est de réduire donc la redondance des données d'entrée afin d'en avoir une présentation interne la plus efficace possible. L'application de l'ACI sur les images nous permet d'obtenir des filtres qui seront utilisés pour l'extraction des caractéristiques. Ces filtres permettent donc à la méthode de s'adapter quelque soit la nature de l'image. Dans [22], cette méthode a été mise en place lors d'une étude pour la classification d'une base de données de 540 images. Parmi ces images, 200 ont été utilisées pour l'extraction des filtres ACI par catégories, 50 de ces 200 sont utilisées pour extraire les filtres de nature de toutes catégories et les 340 ont été utilisées pour la validation du modèle. Dans cette étude, l'algorithme du plus proche voisin (kNN) a été utilisé pour la classification en se basant sur le calcul de la distance euclidienne entre les pixels des images. L'auteur a obtenu un taux performant de 87.2% avec 100 filtres (200 images) ACI en combinaison de deux autres filtres (Butterworth+Haming).

En conclusion, nous retenons que l'ACI est une méthode performante du faite qu'elle se base sur l'extraction des filtres, mais ses principaux défis restent le bon choix du nombre de filtres et la dimension de l'espace de travail.

2.4 Conclusion

La recherche bibliographique est une phase importante dans toute étude. Dans cette partie, nous avons mis la lumière sur la notion de scène naturelle, ses caractéristiques et les principaux modèles utilisés pour leur reconnaissance. Nous avons vu qu'il existe principalement deux modèles utilisés à savoir la considération de la scène comme ensemble d'objets (local) et sa considération comme caractéristiques propres (globales). De nos jours, c'est cette dernière approche qui est plus fréquente.

Par la suite, nous avons vu quelques méthodes ou approches utilisées par les systèmes de reconnaissance de scène naturelles et les travaux qui s'en ont basés. A ce niveau, il ressort que les méthodes de deep learning (CNN), de motifs fréquents, d'analyse en composantes indépendantes (ACI) et de Baye naïf permettent d'obtenir de bons résultats.

Ainsi, en se basant sur ces recherches et travaux, nous pouvons donc doré et déjà percevoir des pistes de solutions pour la résolution de notre problème dans les prochaines étapes.

3 Solution proposée (Conception)

Le choix d'une solution de reconnaissance de forme dépend d'une part de la nature des images à traiter, et d'autre part, de l'objectif recherché ou des résultats attendus du système à concevoir. Dans les lignes qui suivent, nous allons faire la description de la base de données d'images et la proposition d'une solution pour la reconnaissance des scènes naturelles.

3.1 Problématique

Tout système de reconnaissance de scènes cherche à répondre avec la plus grande précision à la problématique suivante : *étant donné un ensemble d'images de scènes naturelles, la tâche consiste à dire quelle image caractérise une scène de l'intérieur et quelle image caractérise une scène de l'extérieur parmi plusieurs images stockées dans une base de données. Ainsi, cette opération peut être mise en oeuvre sur un ensemble d'images issues de diverses scènes à travers une base TEST.*

La reconnaissance des scènes naturelles pose de nombreux défis. Ainsi, de tels systèmes doivent pouvoir s'affranchir des problèmes suivants :

- Les conditions d'illumination : l'intensité et la direction d'éclairage lors de la prise de vue influent énormément sur l'apparence du visage dans l'image. En effet, dans la plupart des applications courante, des changements dans les conditions d'éclairage sont inévitables, notamment lorsque les vues sont collectées à des heures différentes, en intérieur ou en extérieur.
- La configuration spatiale des objets similaires dans des scènes de classes différentes. Par exemple, nous pouvons noter la présence de d'un fauteuil dans un salon qui peut exister aussi dans une chambre de même que les couleurs de fond.

3.2 Présentation de la base de données d'images

La qualité ou la performance d'un système de reconnaissance de formes dépend fortement de la nature des images de la base de données et/ou du système d'acquisition d'image (caméra). Ainsi, il est primordial de faire une analyse préalable des images de la base avant de penser à la solution à mettre en place. Dans le cadre de notre étude, les images proviennent de *Stanford Vision Lab*² et notre la base est « 13 Natural Scene Categories ». C'est une base qui a été construite par les auteurs dans [11].

Cette base contient 13 dossiers représentant 13 catégories de scènes naturelles (Figure 1). Chaque dossier contient à son tour des centaines d'images différentes correspondant à la nature de la scène : highway (260 images), inside of cities (308 images), tall buildings (356 images), streets (292 images), suburb residence (241 images), forest (328 images), coast (360 images), mountain (374 images), open country (410 images), bedroom (174 images), kitchen (151 images), livingroom (289 images) and office (216 images). La taille moyenne de chaque image est d'environ 250x300 pixels, avec 256 niveaux de gris par pixel. Les images sont en format .JPG.

2. http://vision.stanford.edu/resources_links.html

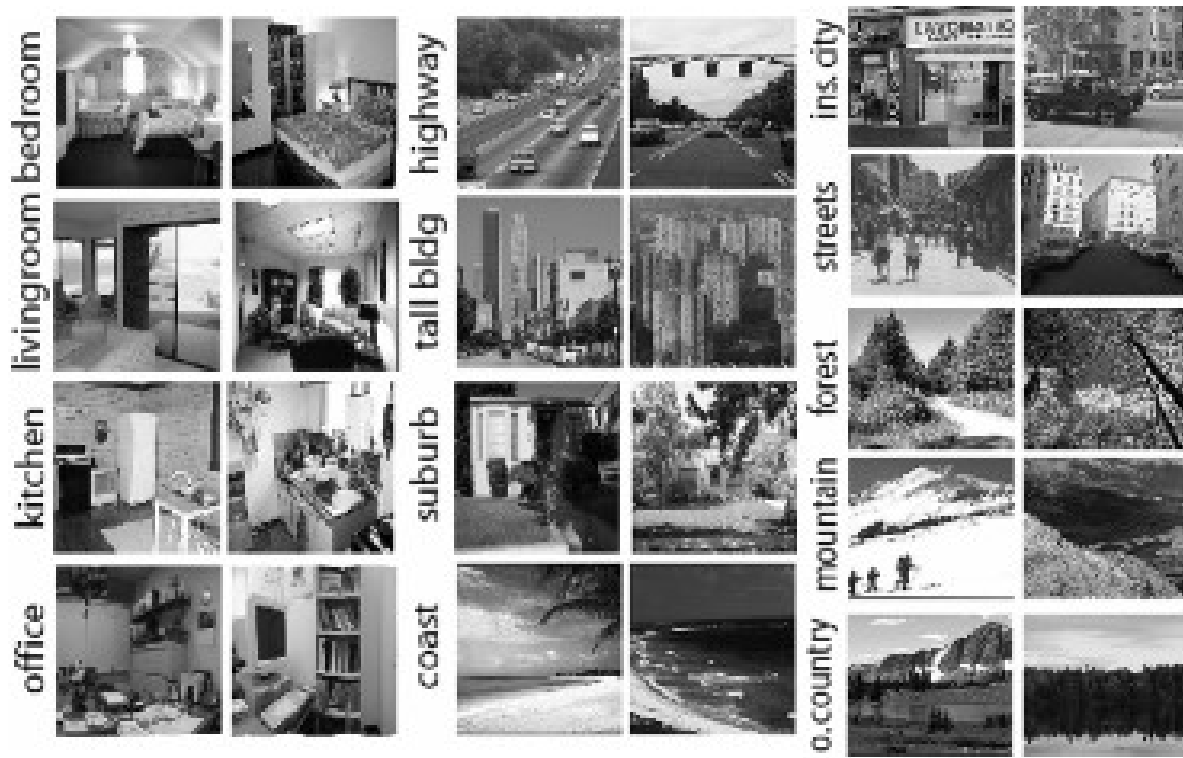


Figure 1 13 Natural Scene Categories

L'analyse de cette base appelle quelques commentaires :

- les images contiennent des objets propres et des caractéristiques propres à chaque catégorie de scène ;
- les images sont déjà en niveaux de gris qui est l'espace approprié pour le traitement sur les images. Ainsi, des les étapes de pré-traitement, nous aurons pas faire cette conversion ;
- chaque catégorie de scène dispose de plusieurs images (minimum 150) différentes avec différentes avec divers contenus. Cette diversité permet de construire un système d'apprentissage robuste.

3.3 Proposition de solution

Nous avons vu en Section1 (État de l'art), qu'il existait deux modèles dans la reconnaissance des scènes naturelles : l'approche globale en considérant les caractéristiques propres des images et l'approche locale à travers les objets propres constituant l'image. Nous avons vu également que les systèmes de reconnaissance de forme utilisent diverses méthodes ou algorithmes d'apprentissage automatique (Modélisation hiérarchique bayésienne, CNN, SVM, ACI...) pour la classification des images.

La reconnaissance automatique des images s'effectue en deux étapes principales : (1) extraction et normalisation des caractéristiques de la scène, (2) identification et/ou vérification pour la classification.

La reconnaissance des scènes pose de nombreux défis car les scènes naturelles sont composées d'objets déformables 3D. Cependant, dans le cadre du présent travail, nous nous limitons à une

reconnaissance à partir d'une image 2D en environnements non contraints.

Dans le cadre de notre étude, nous allons utiliser une approche locale. En effet, au regard de la nature des images de la base, nous allons utiliser les objets caractéristiques de chaque image pour constituer un dictionnaire de mot (bag of words) comme présenté dans [11]. Pour ce faire nous allons subdiviser le travail en deux phases :

- une étape d'apprentissage pour construire un modèle apprentissage automatique ;
- une étape de test qui va servir à tester et valider le modèle obtenu (voir Figure2).

A cet effet, nous avons subdivisé notre base d'images conformément aux exigences du projet c'est à dire en deux bases : TRAIN et TEST. La base TRAIN contient les 100 premières images de chaque catégorie de scène (1300 images). Le reste des images est sauvegardé dans le dossier TEST.

Étape d'apprentissage

C'est dans cette étape que nous allons construire le « bag of words » qui va regrouper l'ensemble des caractéristiques des images (scène) présentes dans la base TRAIN. C'est une méthode qui permet de considérer l'image comme étant un ensemble de régions (zones d'intérêts). Il s'agit donc d'une approche locale. nous permet de constituer un ensemble un dictionnaire de mots de toutes les images dans un seul sac. En effet, ce dictionnaire de mots représente l'ensemble des caractéristiques ou les points clés que compose chaque image. Afin de pouvoir extraire ces caractéristiques, nous avons fait recoure au de scripteur SIFT. Le descripteur SIFT est une méthode qui permet de générer ou d'extraire les points clés de chacune des images. Son principe de fonctionnement est comme suit ;

- découpage de l'image en plusieurs blocs (keypoints) ;
- détermination des histogrammes d'orientation pour chaque bloc ;
- détermination l'orientation dominante (vecteurs caractéristiques) pour chaque bloc ;
- regroupement des vecteurs caractéristiques pour la construction du dictionnaire

Une fois le dictionnaire élaboré, nous appliquons la méthode des kmeans sur l'ensemble des caractéristiques obtenues (dictionnaire). L'application du kmeans a pour but de réduire la dimension de l'espace caractéristique. En effet, il nous permet de générer des centroïdes qui regrouperons au mieux l'ensemble des caractéristiques convergeants à un point de vu commun.

Étape de test

Après l'étape d'extraction des caractéristiques, nous pouvons maintenant effectuer la classification des images. La phase de test, nous pouvons noter les étapes suivantes :

- l'extraction des caractéristiques locales des images à classifier ;
- l'agrégation des histogrammes issus du sac à mots ;
- la mise en correspondance du vecteur caractéristique de l'image requête avec l'ensemble des vecteurs caractéristiques de la base TRAIN grâce au classificateur (SVM) afin de prédire la classe (catégorie) de l'image.

La figure ci-dessous (Figure2) nous donne un schéma illustratif de la construction du modèle.

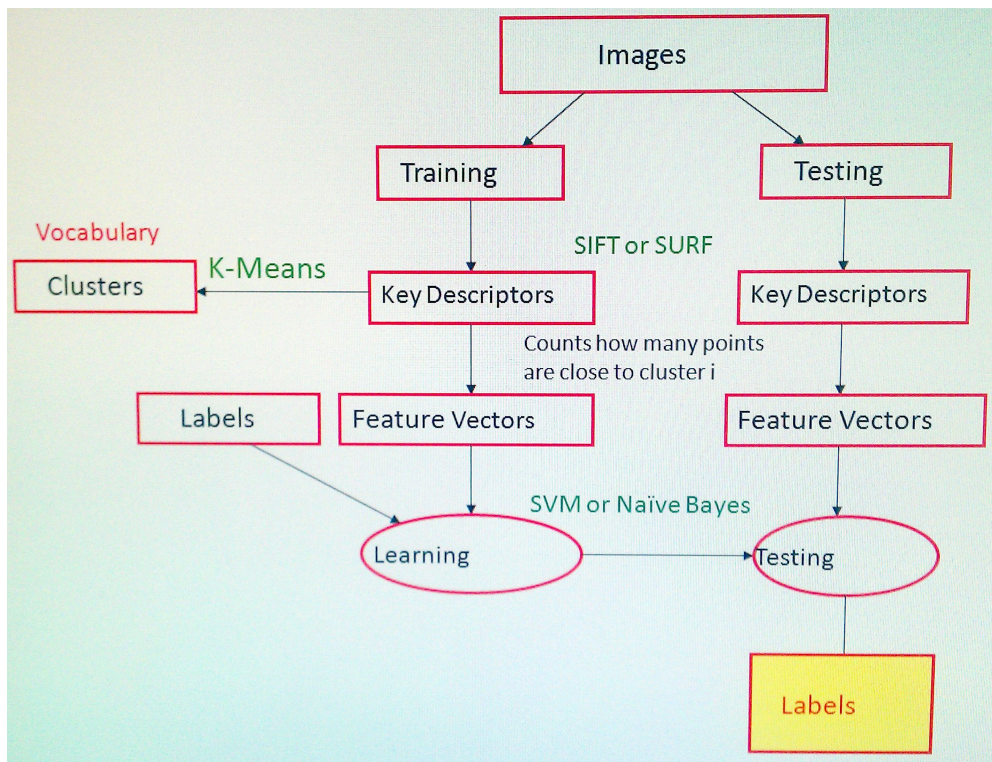


Figure 2 Architecture système de reconnaissance avec bag of words

3.4 Outils de réalisation

Le programme de reconnaissance de scènes sera codé en Python. Nous utiliserons pour ce faire les bibliothèques d'OpenCV notamment SIFT pour la détection et le calcul des points d'intérêts, les librairies sklearn (kmeans, SVM, évaluation) pour la construction du modèle. Le choix du langage Python n'est pas fortuit. En effet, en plus d'être un choix de l'équipe projet, ce langage présente de nombreux avantages parmi lesquels nous avons : la souplesse d'utilisation, l'intégration de nombreuses librairies de deep learning et sa compatibilité avec d'autres librairies telles que Opencv.

Pour le développement et les expérimentations nous allons utiliser un ordinateur portable avec les caractéristiques suivantes :

- Processeur : Intel(R) Core™ M i5-M370 @2.4 GHZ
- RAM : 8.00 Go
- OS : Linux Ubuntu 16.04 LTS
- Python 2.7 ou Python 3 comme langage de programmation ;
- Librairies : SciPy avec NumPy pour les opérations mathématiques, OpenCV et Pil pour les traitements d'images, Theano et Keras pour le deep learning.
- Jupyter notebook comme éditeur.

3.5 Évaluation du système de reconnaissance

Afin de tester la performance de notre système de reconnaissance, nous avons utilisé les images de la base TEST pour valider le modèle construit. Pour ce faire, nous construisons une matrice de confusion qui fera ressortir la capacité du système à classer les images de test. Ainsi, nous pourrions voir le classement des images dans chaque catégorie de scène. Par la suite, nous calculons quelques indicateurs tels que le taux de performance, le rappel et la précision.

3.6 Conclusion

Dans cette étape du projet, il s'agissait de proposer une solution de notre système de reconnaissance de scènes naturelles sur la base de recherche bibliographique (état de l'art) effectuée dans la première partie. Ainsi, notre solution s'appuie sur les résultats de plusieurs travaux de recherches.

Notre système utilise la méthode locale avec la technique de « bag of words » pour l'extraction des caractéristiques (feature vectors). Nous utilisons par la suite les SVM pour construire le modèle et faire la classification des images de la base TEST.

La solution ainsi proposée est loin d'être exhaustive. En effet, il se peut qu'au cours de l'implémentation quelques changements subviennent.

4 Implementation

4.1 Préparations des bases de données TRAIN et TEST

Conformément aux exigences du sujet, nous avons divisé la base d'images en deux parties à savoir une base TRAIN et une base TEST. La base TRAIN est composée des 100 premières images de chaque classe de scène. Quant à la base TEST, elle regroupe le reste des images de chaque classe n'ayant pas participé à la phase d'apprentissage.

1. La nomenclature des classes : Chaque classe d'image est déterminée par le nom de son répertoire. Le nom de la classe est composé du label qui la caractérise suivi du nom de la scène considérée : "*label*". "*nom_classe*". Exemple : BEDROOM.bedroom représente la scène bedroom précédée de son label BEDROOM qui la caractérise)
2. L'ensemble des images de chaque classe est récupéré dans un fichier *DatasetFile.txt* avec un script de façon automatique. Ce fichier contient respectivement dans sa première et deuxième colonnes le label (classe) et le chemin relatif d'accès (Path) de chaque image sous la forme "*label chemin_nom_image*".

Ex. : BEDROOM SceneClass13/BEDROOM.bedroom/image_0108.jpg.

4.2 Implémentation du programme avec l'approche Bag-Of-Words

Dans cette partie, nous présentons les étapes de la mise en oeuvre de notre système. Pour réaliser le modèle, nous avons créé quatre fichiers à savoir BoW.py, utils.py, classifier.py et outils.py :

1. `outils.py` :

C'est le script qui permet de récupérer dans un fichier.txt le label et le chemin d'accès de chacune des images.

2. `BoW.py` :

C'est le fichier principal de notre programme. Il fait appel à plusieurs méthodes ou fonctions définies dans les fichiers suivant :

3. `utils.py` :

C'est dans ce fichier que s'effectue :

- le chargement (lecture) de toutes les images contenues dans les deux bases à savoir la base TRAIN et TEST (fonction *load_data*) ;
- l'extraction des caractéristiques ou des keypoints par le de scripteur SIFT (fonction (fonction *extract_sift_descriptors*) ;
- la construction du dictionnaire (fonction *build_codebook*) ;

4. `classifier.py` :

Le fichier classifier nous permet d'effectuer la classification des images à partir des labels et des caractéristiques retenues dans le dictionnaire de données ou codebook. Il effectue les taches suivantes :

- la récupération des labels stockés dans le fichier labels.txt (fonction *get_label*).
- la mise en correspondance de chaque label avec la classe correspondante (fonction *svm_classier*) ;
- le calcul de la matrice de confusion (fonction *svm_classier*) ;
- le calcul du taux de précision, du rappel et du f1 (fonction *svm_classier*).

5 Expérimentations et analyse des résultats

Dans cette partie, nous faisons une présentation suivie d'une analyse critique des résultats obtenus lors des expérimentations que nous avons eu à effectuer.

5.1 Expérimentations

Afin de mieux comprendre le comportement de notre modèle, nous avons procédé à l'utilisation des données avec la répartition précédemment indiquée. Une base TRAIN composée des 100 premières images de chaque classe et une base TEST constituée de l'ensemble des images de chacune des bases n'ayant pas participé à la construction du modèle. Le dictionnaire de données que nous établissons à base des caractéristiques extraites par le SIFT joue un rôle très primordiale dans la phase de classification. Nous avons donc effectuée nos tests en faisant varier sa taille. Par les captures suivantes, nous présentons les résultats obtenus. De plus, les résultats sont résumés dans le tableau ci-dessous (Tab 2) :

- Premier TEST : VOC_SIZE= 50

	precision	recall	f1-score	support
BEDROOM	0.27	0.33	0.30	116
CALSUBURB	0.61	0.93	0.74	141
KITCHEN	0.28	0.34	0.30	110
LIVINGROOM	0.39	0.31	0.35	189
MITCOAST	0.58	0.58	0.58	260
MITFOREST	0.73	0.92	0.81	228
MITHIGHWAY	0.41	0.41	0.41	160
MITINSIDECI	0.62	0.50	0.55	208
MITMOUNTAIN	0.52	0.64	0.57	274
MITOPENCOUNT	0.60	0.34	0.43	310
MITSTREET	0.54	0.63	0.58	192
MITTALLBUILD	0.73	0.45	0.56	256
PAROFFICE	0.38	0.50	0.43	115
avg / total	0.55	0.54	0.53	2559

Figure 3a : Détail sur précision et recall par classe

```

Matrice de confusion.
[[ 38  3 15 20  0  2  6  2  8  2  6  2 12]
 [ 0 131  0  3  0  2  1  0  1  0  0  0  3]
 [ 12  3 37 25  0  0  0  6  0  0  4  0 23]
 [ 33 12 33 59  1  1  0  5  0  0 12  2 31]
 [ 5  5  0  0 152  3 21  1 38 25  2  8  0]
 [ 1  0  0  0  0 209  0  0 10  8  0  0  0]
 [ 5 10  4  0 36  1 66  7  5  5  7 13  1]
 [ 9  1 23  6  1  0  7 104  0  0 33  8 16]
 [ 2 13  1  0 19 29 16  0 174 18  2  0  0]
 [ 0 15  0  0 44 35 21  1  80 105  7  2  0]
 [ 13  3  3  5  2  0  7 15 13  3 121  6  1]
 [ 7 14  6  9  9  4 16 24  7 10 27 116  7]
 [ 14  4 11 23  0  0  0  2  0  0  2  1 58]]

Precision = 0.535
Le temps d'exécution est de : 4872.848404407501 secs

```

Figure 3b : Matrice de confusion

— Deuxième TEST : VOC_SIZE= 100

	precision	recall	f1-score	support
BEDROOM	0.31	0.34	0.33	116
CALSUBURB	0.72	0.88	0.79	141
KITCHEN	0.29	0.22	0.25	110
LIVINGROOM	0.42	0.43	0.43	189
MITCOAST	0.57	0.74	0.64	260
MITFOREST	0.71	0.90	0.79	228
MITHIGHWAY	0.55	0.38	0.44	160
MITINSIDECI	0.56	0.50	0.53	208
MITMOUNTAIN	0.59	0.70	0.64	274
MITOPENCOUNT	0.59	0.32	0.41	310
MITSTREET	0.51	0.72	0.60	192
MITTALLBUILD	0.68	0.39	0.50	256
PAROFFICE	0.36	0.42	0.38	115
avg / total	0.55	0.55	0.54	2559

Figure 4a : Détail sur précision et recall par classe

```

-----
Matrice de confusion.
[[ 40  2 12 25  5  3  2  1  3  2  5  5 11]
 [  0 124  0  5  0  7  0  0  1  0  3  0  1]
 [ 17  1 24 24  0  0  1  7  0  0  6  3 27]
 [ 26  7 14 82  1  1  0  5  0  3 18  3 29]
 [  2  3  1  0 193  2  9  1 22 21  2  4  0]
 [  1  0  0  0  0 205  0  0 12 10  0  0  0]
 [  5  4  1  0 49  1 60 13  1  6 10 10  0]
 [  5  1 20  6  2  1  2 105  0  0 39 12 15]
 [  2  7  0  0 15 29  8  0 191 18  3  1  0]
 [  2 12  0  1 50 37 17  1 84 99  5  2  0]
 [  5  2  3  8  1  0  2 13 10  4 139  5  0]
 [  3  8  4 16 23  2  8 37  2  6 42 101  4]
 [ 19  1  5 29  2  0  1  5  0  0  2  3 48]]
-----
Precision = 0.551
-----
Le temps d'exécution est de : 11957.769581079483 secs

```

Figure 4b : Matrice de confusion

— Troisième TEST : VOC_SIZE= 200

	precision	recall	f1-score	support
BEDROOM	0.34	0.39	0.36	116
CALSUBURB	0.67	0.89	0.77	141
KITCHEN	0.29	0.28	0.28	110
LIVINGROOM	0.39	0.44	0.41	189
MITCOAST	0.60	0.61	0.61	260
MITFOREST	0.72	0.90	0.80	228
MITHIGHWAY	0.55	0.42	0.48	160
MITINSIDECI	0.57	0.57	0.57	208
MITMOUNTAIN	0.57	0.74	0.64	274
MITOPENCOUNT	0.58	0.32	0.41	310
MITSTREET	0.51	0.69	0.58	192
MITTALLBUILD	0.74	0.41	0.53	256
PAROFFICE	0.46	0.43	0.45	115
avg / total	0.56	0.56	0.55	2559

Figure 5a : Détail sur précision et recall par classe

```

-----
Matrice de confusion.
[[ 45  1  9 28  1  3  2  3  2  4  8  2  8]
 [  0 126  0  5  0  5  0  1  1  1  1  0  1]
 [ 10  3 31 32  0  0  1  6  0  0  5  1 21]
 [ 34  8 23 83  0  1  1  7  0  0 14  2 16]
 [  6  4  0  0 159  4 18  0 29 29  3  8  0]
 [  0  1  0  0  0 205  0  0 13  8  1  0  0]
 [  2  2  7  1 34  1 68 11  5  5 11 11  2]
 [  6  1 17 11  0  0  1 118  1  1 35  9  8]
 [  1  8  0  0 12 22  8  0 203 15  5  0  0]
 [  1 11  0  0 55 38 10  1 86 99  8  1  0]
 [  3  5  4  7  0  3  3 14 12  3 133  4  1]
 [  9 16  5 13  3  1 11 44  4  6 37 106  1]
 [ 14  2 12 33  0  0  1  1  0  0  2  0 50]]
-----
Precision = 0.557
-----
Le temps d'exécution est de : 21507.786287784576 secs

```

Figure 5b : Matrice de confusion

— Quatrième TEST : VOC_SIZE= 500

	precision	recall	f1-score	support
BEDROOM	0.33	0.40	0.36	116
CALSUBURB	0.77	0.93	0.84	141
KITCHEN	0.34	0.34	0.34	110
LIVINGROOM	0.49	0.50	0.49	189
MITCOAST	0.61	0.70	0.65	260
MITFOREST	0.75	0.94	0.83	228
MITHIGHWAY	0.62	0.44	0.52	160
MITINSIDECI	0.59	0.55	0.57	208
MITMOUNTAIN	0.62	0.70	0.66	274
MITOPENCOUNT	0.61	0.41	0.49	310
MITSTREET	0.56	0.70	0.62	192
MITTALLBUILD	0.65	0.49	0.56	256
PAROFFICE	0.55	0.47	0.51	115
avg / total	0.59	0.59	0.59	2559

Figure 6a Détail sur précision et recall par classe

```

Matrice de confusion.
[[ 46  3  7 29  2  2  1  3  1  4  5  7  6]
 [  0 131  0  1  0  3  0  3  0  0  2  0  1]
 [ 16  3 37 19  0  0  0  8  0  0  7  1 19]
 [ 34  5 19 94  1  0  0  9  0  0 10  3 14]
 [  1  1  0  0 183  2  7  0 27 27  1 11  0]
 [  2  1  0  0  0 214  0  0  2  8  1  0  0]
 [  1  4  2  1 37  1 71  9  1  8  5 19  1]
 [  6  1 24  9  0  0  4 114  0  0 34 13  3]
 [  1  4  1  0 16 33  4  0 191 21  2  1  0]
 [  2  9  0  1 51 30 10  1 75 126  3  2  0]
 [  7  2  1  8  1  1  6  7  9  6 135  9  0]
 [  6  6  3 11 10  1 12 35  3  7 36 126  0]
 [ 18  1 16 18  0  0  0  4  0  0  1  3 54]]

Precision = 0.595
Le temps d'exécution est de : 65213.385697841644 secs

```

Figure 6b Matrice de confusion

— Cinquième TEST : VOC_SIZE= 1000

	precision	recall	f1-score	support
BEDROOM	0.31	0.38	0.34	116
CALSUBURB	0.75	0.91	0.83	141
KITCHEN	0.34	0.34	0.34	110
LIVINGROOM	0.46	0.56	0.50	189
MITCOAST	0.62	0.74	0.67	260
MITFOREST	0.73	0.92	0.81	228
MITHIGHWAY	0.66	0.50	0.57	160
MITINSIDECI	0.63	0.61	0.62	208
MITMOUNTAIN	0.60	0.72	0.65	274
MITOPENCOUNT	0.64	0.37	0.47	310
MITSTREET	0.56	0.68	0.61	192
MITTALLBUILD	0.68	0.45	0.54	256
PAROFFICE	0.56	0.41	0.47	115
avg / total	0.60	0.60	0.59	2559

Figure 7a : Détail sur précision et recall par classe


```

Matrice de confusion.
[[ 44  3 11 29  3  2  2  3  3  0  6  5  5]
 [ 0 129  0  3  0  5  0  2  0  0  1  0  1]
 [13  1 37 32  0  0  0  8  0  0  5  2 12]
 [34  7 16 105  1  0  1  6  0  0  7  2 10]
 [ 1  2  0  0 193  2  6  1 22 24  1  8  0]
 [ 0  0  0  0  0 210  0  0  7 11  0  0  0]
 [ 3  1  1  2 34  1 80  8  4  5  7 14  0]
 [ 7  2 15 10  0  1  1 126  0  0 26 14  6]
 [ 2  3  0  1 16 32  5  0 196 16  3  0  0]
 [ 3  9  0  0 55 34  9  0 80 115  3  2  0]
 [ 8  4  2 15  0  1  6  9 10  2 130  5  0]
 [ 5  8  3 17  9  1 11 33  4  6 42 114  3]
 [23  2 23 15  1  0  0  3  0  0  0  1 47]]

Precision = 0.596
Le temps d'exécution est de : 95032.04631090164 secs

```

Figure 7b : Matrice de confusion

VOC_SIZE	Précision moyenne (%)	temps d'exécution (h)
50	53	1.35
100	55	3.32
200	56	5.97
500	59	18.11
1000	60	26.40

Tab2 : Résultats des expérimentations avec les modèle CNNs

5.2 Analyse des résultats obtenus

Les résultats ci-dessus nous montrent en plus de la matrice de confusion, d'autres indicateurs tels que le rappel et la précision fournies pour chaque classe de scène pour chaque "codebook" utilisée. De plus, nous avons les temps moyens d'exécution correspondant. L'analyse des résultats appelle les commentaires ci-après :

- Les scènes naturelles de l'extérieur (CALSUBURB, MITFOREST, MITCOST, MITHIGWAY, MITTALLBUILDING, MITOPENCOUNTRY, MITMOUNTAIN, MITINSIDE-CITY, MITSTREET)) ont une meilleure reconnaissance par rapport à celles de l'intérieur (BEDROOM, KITCHEN, PAROFFICE, LIVINGROOM). Ce résultat s'explique principalement par la bonne qualité des conditions d'illumination. En effet, les scènes de l'extérieur bénéficient d'un meilleur éclairage (soleil) que les images de l'intérieur généralement peu éclairées (lampes éclectiques). De plus, les images de l'extérieur présentent moins d'objets de divers caractéristiques (couleur, forme...) contrairement à celles de l'intérieur qui sont généralement composées de plusieurs objets divers et de couleurs et formes variées (tables, chaises, fauteuils..).
- La performance du système s'améliore légèrement lorsque le nombre de clusters augmente. En effet, lorsque VOC_SIZE = 50 nous avons une précision moyenne de 53% contrairement à 60% lorsque VOC_SIZE = 1000. Cela pourrait bien s'expliquer du fait que la taille du dictionnaire de caractéristiques ou de mots représentent les points clés de chaque image de la base TEST conçue à partir de l'ensemble des classes. Plus le nombre de mots clés extraits de chaque image est élevé plus l'image est bien décrite donc une grande quantité

d'information sur l'image. Réciproquement, plus le nombre de mots clés est petit, plus on a moins d'informations sur l'image induisant ainsi une mauvaise prédiction lors de la phase de classification.

- Le temps d'exécution est également un critère d'évaluation très important. En effet, un bon système est celui qui fournit de bons résultats dans un temps relativement bref. Le temps d'exécution dépend non seulement du matériel mais aussi des méthodes et algorithmes utilisés. Ainsi, dans notre cas présent, nous avons remarqué que le temps d'exécution est plus élevé lorsque la taille du "codebook" est important.

6 Conclusion générale

Pour ce deuxième projet, nous avons effectué une étude et une expérimentation de la reconnaissance des scènes naturelles. La base d'images utilisée dans ce projet est "13 naturel scene categories". Pour ce faire, nous avons construit un modèle à partir d'une base d'apprentissage constituée des 100 premières images de chaque catégorie. Le modèle ainsi construit utilise une approche locale notamment les points ou zones d'intérêt avec la méthode SVM comme classifieur.

Pour l'évaluation de notre système, nous avons construit la matrice de confusion, calculé les indicateurs tels que la précision, le rappel et le f1 pour chacune des classes. En outre, nous avons calculé le temps d'exécution de notre système. Pour ce faire, nous avons utilisé la base de TEST construite à cet effet.

Nous avons effectué plusieurs expérimentations afin d'optimiser les performances du système. A cet effet, nous avons changé principalement la taille des clusters. Les résultats obtenus ont permis de montrer que plus le nombre de clusters est grand, plus la précision du système s'améliore. Par contre le temps d'exécution augmente presque doublement avec la taille du "codebook". De plus, nous avons observé que les scènes de l'extérieur avaient une bonne reconnaissance comparativement à la scène de l'intérieur.

En somme, nous pouvons dire que ce projet a été très instructif pour nous. En effet, il nous a permis de renforcer nos connaissances théoriques et pratiques dans la reconnaissance de formes et en particulier la reconnaissance des scènes naturelles.

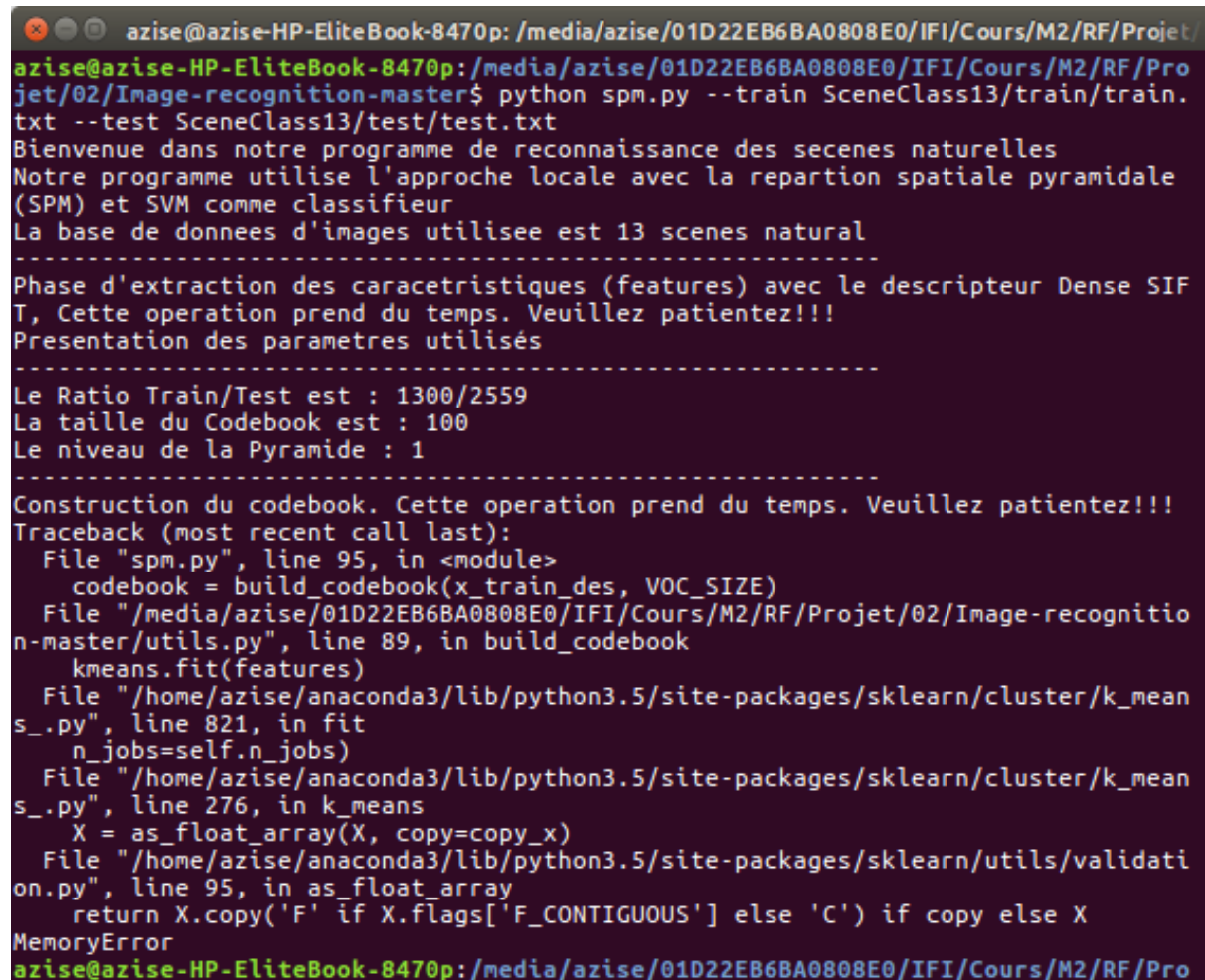
6.1 Difficultés rencontrées

la principale difficulté que nous avons rencontrée était le temps d'exécution. En effet, dans la recherche de meilleurs paramètres pour optimiser notre système, nous avons voulu effectuer plus d'expérimentations. Malheureusement, nous avons été limitée lors de l'utilisation de grande valeur pour la taille du "codebook". Par exemple, lorsque `VOC_SIZE = 1000`, il a fallu plus d'une journée entière (26 heures) pour avoir les résultats.

6.2 Perspectives

Notre futur travail consistera à continuer l'implémentation de l'approche de la correspondance pyramidale spatiale (spatial pyramidal matching (SPM)) comme décrit dans [8]. En effet, nous

avons voulu mettre en oeuvre cette solution pour comparer les deux approches à savoir Bow et SPM. Malheureusement, nous avons rencontré des erreurs lors de la mise en oeuvre (voir figure ci-dessous, Figure 8).



```
azise@azise-HP-EliteBook-8470p: /media/azise/01D22EB6BA0808E0/IFI/Cours/M2/RF/Projet/02/Image-recognition-master$ python spm.py --train SceneClass13/train/train.txt --test SceneClass13/test/test.txt
Bienvenue dans notre programme de reconnaissance des scenes naturelles
Notre programme utilise l'approche locale avec la repartition spatiale pyramidale (SPM) et SVM comme classifieur
La base de donnees d'images utilisee est 13 scenes natural
-----
Phase d'extraction des caractéristiques (features) avec le descripteur Dense SIFT, Cette operation prend du temps. Veuillez patientez!!!
Presentation des parametres utilisés
-----
Le Ratio Train/Test est : 1300/2559
La taille du Codebook est : 100
Le niveau de la Pyramide : 1
-----
Construction du codebook. Cette operation prend du temps. Veuillez patientez!!!
Traceback (most recent call last):
  File "spm.py", line 95, in <module>
    codebook = build_codebook(x_train_des, VOC_SIZE)
  File "/media/azise/01D22EB6BA0808E0/IFI/Cours/M2/RF/Projet/02/Image-recognition-master/utils.py", line 89, in build_codebook
    kmeans.fit(features)
  File "/home/azise/anaconda3/lib/python3.5/site-packages/sklearn/cluster/k_means_.py", line 821, in fit
    n_jobs=self.n_jobs)
  File "/home/azise/anaconda3/lib/python3.5/site-packages/sklearn/cluster/k_means_.py", line 276, in k_means
    X = as_float_array(X, copy=copy_x)
  File "/home/azise/anaconda3/lib/python3.5/site-packages/sklearn/utils/validation.py", line 95, in as_float_array
    return X.copy('F' if X.flags['F_CONTIGUOUS'] else 'C') if copy else X
MemoryError
azise@azise-HP-EliteBook-8470p: /media/azise/01D22EB6BA0808E0/IFI/Cours/M2/RF/Projet/02/Image-recognition-master$
```

Figure 8 : Erreur mémoire lors de l'exécution du SPM

Références bibliographiques

Références scientifiques

- [1] Murat Kunt. *Reconnaissance des formes et analyse de scènes*, volume 3. PPUR presses polytechniques, 2000.
- [2] Alinda Friedman. Framing pictures : The role of knowledge in automatized encoding and memory for gist. *Journal of experimental psychology : General*, 108(3) :316, 1979.
- [3] H.G. Barrow and J.M. Tenenbaum. Interpreting line drawings as three-dimensional surfaces. *Artificial Intelligence*, 17(1) :75 – 116, 1981.
- [4] Irving Biederman. Recognition by components : A theory of human image understanding. *Psychological Review*, 94(2) :115 – 147, 1987.
- [5] D. Marr. *Vision : A Computational Investigation into the Human Representation and Processing of Visual Information*. W.H. Freeman and Co., New York, NY, 1982.
- [6] James R. Antes, James G. Penland, and Richard L. Metzger. Processing global information in briefly presented pictures. *Psychological Research*, 43(3) :277–292, 1981.
- [7] Geoffrey R. Loftus, Walter W. Nelson, and Howard J. Kallman. Differential acquisition rates for different types of information from pictures. *The Quarterly Journal of Experimental Psychology Section A*, 35(1) :187–198, 1983.
- [8] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features : Spatial pyramid matching for recognizing natural scene categories. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 2169–2178. IEEE, 2006.
- [9] A Martínez, L Anllo-Vento, M I Sereno, L R Frank, R B Buxton, D J Dubowitz, E C Wong, H Hinrichs, H J Heinze, and S A Hillyard. Involvement of striate and extrastriate visual cortical areas in spatial attention. *Nat Neurosci*, 2(4) :364–369, April 1999.
- [10] Philippe G. Schyns and Aude Oliva. From blobs to boundary edges : Evidence for time- and spatial-scale-dependent scene recognition. *Psychological Science*, 5(4) :195–200, 1994.
- [11] Fei-Fei Li and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR '05, pages 524–531, Washington, DC, USA, 2005. IEEE Computer Society.
- [12] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11) :2278–2324, 1998.
- [13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q.

- Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
- [14] Fan Hu, Gui-Song Xia, Jingwen Hu, and Liangpei Zhang. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sensing*, 7(11) :14680–14707, 2015.
 - [15] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat : Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv :1312.6229*, 2013.
 - [16] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv :1409.1556*, 2014.
 - [17] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe : Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.
 - [18] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.
 - [19] A. Cornuéjols, M. Sebag, and J. Mary. Classification d’images à l’aide d’un codage par motifs fréquents. In *Workshop sur la fouille d’images (RFIA-04)*, Toulouse (France), 2004.
 - [20] A. Cornuéjols, M. Sebag, S. Jouteau, and J-S Liénard. Nouveaux résultats en classification à l’aide d’un codage par motifs fréquents. *Extraction des connaissances et apprentissage*, 17(2) :521–532, 2003.
 - [21] Salim CHITROUB. Analyse en composantes indépendantes d’images multibandes : Faisabilité et perspectives. *Revue de Télédétection*, 7(1-2) :3–4, 2007.
 - [22] Hervé Le Borgne. *Analyse de scènes naturelles par composantes indépendantes*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2004.
 - [23] Hervé Le Borgne, Nathalie Guyader, Anne Guérin-Dugué, and Jeanny Hérault. Classification of images : Ica filters vs human perception. In *Signal Processing and Its Applications, 2003. Proceedings. Seventh International Symposium on*, volume 2, pages 251–254. IEEE, 2003.
 - [24] Anna Bosch, Andrew Zisserman, and Xavier Muñoz. Scene classification using a hybrid generative/discriminative approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(4) :712–727, April 2008.
 - [25] Sugata Banerji, Atreyee Sinha, and Chengjun Liu. A new bag of words lbp (bowl) descriptor for scene image classification. In *International Conference on Computer Analysis of Images and Patterns*, pages 490–497. Springer, 2013.