Table des MATIÈRES

[1 Etat de l’art 4](#_Toc514669402)

[1.1 Introduction 4](#_Toc514669403)

[1.2 Les bases de données existantes 4](#_Toc514669404)

[1.2.1 SUN 4](#_Toc514669405)

[1.2.2 MIT Indoor 4](#_Toc514669406)

[1.2.3 LabelMe 4](#_Toc514669407)

[1.3 Travail communautaire 5](#_Toc514669408)

[1.3.1 Mesures contre le vandalisme 6](#_Toc514669409)

[1.4 Méthodes de classification de scènes 6](#_Toc514669410)

[1.4.1 Méthodes basées sur les caractéristiques globales de l’image 6](#_Toc514669411)

[1.4.2 Méthodes basées sur les caractéristiques locales de l’image 7](#_Toc514669412)

[1.4.3 Méthodes basées sur Sac de mots (bag-of-words) 7](#_Toc514669413)

[1.4.4 Formalisme des fonctions de croyance 8](#_Toc514669414)

[1.5 Conclusion 9](#_Toc514669415)

[2 Conception 10](#_Toc514669416)

[2.1 Introduction 10](#_Toc514669417)

[2.2 Conception de la base de données 11](#_Toc514669418)

[2.2.1 Structure et contenu de la base de données LRIA\_BD 12](#_Toc514669419)

[2.2.2 Téléchargement d’une instance de la base de données 14](#_Toc514669420)

[2.3 Classification en utilisant les fonctions de croyance 15](#_Toc514669421)

[2.3.1 Structures de données utilisées pour le classificateur 15](#_Toc514669422)

[2.3.2 Attribution de masse et le calcul de fréquences 19](#_Toc514669423)

[2.3.3 Apprentissage du classificateur 20](#_Toc514669424)

[2.3.4 Classification d’une scène 21](#_Toc514669425)

[2.4 Travail communautaire 23](#_Toc514669426)

[2.4.1 Gestion des utilisateurs 24](#_Toc514669427)

[2.4.2 Gestion des données 24](#_Toc514669428)

[2.5 Conclusion 25](#_Toc514669429)

[3 Résultats et implémentation 26](#_Toc514669430)

[3.1 Introduction : 26](#_Toc514669431)

[3.2 Outils de développement 26](#_Toc514669432)

[3.2.1 Architecture MVC 26](#_Toc514669433)

[3.2.2 Langages utilisé 27](#_Toc514669434)

[3.2.3 Outils utilisé 27](#_Toc514669435)

[3.3 Site web et base de données 28](#_Toc514669436)

[3.3.1 Partie utilisateur 29](#_Toc514669437)

[3.3.2 Partie administrateur 34](#_Toc514669438)

[3.3.3 Implémentation de la base de données 36](#_Toc514669439)

[3.4 Résultats de classification 37](#_Toc514669440)

[3.4.1 Tests des paramètres du classificateur 39](#_Toc514669441)

[3.4.2 Comparaison avec l’état de l'état 43](#_Toc514669442)

[3.5 Conclusion 45](#_Toc514669443)

Introduction générale

Nous sous-estimons notre capacité extraordinaire à comprendre ce que nous voyons, même un enfant de trois ans pourquoi pas 4 ou 2 ?? peut facilement distinguer une chambre à coucher d'un salon. Bien qu'il semble simple pour nous, la reconnaissance et classification de scènes est encore un défi énorme pour les ordinateurs. Cependant, pour ~~s~~ce faire, l’existence d’une base de données convenable en termes de volume et cohérence du jeu de données est critique pour la qualité de la classification. Malheureusement, les bases de données dédiées à cela sont aussi rares que~~’on~~ nous les retrouvons revenir à chaque étude menée afin de tester les méthodes ~~y~~ proposées.

Dans ce même contexte, l’objet de notre travail consiste à contribuer à ~~satisfaire le~~  à répondre au besoin exprimé dans le domaine de classification de scènes en deux façons.

La première est d'établir une base de données permettant la génération automatique des sous-bases de données selondes paramètres établis. Cette base de données repose sur la collaboration d’une communauté d’utilisateurs en ligne participant à l’enrichir, mais aussi sans négliger l'aspect de cohérence, en armant cette base de données avec un système de validation et filtrage de données.

La seconde est de proposer une méthode de classification de scènes ~~se limitant~~ basées sur les objets annotées qu’elles comportent. Cette méthode est inspirée de la théorie des fonctions de croyance, une théorie mathématique qui nous a permis de réaliser un modèle fondée sur l’idée que les ensembles d’objets apportent plus d’information que des objets séparés.

Finalement, la présentation de ce travail se fait au sein de notre site web.

Dans ce mémoire, nous allons ~~exploiter~~ présenter ce projet en trois chapitres

1. État de l’art, ou nous ~~parlerons~~ présenterons quelques travaux et recherche qui ont attrait au thème de notre projet ~~déjà faites concernant notre travaille~~.
2. Conception, dans lequel nous allons ~~discuter~~ présenter nos idées , notre vision et les algorithmes que nous avons mis au point pour ~~la conception~~ répondre aux objectifs de notre ~~du~~ projet.
3. Résultat et implémentation, au sein duquel nous présenterons l’implémentation de notre travail~~le~~ et différentes outils et langages utilisées pour cela. Ensuite, nous présenterons les résultats des tests de notre approche proposée sur plusieurs bases de données, et puis nous terminerons par des comparaisons ~~ces~~ résultats obtenus avec ceux~~lle~~ des autres travaux du même domaine.

# Etat de l’art

## Introduction

La reconnaissance de scènesdemeure un thème de recherche très convoités dans le domaine de la vision par ordinateur malgré plusieurs travaux qui on y était faits. Cette branche a de nombreuses applications, notamment: la recherche d’images sémantiquement similaires, auto-localisation, conception des robots, appareils de surveillance et assistants personnels, etc.

Dans ce chapitre nous présenterons un état de l’art sur le domaine de classification de scènes, à savoir, les méthodes de classification les plus répondues, les bases de données dédiées à cela,le travail communautaire et le formalisme des fonctions de croyance.

## Les bases de données existantes

Un grand besoin de bases de données fiables se manifeste dans le domaine de la classification des scènes, toutefois, seules quelques-unes y sont disponibles, et qui sont encore limitées face à la complexité du problème. Nous présentons brièvement trois des plus célèbres d’entre elles :

### SUN

SceneUNderstanding (SUN)[1][2]est une base de données contenant à présent 908 catégories de scène, 313884 objets segmentés (3819 catégories d’objets) et de 131067 images. Cette base est construite sur une hiérarchie riche séparant même les catégories de scènes en 3 grandes classes : indoor, outdoor et naturel.

### MIT Indoor

Comme son nom l’indique, MIT Indoor[4][3] est une base de données de scènes d'intérieur (indoor). A. Quattoni et A.Torralba [4]y ontcréé un ensemble de 67 catégories de scènes d'intérieur (les plus grandes disponibles) couvrant un large éventail de domaines.

### LabelMe

Torralba et al.[5]ont introduit un outil d'annotation en ligne qui permet aux utilisateurs en ligne d'étiqueter les objets et leur étendue spatiale dans les images. Ils ont recueilli plus de 400,000 annotations couvrant une variété de classes de scènes et d'objets. Cet outil a était utilisé pour l’annotation de la base de données SUN [2].

La figure 1.1 rassemble 3 captures d’écran prises depuis le site web LabelMe [6].



Figure ‎1.1 Captures d’écran du site web LabelMe[6].

Le site LabelMe [6] à sa propre base de données, dans laquelle les utilisateurs inscris dans le site peuvent importer leurs propres images dans des répertoires, ils peuvent également annoter leurs images ou celles des autres utilisateurs.

## Travail communautaire

Certaines tâches sont difficiles à réaliser pour une personne, par exemple le développement d'une application ou le remplissage d'une base de données, il est donc nécessaire de travailler en groupe. Cependant, cette méthode a ses inconvénients, puisqu'il faut s’assurer de la cohérence des contributions (données) des personnes pour éviter les erreurs ou le vandalisme.

Les sites utilisant cette stratégie sont appelés «wikis».Wikipédia [7] est le meilleur exemple de wiki réussi, on cite aussiWikiHow[8], Wiktionary[9], et même Google Traduction [10] en impliquant des gens pour améliorer ces suggestions.

Quant au site web LabelMe[6][11]qui utilise également cette stratégie (comme expliqué en haut), le problème qui se pose est l’absence d’un système explicite de validation de données, par conséquent, des mauvaises contributions peuvent facilement y pénétrer.

### Mesures contre le vandalisme

Il existe plusieurs façons de lutter contre les actes de vandalisme, prennent par exemple le site web Wikipédia [7] qui utilise les stratégies ci-après(ce qui suit est un extrait depuis« VandalismedansWikipédia »[12]) :

1. En utilisant la fonctionnalité d'historique de Wikipédia, qui conserve toutes les versions antérieures d'un article afin de pouvoir es restaurer.Il existe plusieurs façons de détecter le vandalisme afin qu'il puisse être reversé :
   * Dans de nombreux cas, le vandalisme est automatiquement détecté et annulé par un « bot » (robot). Le vandale est toujours averti sans intervention humaine.
   * Wikipédia a une page spéciale qui répertorie tous les changements les plus récents. Certains utilisateurs surveillent ces changements pour un éventuel vandalisme.
   * Tout utilisateur enregistré peut activer la surveillance pour une page de son intérêt (par exemple une page qu'il a créé).
   * Tout lecteur qui tombe par chance sur un vandalisme peut le reverser.
2. Verrouiller les articles afin que seuls les utilisateurs établis, ou dans certains cas, seuls les administrateurs peuvent les modifier. Les articles « semi-protégés » ne peuvent être édités que par ceux dont le compte est considéré comme « auto confirmé ». Les articles entièrement protégés ne peuvent être édités que par des administrateurs. La protection est généralement établie à la demande d'un ou plusieurs utilisateurs, puis un administrateur familier avec les directives de protection choisit de répondre ou non à cette demande sur la base des directives.

Figure ‎1.2 Niveaux de protection sur Wikipédia

1. Bloquer et interdire la modification à ceux qui ont commis plusieurs actes de vandalisme pendant un certain temps ou, dans certains cas, indéfiniment.Cela pour empêcher la production d'autres dommages.

## Méthodes de classification de scènes

Pour répondre aux demandes exigeantes de classification et identification de scène, le monde de la recherche s’est vu voir déployer plusieurs méthodes, aussi diverses les unes que les autres. Nous présentons quelques travaux les plus courants dans le domaine de classification de scène.

### Méthodes basées sur les caractéristiques globales de l’image

Les méthodes basées sur cette approche extraient des caractéristiques de bas niveau comme les couleurs et les textures.

Vailaya et al.[13]ont utilisés cette méthode pour classifier « city » vs « paysage », Ensuite, ils étendent la méthode à un système de classification hiérarchique.

Chang et al.[14]ont créés une fonction de confiance ou croyance pour chaque catégorie de scène, une approche plus fiable est cellede Ulrich et Nourbakhsh[15] où il ontutilisé des histogrammes de couleur comme des signatures de bas-niveau de l’image et un schéma de classification à k plus proches voisins.

Oliva et Torralba[16]ont définile concept d’enveloppe spatiale basée sur des caractéristiques comme naturalité ou ouverture dont chaque une est une dimension dans l'espace de l'enveloppe spatiale.

C. Siagian et L. Itti[17]L. Itti et al.[18]ont utilisé des cartes en relief visuel de bas niveau comme l’orientation, couleur et l’intensité. Plus tardA. Vailaya et al. [19]l’on étendue en un schéma hiérarchique de classification en ajoutant d’abord une classification « indoor » vs « outdoor ».

### Méthodes basées sur les caractéristiques locales de l’image

M. Szummer and R. Picard.[20]S. Paek and S.Chang[21]ont utilisé une extension de la première méthode (caractéristiques globales de l’image 1.4.1) ou l’image est partitionnée en blocs locaux et classifie chaque bloc puis combine le tout selon un vote ou un classificateur probabiliste.

A. Mojsilovic et al.[22]C. Fredembach et al. [23]ont utilisé ces méthodes dans la classification des régions sémantiques d’imagescomme ciel, gazon, montagne.Cette méthode utilise une procédure de segmentation d’images et classifie chaque segment.

A. Quattoni and A.Torralba[4]ontcombiné les deux méthodes déjà mentionnées en créant un algorithme de reconnaissance de scènes qui combine des informations locales et globales et ont obtenu des résultats excellents dans le cas des scènes « indoor ».

### Méthodes basées sur Sac de mots (bag-of-words)

Une des méthodes de classification de scènes les plus récentes, dontL. Fei-Fei and P. Perona[24]ont utilisé, pour classifier les scènes, un modèle d'allocation de Dirichlet latente (LDA) qui est un exemple d’un « topic model » (modèle thématique) qui est un modèle probabiliste permettant de déterminer des sujets ou thèmes abstraits dans un document.Alors queA. Bosch et al.[25]ont utilisé une analyse sémantique probabiliste latente (PLSA) avec des caractéristiques locaux invariants.

S. Lazebnik et al. [26]ontmodifié la représentation sac de mots en utilisant des pyramides spatiales qui partitionne l’imageen des sous-régions et capture les relations spatiales entre les différentes parties.

Actuellement ces méthodes représentent des approches de pointe de reconnaissance de scène, mais chacunea ses désavantages, d’où la nécessité de recherche pour l’amélioration et optimisation de ces méthodes ou mise en pointe de nouvelles approches avec de meilleurs résultats.

### Formalisme des fonctions de croyance

La théorie des fonctions de croyance où théorie de Dempster-Shaferest une théorie mathématique qui vise à combiner des preuves distinctes pour calculer la probabilité d'un évènement. Elle offre une alternative à la manière bayésienne pour quantifier les jugements subjectifs avec des probabilités. Pour une question d'intérêt :

* La méthode bayésienne accorde des probabilités directement aux réponses possibles à celle-ci, ce qui peut être parfois délicat en cas de manque d’informations.
* Alors que pour les fonctions de croyance, cela peut être fait pour des combinaisons de réponses. Ensuite, les implications de ces probabilités sont utilisées pour répondre à cette question.

#### Fonction de masse

Soit Ωun ensemble fini de réponses possibles à une question Q(cadre de discernement). Une fonction de masse (normalisée) sur Ω est une fonction de tel que   
 et

|  |  |  |
| --- | --- | --- |
|  |  | ‎1.1 |

La valeur représente la croyance allouée à l’ensemble A sans prendre en considération ses parties (sous-ensembles).

#### Degré de croyance

Le degré de croyance (‘beleif’ en anglais) en un ensemble A ⊂ Ω est décrit par  tel que:

|  |  |  |
| --- | --- | --- |
|  |  | ‎1.2 |

Cette mesure représente la totalisé de la croyance allouée à A. Contrairement à la probabilité, la croyance n’est pas additive :

|  |  |  |
| --- | --- | --- |
|  |  | ‎1.3 |

#### Exemple

Big Boss a décidé que M. Jones devait mourir, il y a 3 tueurs possibles : Peter, Paul, Mary. Big Boss désigne à pile ou face le sexe du tueur (avec une pièce non truquée). On n’a aucune idée sur le choix entre Peter et Paul, dans le cas où un homme est choisi.  
M. Jones est tué par un tueur de Big Boss. (Cet exemple est extrait depuis [27]).

Question : qui a tué M. Jones ? Modélisation du problème :

|  |  |
| --- | --- |
| Cadre des fonctions de croyance  et  (notons que) | Cadre probabiliste  ,  Et par le principe de raisonnement insuffisant (PRI) :  et |

## Conclusion

Nous avons présenté dans ce chapitre un aperçu global sur les approches et bases de données les plus utilisées dans le domaine de classification de scènes. A travers cet état d’art, nous avons pu saisir le besoin se manifestant dans le domaine de classification de scènes, que ce soit en termes de jeux de données ou de méthodes de classification.

# Conception

## Introduction

Dans le chapitre 1, nous avons présenté quelques travaux rapportés dans l’état de l’art, ce qui met en cause le manque et le besoin d’innovation pour en trouver des solutions. Dans ce qui suit, nous allons présenter notre modeste contribution dans ce domaine. Dans un premier temps, nous avons créé et mis au point une base de données générique permettant de générer des instances de bases de données téléchargeables. Mais pour ne pas interroger l’intégrité de la base de données ou débattre son contenu, nous ne pouvons pas s’occuper du remplissage par nous-même, nous devons déléguer cette tâche à la communauté et donner une voix à ces membres, cela en tannant garde contre les éventuelles défaillances en introduisant un système de validation de données et de gestion des utilisateurs. Pour se faire, nous avons procédé par la mise en œuvre d’un site web due à la simplicité et la facilité d’accès aux membres de la communauté. Nous voulons aussi essayer de créer notre propre classificateur, pour cela nous avons proposé un modèle basé sur les « fonctions de croyance ».

La figure 2.1 montre un schèma récapitulatif de notre site web.

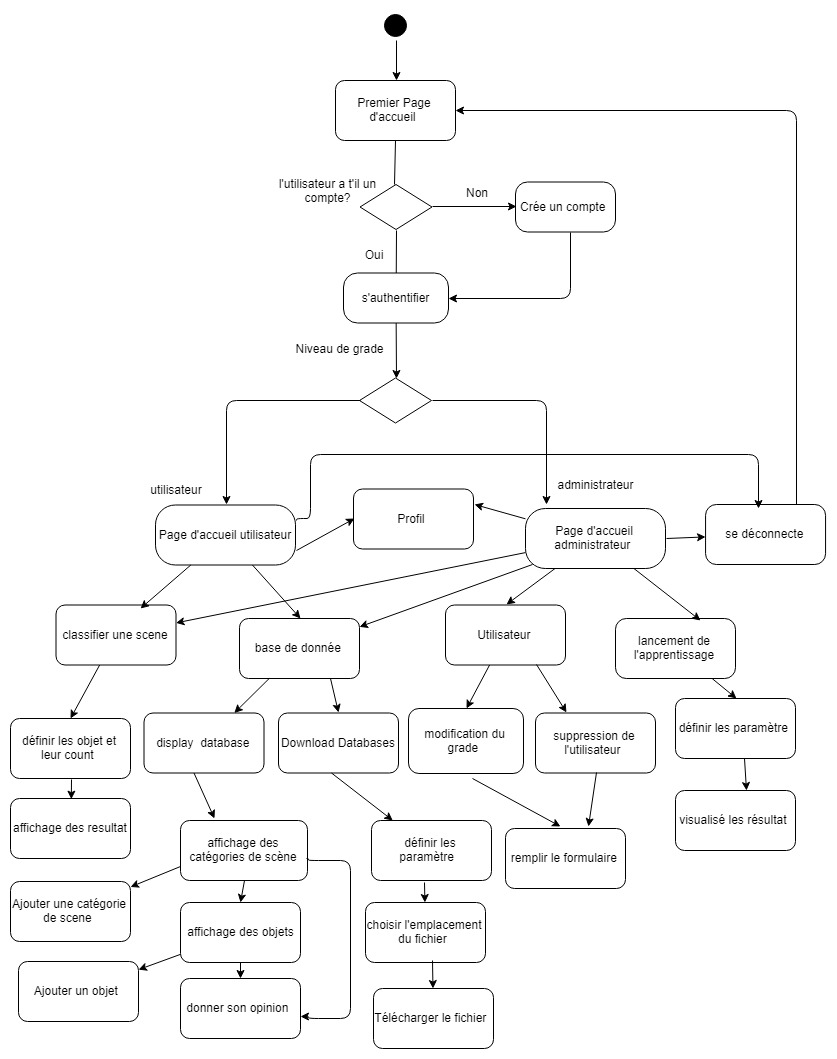


Figure ‎2.1 schéma général du site web.

## Conception de la base de données

L’utilisation des bases de données importantes en volume est cruciale pour tester les différentes méthodes de classifications proposées par la communauté des chercheurs,et vu le manque et/ou défaillance des bases de données existantes, nous avons été amenés à mettre en œuvre une base de données qui correspond aux critères suivants:

1. Elle doit être générique et instanciable, i.e. permettant de générer (et télécharger) plusieurs bases de données enfants selon des critères établis.
2. Un jeu de données ouvert à la contribution d’une communauté dans le web, mais sans négliger l’aspect de cohérence.
3. La délégation de la validation et surveillance des données aux membres les plus dignes de confiance, Cela à l’aide système de validation de données.

Notre base de données, nommée LRIA\_BD, contient des catégories de scènes, et dans chacune les objets s’y trouvant habituellement avec leurs quantités et fréquences d’apparition. Dans la phase d’instanciation, ces fréquences sont reflétées dans les instances de scènes générées.

La figure 2.2 illustre un simple exemple d’une classe de scènes (cuisine) et son instanciation en plusieurs scènes.

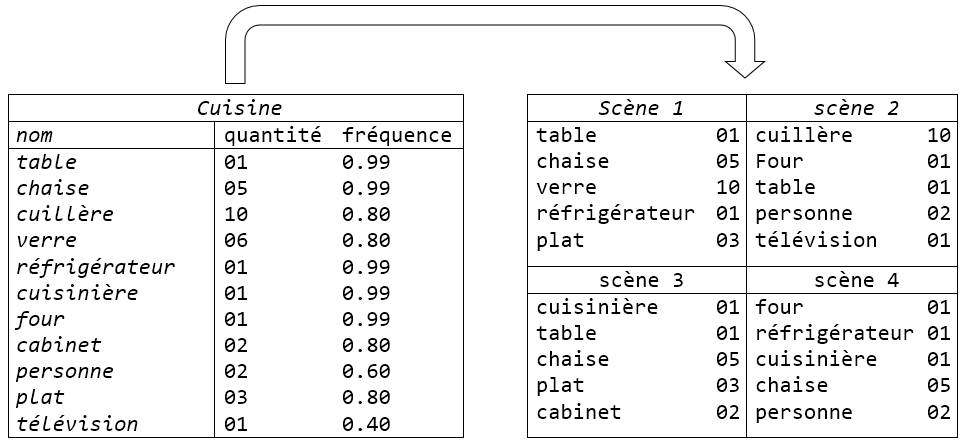


Figure ‎2.2 Instanciation de la classe de scènes « cuisine ».

Pour résumer, notre conception devrait principalement garantir les besoins suivantes :

* le stockage des différentes classes de scènes.
* le stockage des différents objets relatifs à chaque classe de scènes.
* le stockage des utilisateurs enregistrés et leurs informations relatives.
* la disposition d’un system de validation de données automatique basé sur le travail communautaire.

### Structure et contenu de la base de données LRIA\_BD

#### Les objets

Un ‘objet’ est défini comme une entité pouvant apparaitre dans une scène.Nous utilisons ce terme dans un sens général, par exemple ‘table’, ‘personne’, ‘plante’ et même ‘ciel’. Pour chaque classe de scènes, nous stockons les objets y appartenant et leurs informations relatives, à savoir :

1. L’identifiant de la classe de scènes parente (à laquelle appartient cet objet).
2. Le nom de l’objet : un nom singulier simple et général.
3. Une estimation de la fréquence d’occurrence de l'objet dans la classe parant, par exemple: une table est très fréquente (90%) dans une cuisine.
4. Une estimation de la quantité de l'objet dans la classe de scènes parente, par exemple: on trouve en moyenne 20 tables dans une cafétéria.

#### Les classes de scène

Nous appellerons ‘classe de scènes’, une abstraction généralisant un ensemble d’endroits ou espaces pouvant être occupé par l’homme. Une classe de scènes est définie parun nom, un type, et un ensemble d'objets.Le type étant une description sémantique supplémentaire de la classe (par exemple « indoor » / « outdoor »). Deux classes de scènes différentes ne peuvent pas avoir le même nom et le même type.

#### Les utilisateurs

Un utilisateur est une personne inscrite dans notre site web.Il estidentifiable par son email ou par son nom d’utilisateur et bénéficie d'un ‘grade’ déterminant les privilèges qui lui sont accordés (ces privilèges sont abordés dans la section‎2.4)

#### Le système de validation de données

Ce système se base sur la contribution des utilisateurs enregistrés et leurs opinions envers le contenu de la base pour prendre des décisions, à savoir : la validation/invalidation d’un objet/classe de scènes, la gratification ou pénalité des utilisateurs. Nous abordons cela en détails dans la section‎2.4.

#### Remplissage de la base de données

Les données relatives au classes et aux objets (les noms) ont était écrites en anglais et en minuscule. Nous avons utilisé les mêmes noms pour chaque catégorie d’objets en évitant les synonymes, ceci pour minimiser les problèmes lexicaux. Le remplissage de notre base de données était basé sur 3 sources :

1. la base de données SUN[2].
2. à partir de notre propre prospection, en recherchant des images liées au nom d'une classe de scènes sur le web (Google Images), et en nommant les objets y apparus fréquemment.
3. de la contribution de certains de nos amis et de nos proches.

Nous avons à présent 118 classes de scènes et 981 objets (370 objets distincts). La tâche n’était pas aussi simple qu’elle en a l’air. Pour une personne, le nombre de classes rempli par jour était entre 5 et 7 malgré notre meilleur effort, notamment à cause de l’ignorance des noms des objets, particulièrement dans les magasins (ou les objets sont divers), les scènes à contexte industriel ou spéciale (puisque il faut faire des recherches supplémentaires pour nommer les objets) et les scènes naturelles (car elles sont mieux identifiées par leurs ‘caractéristiques’ et aspects naturelles). En voici deux exemples illustrés par la firgure 2.3.



Figure ‎2.3 Exemples de scènes où la dénomination des objets est difficile

### Téléchargement d’une instance de la base de données

Un des avantages de la base de données que nous avons développée c’est le fait de donner la possibilité d’instancier une sous base de données paramétrée par :

1. Le nombre de classe de scènes
2. Le nombre de scènes total
3. Le nombre minimal d’objets par scène
4. Le Degré de précision, qui sert à spécifier le nombre d’objet appartenant à la scène et le nombre d’intrus (bruit).

Le téléchargement d’une instance de la base de données LRIA\_BD se fait selon l’algorithme 2.1 :

Algorithme ‎2.1Génération et téléchargement d’une base de données.

**Entrées** :  
. nbr\_categories : nombre de catégories de scènes  
. nbr\_scenes : nombre de scènes  
. nbr\_obj : nombre minimale d’objets par scènes  
. precision : le rapport des objets correctes sur le nombre d’objets totale. /\* Un objet correct (ou non bruit) est un objet appartenant réellement a la catégorie de scènes \*/  
**Sortie** : une base de données a télécharger.

**Début**

1. . Récuperer autant de catégories de scènes possibles de la BDD, pour satisfaire la demande exprimée par les paramétres.  
   . nbr\_categories = le nombre de catégories récuperées de la BDD
2. // distribuer les scènes : chaque catégorie de scènes doit être remplie par un certain nombre de scènes  
   . min = nbr\_scenes / nbr\_categories / 2 /\* 50% du cas d’une distribution égale\*/  
   . pour chaque catégorie de scènes : catégorie.nbr\_scenes = min  
   . disribuer le rest des scènes aléatoirement sur les catégories de scènes
3. // remplir les scènes par les objets ‘correctes’ :  
   . pour chaque catégorie de scènes :  
    . pour chaque scène (vide) dans catégorie :   
    . pour chaque objet correct dans catégorie :  
    . si obj.frequence >un nouveau nombre aléatoire dans [0,1] :  
    scène.ajouter(objet)  
   . completer la scène avec des objets incorrectes (bruit)
4. Livrer la base de données générée dans une fenetre de téléchargement pour l’utilisateur

**Fin**

## Classification en utilisant les fonctions de croyance

Dès le départ, notre but était de réaliser un classificateur capable de catégoriser une scène en se limitant uniquement à ses objets, cela en donnant pour chaque catégorie proposée un degré de certitude.Pour ce faire, nous nousbasonssur les données suivantes :

1. La fréquence des objets dans leurs catégories de scènes, par exemple : table, chaise et réfrigérateur sont des objets que nous trouvonsfréquemment dans une cuisine.
2. Les ensembles d’objets peuvent être plus indicatifs que des objets uniques, exemple :pour la classe « plage », l’ensemble d’objets {sable, mer, palmier} est très indicatif.

Le modèle de fonctions de croyance par contre correspondait mieux à nos besoins comme il le montre le tableau 2.1 :

|  |  |
| --- | --- |
| **concept des fonctions de croyance** | **Analogie avec notre idée** |
| pour un ensemble d’éléments Ω, chaque combinaison d’éléments (sous-ensemble) de Ω lui est alloué une crédibilité ou une masse | nous voulons accorder pour chaque ensemble d'objets, une valeur indiquant son appartenance à une classe de scènes. |
| définition d’une manière d’estimer des croyances et possibilité de représenter des informations incomplètes d’une manière riche et flexible [27]. | nous voulons munir nos résultats ou classifications chacune d’une certitude. |

Tableau ‎2.1 Analogie entre notre idée et les concepts du formalisme des fonctions de croyance.

Nous avons utilisé un modèle basé sur ce formalisme tel que chaque ensemble d'objets *E* dans une classe de scènes *C* est alloué une masse donnée selon sa fréquence dans C, sa taille et d'autres paramètres dont nous parlerons par la suite.

Dans ce qui suit, nous présenterons les structures de données que nous avons utilisées pour concevoir notre classificateur.

### Structures de données utilisées pour le classificateur

A la base, l'idée était d’avoir une structure de données capable d’indiquer pour chaque couple (*C* : classe de scènes, *E* : ensemble d'objets) toutes les données qu’y sont relatives, notamment la masse de l’assertion « *E* appartient à *C ».* Le tableau 2.1 montre un aperçu de cette structure ou les lignes représentent les catégories de scènes et les colonnes représentent les ensembles d’objets.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | {bed} | {tree} | … | {car,road} | {sand,sea} | … | {tree,rock,bird} |
| *Bedroom* | 0.025 | - | - | - | - | - | - |
| *Street* | - | 0.01 | - | - | - | - | - |
| *Corridor* | - | - | - | - | - | - | - |
| *Park* | - | 0.02 | - | 0.25 | - | - | 0.25 |
| *Beach* | - | 0.01 | - | 0.02 | 0.1 | - | - |
| *Forest* | - | 0.03 | - | 0.08 | - | - | 0.3 |
| *…* | - | - | - | - | - | - | - |

Tableau ‎2.2Matrice de masses relatives aux ensembles d’objets par rapport à une classe de scènes.

Seulement, la masse d’un ensemble d’objets dépend de la quantité de chaque objet qu’il contient, comme exemple: 1 table et 3 chaises aura plus de masse relative à ‘cuisine’ que 20 tables et 50 chaises ; qui tend plus vers ‘restaurant’ par exemple.Par conséquent, la masse doit être calculée dans la phase de classification (et non pas dans la phase d’apprentissage), car il est impossible de le faire pour toutes les quantités possibles au préalable, par ce fait, les besoins ci-après devaient être satisfaits par notre implémentation:

1. L’optimisation des ensembles d'objets dont la masse relative est nulle
2. La vérification rapide de l’appartenance d’un ensemble d’objet a une classe de scènes
3. La recherche rapide d’un ensemble d’objets (ce qui est crucial pour la rapidité de la classification)

Sous-entendu, il est nécessaire d’avoir un double accès aux données de la structure :

1. pour chaque ensemble d'objets, un accès à toutes les classes de scènes dont il fait partie
2. pour chaque classe de scènes, un accès à tous les ensembles d’objets qu’elle contient

Pour ce faire, nous avons utilisé une implémentation orientée objet (en Java), nous présentons en ce qui suit les structures de données les plus importantes dans notre implémentation.

La figure 2.4 illustre les structures de données du classificateur :

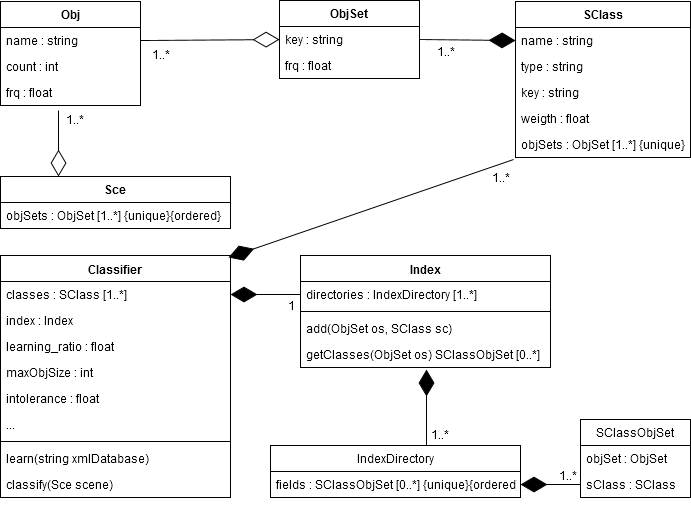


Figure ‎2.4Diagramme des classes du classificateur.

#### Objet (Obj)

Les objets ou les entités dont une scène est composée, par exemple ‘table’, ‘personne’ et même ‘ciel’sont munis d’un ‘count’représentant sa quantité dans une scène donnée (ex: 100 chaises dans une salle de conférences).La fréquence ‘frq’ d’un objet sert à optimiser le classificateur en éliminant les objets de fréquence inférieur à un certain seuil d’intolérance.

#### Ensemble d’objets (ObjSet):

Il s’agit d’une collection d’objets triés selon leurs noms. Les ensembles d’objets sont comparables par leurs clés ‘key’ : un nom unique généré en concaténant les noms des objets membres avec la séparation ‘/’, ex : « voiture/rue ».

#### Classe de scènes (SClass)

Nous rappelons qu’une ‘classe de scènes’ est une abstraction généralisant un ensemble d’endroits ou espaces pouvant être occupé par l’homme. Une classe de scènes contient des ensembles d'objets ‘objSets’ sans répétition, nous avons utilisé pour cela une implémentation en un arbre binaire de recherche. Nous utilisons une valeur appelée ‘Weight’, qui est propre à chaque catégories de scènes, pour normaliser les masses des ensembles d'objets dans cette classe de scènes (rendre leur somme égales à 1):

|  |  |  |
| --- | --- | --- |
|  |  | ‎2.1 |

Avec étant la fréquence de *e* dans cette classe et sa fréquence dans la base de données (voir la formule ‎2.2).

#### Scène (Sce)

Une scène est un ensemble ordonné d’objets.Ces derniers sont ordonnés selon leurs noms afin d’optimiser la rechercher d’un objet donné, cela pour pouvoir effectuer plusieurs calcules lors de l’apprentissage (fréquences, la quantité ‘count’ moyenne, etc.).

#### Classificateur

Le classificateur est l’entité contenant toutes les données de l'apprentissage. Chaque instance de cette de classificateur est capable d’apprendre à partir d’une base de données, et par la suite faire des opérations de classification sur une ou plusieurs scènes. Les éléments les plus pertinents de classificateur sont :

* « classes » :un tableau contenant toutes les classes de scènes, et tous les ensembles d’objets qu’y font partie. Il permet donc un accès depuis une classe de scènes à tout son contenu.
* « index » : l’index fourni l’accès depuis les ensembles d’objets vers toutes les classes de scènes où ils se trouvent. Cette structure  ne contient que des références.
* « learning\_ratio » : désigne le taux d’apprentissage depuis la base de données d’étude, tel que dans chaque catégorie de scènes, nous prenons comme échantillon (pour l’apprentissage) un pourcentage de scènes égal à cette valeur.
* « maxObjSetSize » : représente la cardinalité maximale des ensembles d’objets gérés par le classificateur, ces ensembles d'objets sont générées depuis l’échantillon d'apprentissage.
* « intolerance » : désigne une valeur entre 0 et 1 utilisée comme seuil de fréquences des objets, en éliminant tout objet ayant une fréquence inférieure à ce seuil. Cela dans le butd’optimiser le classificateur en termes de temps d’apprentissage.

#### Index

L’index (voir la figure 2.5) est utilisé pour rechercher efficacement les ensembles d’objets, il correspond à un tableau d’annuaires (directories) séparant les ensembles d’objets selon leurs tailles. Un annuaire « IndexDirectory »contient les clés de tous les ensembles d’objets appris d’une taille donnée (en omettant les quantités d’objets, Ex : « table/chair » au lieu de « 5table/20chair »).

Les clés sont ordonnées et uniques. Chaque clés *k* pointe vers toutes les couples (*Os* : ObjSet, *Sc* : SClass) tel que *Os* a pour clés *k*, et que *Os* appartient à *Sc*.

Nous avons implémenté « IndexDirectory » sous forme d’arbre binaire de recherche, ce qui permet d’accélérer considérablement le temps de recherche et l’ajout d’éléments. Voici un schéma illustrant la structure de l’index.

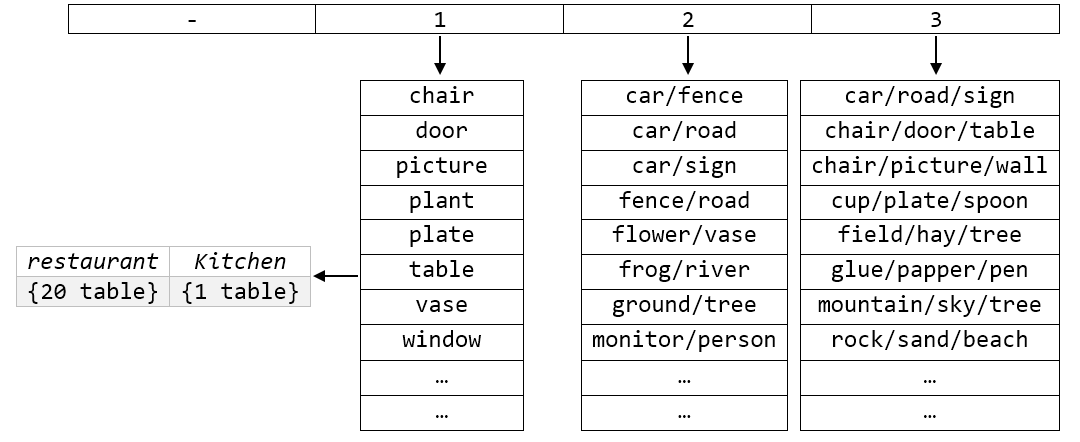
****

Figure ‎2.5 Structure  de l'index.

### Attribution de masse et le calcul de fréquences

A chaque ensemble d'objets dans le classificateur est allouée une masse désignant son appartenance à une certaine catégorie de scènes. L’efficacité de notre classificateur dépend crucialement de la qualité de notre méthode d’attribution de masses.

Pour se faire, nous étions amenés à concevoir une formule permettant, depuis un ensemble de paramètres concernant un ensemble d’objets, d’accorder une masse à ce dernier.Au départ, nous voulions que notre formule corresponde aux descriptions illustrés dans le tableau 2.3 :

|  |  |
| --- | --- |
| **Paramètre** | **Effet sur la masse** |
| Fréquence dans la classe de scènes, notée | favorable |
| Fréquence dans la base de données, notée | défavorable |
| Cardinalité | favorable |

Tableau ‎2.3 Effets désirés des paramètres de la fonction de masse.

Nous avons pensé à réaliser ces descriptions en multipliant les paramètres favorables avec l’inverse des paramètres non favorables comme suit :

De plus, il est nécessaire d’assurer que la somme des masses relatives à une catégorie de scènes soit égale à 1. Pour cela, nous utilisons comme diviseur un attribut appelé ‘weight’ propre à chaque catégorie de scènes (voir la formule‎2.1).

Nous arrivons alors à établir une première formule d’attribution de masse 2.2.

|  |  |  |
| --- | --- | --- |
|  |  | ‎2.2 |

Le paramètre *n*utilisé dans cette formule est appelé « exposant de cardinalité », il a pour but de valoriser les ensembles d’objets de grandes cardinalités d’une façon paramétrable.

En utilisant la formule 2.2 durant les tests de classificateur, nous avons remarqué, que parfois des classes de scènes similaires sont confondues, nous avons pensé alors qu’il est plus intéressant de favoriser les ensembles d’objets « propres » a une classe de scènes par rapport aux autres, cela en introduisant un facteur que nous appelons « facteur d’unicité » ou « α » dans la formule 2.3

|  |  |  |
| --- | --- | --- |
|  |  | ‎2.3 |

Cette fois, nous multiplions la masse des éléments existant uniquement dans une catégorie de scènes par α afin de mieux identifier cette catégorie et ainsi raffiner la classification. Il est à noter que pour les formules 2.2 et 2.3 sont égales.

Nous utilisons la formule 2.3 pour l’attribution de masses dans notre classificateur.

#### Calcul des fréquences

Soit *E* un objet appartenant à une classe de scènes :

1. *la fréquence de E dans C* = nombre de scènes dans C contenant E / nombre de scènes dans C
2. *La fréquence de E dans la BDD*= la fréquence moyenne des classes de scènes dans la BDD

#### Similarité entre objets et ensembles d’objets

La similarité entre deux objets est une quantité que nous proposons pour exploiter l’information de quantité d’objets pour améliorer la classification. Par exemple, le nombre d’arbres est un critère important pour distinguer « jardin » de « forêt ». Cette mesure se calcule comme suit :

|  |  |  |
| --- | --- | --- |
|  |  | ‎2.4 |

Plus deux objets ont des quantités proches, plus ils sont similaires. La constante *k* sert à augmenter la similarité entre les objets de petites quantités et aussi de définir une norme de similarité entre les objets.

La similarité entre deux ensembles d’objets ayant la même clés (i.e. égales en terme d’éléments) = la moyenne des similarités entre chaque objet d’un ensemble et son correspondant dans l’autre ensemble.

L’apprentissage se faitune fois depuis une seule base de données, le classificateur peut être sauvegardé par la suitesur disque pour être sollicité dans le futur. Souvent, Le temps de chargement d’un classificateur est nettement plus inférieur à celui d’apprentissage.

### Apprentissage du classificateur

L’apprentissage se fait une seule fois est depuis une seule base de données, le classificateur peut être sauvegardé en suite sur disque pour être sollicité dans le futur. Les paramètresintroduitspour lancer l’apprentissage sont les suivants :

1. Une base de données XML (d’un format bien définit) contenant des catégories de scènes, et dans chacune, des instances de scènes (des ensembles d'objets).
2. Les paramètres du classificateur, à savoir :
   1. Unseuil d’intolérance (définie précédemment dans‎2.3.1.5) utilisée pour l'optimisation.
   2. La taille maximale des ensembles d'objets.
   3. Le rapport de données à apprendre ‘Learning ratio’, le reste est réservé pour le test.
   4. paramètres des formules utilisées (voir les équations‎2.2 et ‎2.3).

L’apprentissage du classificateur depuis une base de données suit les étapesde l’algorithme 2.2

Algorithme ‎2.2 Algorithme d’apprentissage du classificateur depuis une base de données.

**Entrées** :  
 . bdd.xml : une base de données (fichier xml) contenant scènes étiquetés  
 . intolerance : pour optimiser le classificateur  
 . maxObjSize : la taille maximale des ensembles d'objets.  
 . Learning ratio : le rapport de données à apprendre (le rest est réservé pour le test).  
 . paramètres des formules utilisées.  
**Sortie** : vide

**Début** :

. analyse et extraction des données depuis 'bdd.xml'  
 . pour i = 0 à classifier.nbr\_de\_classes :

/\* soit currClass la i-ème classe de scènes parmi celles qui ont été  
 extraites, cette dérniere contient toutes les scenes d'apprentissage (selon  
 learning\_ratio : le rapport d'apprantissage) \*/  
 1- calculer les quantités 'count' moyennes des objets dans currClass

2- calculer les fréquences des objets dans currClass

3- si classifier.est\_optimisé : éliminer les objets tq   
 objet.frq < intolerrance

4- pour chaque scene dans currClass :  
 // une scene est un ensemble d'objets  
 pour chaque partie p de scene tq (taille(p) < maxObjSize) :

ajouter p à currClass.objects;

5- pour chaque os dans currClass.objSets :  
 calculer la fréquence de os dans currClass  
 calculer la fréquence de os dans la BDD  
 index.add(os, currClass) // ajouter os a l'index

. calculer les 'poids'(weights) des classes :  
 pour chaque sClass dans classifier.classes  
 sClass.weight = somme([mass(os.frq, os.frq\_BDD(), os.size, 1) pour os dans sClass.objSets]);

**Fin**

### Classification d’une scène

Nous rappelons qu’une scène est un ensemble d'objets munis de leurs quantités d’apparition dans la scène.

Sous-entendu, nous ne classifions pas directement des images en faisant une détection d’objet, mais nous travaillons sur les annotations présentes dans ces dernières. Nous avons focalisé notre étude sur la phase de classification de scènes étant donnée que les objets sont déjà détectés.

Algorithme ‎2.3 Algorithme de classification d’une scène.

**Entrées** : scene (une scène, i.e. un ensemble d'objets munis de leurs quantité)  
**Sortie** : ClassifResult : un tableau contenant des classifications possibles (des classes de scènes) munies chacune d’une croyance (beleif). Les classifications sont dans l’ordre décroissant

**Début** :

. ClassifResult *result* /\* un tableau contenant les classes de scènes possibles munis chacune d’une croyance (beleif) = une somme de masses \*/  
  
. ObjSets = generer\_tout\_les\_combinaisons\_possibles\_de(scene) /\* les ensembles   
 générés aurons une taille <= classifier.maxObjSetSize \*/  
 . pour chaque os dans ObjSets :  
 . references\_relatives\_a\_os = index.getClasses(os)  
 . pour chaque classe dans references\_relatives\_a\_os :  
 . other\_os = references\_relatives\_a\_os.objSet  
 . classe = references\_relatives\_a\_os.SClass  
 . os.frq = other\_os \* similarity(os, other\_os);  
 . m = mass(os.frq, os.frq\_BDD(), os.size, classe.weight)  
 . si classe existe dans result :  
 ajouter mass a la croyance de class dans result  
 sinon  
 result.ajouter(class avec croyance = mass)  
 . retourner result;  
**Fin**

Le test du classificateur se fait suivantl’algorithme 2.4 :

Algorithme ‎2.4 Algorithme du test d’une instance de notre classificateur

**Entrées** : la base de données (fichier xml) utilisée dans l’apprentissage et le classificateur appris.  
Les paramètres du classificateur.  
**Sortie** : un fichier contenant les résultats du test.

**Début**:

1. Récuperer de chaque catégorie de scènes dans la BDD, les scènes réservées pour le test (en utilisant classifier.learning\_ratio)
2. Pour chaque catégorie de scènes dans le classificateur :  
    . pour chaque scène dans cette catgorie :  
    . resultat\_de\_classif = classificateur.classify(scène)  
    . si resultat\_de\_classif.première\_classification() = catégorie alors :  
    . catégorie.TP = catégorie.TP + 1  
    sinon  
    . catégorie.FN = catégorie.FN + 1  
   . catégorie.sensitivity = TP/(TP+FN)
3. Sensitivity = moyenne(catégorie.sensitivity pour catégorie dans classificateur)
4. affichage/retour des résulats.

**Fin**

## Travail communautaire

Dans le but d’enrichir notre base de données et garantir sa consistance, nous proposons d’établir une plateforme collaborative permettant la manipulation et la surveillance des données par ses membres, ces derniers peuvent ajouter leurs propres donnés ou évaluer celles des autres.Pour ce faire, chaque utilisateur (ou membre inscrit) est muni d’un « grade » et de « points de vie », c’est des valeurs numériques tel que :

1. le grade détermine les privilèges qui lui sont accordés.
2. Les points de vie servent à limiter le nombre de dommages fait par un utilisateur, de sort que l'expiration de ses points entrainera la perte des privilèges de modification.

Le tableau 2.3illustre les différents niveaux de grades et leurs privilèges (unniveau donné possède tous les privilèges des niveaux inférieurs):

|  |  |  |
| --- | --- | --- |
| ]0,100[ | Beginner | - ajouter un élément (un objet ou une classe de scènes)  - suggérer la validation un élément  *+ il obtient 3 points de vie* |
| [100,200[ | Metal | - suggérer l’invalidation d’un élément  *+ il obtient 5 points de vie* |
| [200,400[ | Bronze | - peut suggérer la suppression d’un élément invalide  *+ il obtient 5 points de vie* |
| [400,700[ | Silver | - les objets ajoutés auront plus de crédibilité |
| [700,1500[ | Gold | - peut suggérer la suppression d'un élément valide |
| [1500,2000] | Super User | - peut gérer les utilisateurs |

Tableau ‎2.4 Les niveaux de grades et leurs privilèges.

Il est à noter qu’on accorde 50 points de grade pour les nouveaux utilisateurs. Les éléments (les objets et classe de scènes) peuvent être jugés par les utilisateurs (selon leurs grades) comme suit :

1. Si un élément est non valide, il peut être validé ou supprimé.
2. S’il est valide, il peut être invalidé ou supprimé.

Un utilisateur inscrit ayant le privilège de modification (points de vie non nuls) peut insérer des objets où des classes de scènes, la procédure d’ajout d’un objet/classes de scènes correspond au algorithme 2.5 :

Algorithme ‎2.5 Algorithme d’ajout d'un objet où une classe de scènes.

**Entrées :** données concernant l’objet / classe de scènes à inserer dans la base de données **Sortie :**vide

**Début :**

1. Récupérer les données depuis le formulaire et les transférer à la servlet
2. Appeler la méthode d’accès à la base de données.
3. Etablir une connexion à la base de données.
4. Envoyer la requête SQL permettant d’insérer l’élement. Ce dérnier sera « invalide ».

**Fin**

### Gestion des utilisateurs

Toute contribution d’un utilisateur dans la base de données peut affecter sa réputation.Les différentes récompenses/pénalités sont illustrés dans le tableau 2.4:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | validation | invalidation | Suppression d’un élément valide | suppression d'un élément invalide |
| Valider | +02 |  |  | −05 |
| Invalider |  | +02 | +01 |  |
| Supprimer | −05 | +01 | +02 | +02 |
| Auteur | +10 | −03 | −30 | −20 |

Tableau ‎2.5 les différentes récompenses et pénalités relatives au ‘grade’ d’un utilisateur.

Letableau 2.4représente les changements attribués au grade d’un utilisateur selon son opinion ou sa position (les lignes) envers un élément, et la décision prise par le système pour ce dernier (les colonnes). Avec chaque pénalité, l’utilisateur perd 1 point de vie, ou 2 points dans le cas d’une suppression d’un élément valide.

### Gestion des données

Notre système de validation rassemble les opinions ou jugements des utilisateurs afin de pouvoir prendre une décision (validation, invalidation ou suppression d’un élément). Les données récoltées sont :

|  |  |
| --- | --- |
| Le nombre de personnes votant pour :   1. La validation de l’élément, noté *NV* 2. L’invalidation de l’élément, noté : *NI* 3. La suppression de l’élément, noté : N*S* | La somme des grades des utilisateurs votant pour :   1. La validation de l’élément, noté *V* 2. L’invalidation de l’élément, noté : *I* 3. La suppression de l’élément, noté : *S* |

La validation, invalidation et suppression d’un élément se fait selon les algorithmes 2.6 et 2.7. Ces algorithmes sont régit par les règles suivantes:

1. Un minimum de personnes est requis pour pouvoir prendre n’importe quelle décision.
2. Le grade d’une personne donne plus d’importance à son avis.
3. La validation est un avis positif, contrairement à l’invalidation et suppression.
4. Un avis négatif est comparé avec un avis positif pour prendre une décision, et non pas avec un autre avis négatif (par exemple :pour décider entre la suppression et invalidation d’un élément).

Algorithme ‎2.6 Validation ou suppression d’un élément valide.

**Entrées** : le nombre de personnes votant pour la validation, suppression.  
**Sortie** : vide, cet algorithme prend une decision sur l’élément en question, et disribue les récompenses/pénalités au utilisateurs concernés.

**Début** :

. N = NV + NS;  
. G = V + S;  
/\* un minimum de 6 personnes est requis, ayant une somme de grades >= 2000 ou bien 15 personnes (ou plus) ayan un grade moyen de 50\*/  
*formule d’approbation 1 :*  
. si (N >= 6 et (G >= 2000 ou (N >= 15 et G/N >= 50))) alors  
 . P = (V - S)/(V + S);  
 . Si P > +0.75 alors  
 . valider l’élement  
 . gratifier / pénaliser les utilisateurs concernés  
 . geler les opinions concernant cet élément (rendre non modifiables)

. Si P < -0.75 alors  
 . supprimer l’élement  
 . gratifier / pénaliser les utilisateurs concernés  
**Fin**

Algorithme ‎2.7 Invalidation ou suppression d'un élément valide.

**Entrées** : le nombre de personnes votant pour l’invalidation, suppression et validation.  
**Sortie** : vide, cet algorithme prend une decision sur l’élément en question, et disribue les récompenses/pénalités au utilisateurs concernés.

**Début** :

. N = NI+ NS;  
. G = I + S;  
/\* un minimum de 12 personnes est requis, ayant une somme de grades >= 6000 ou bien 22 personnes ou plus ayan un grade moyen de 50\*/  
*formule d’approbation 2 :*  
. si (N >= 12 et (G >= 6000 ou (N >= 22 et G/N >= 50)))alors  
 // voir si on doit supprimer l'élement  
 . P = (V - S)/(V + S)  
 . Si P < -0.75 alors  
 . supprimer l’élement  
 . gratifier / pénaliser les utilisateurs concernés

// voir si on doit invalider l'élement  
 . P = (V - I)/(V + I)  
 . Si P < -0.75 alors  
 . invalider l’élement  
 . gratifier / pénaliser les utilisateurs concernés  
**Fin**

Remarque: les formules d’approbation (colorés en rouge dans les deux algorithmes 2.6 et 2.7) sont paramétrables par les ‘super utilisateurs’ ceux ayant un grade supérieur ou égale à 1500.

## Conclusion

Dans ce chapitre, nous avons présenté la conception de notre base de données et son système de validation ainsi que notre classificateur de scènes. Nous avons également abordé la méthode de classification conçue qui est basée sur les fonctions de croyance. Dans le chapitre 3, nous présenterons notre site web ainsi que les résultats des différents tests effectués sur notre méthode de classification de scènes en utilisant la base de données LRIA\_BD que nous avons mis au point.

# Résultats et implémentation

## Introduction :

Dans ce chapitre nous montrons le fruit de notre labeur ainsi que les outils et l’ensemble des langages de programmation nécessaires pour la réalisation de notre projet, nous l’avons implémenté en deux parties :

1. Partie site web et base de données :
   1. Après avoir discuté l’idée derrière ce projet et sa conception, nous présenterons maintenant les logiciels et les langages de programmation utilisée. Nous allons aussi expliquer le fonctionnement de notre implémentation en illustrant avec captures d’écran.
2. Parie classification :
   1. C’est la partie recherche de notre projet ayant le plus d’importance pour nous due à l’effort et le temps que nous avons investi pour sa réalisation. Dans ce chapitre nous allons vous expliquer les choix des paramètres et les formules utilisées, et discuter les résultats de cette recherche aussi vis-à-vis du travail mentionnée dans le chapitre de l’état de l’art.

## Outils de développement

### Architecture MVC

MVC est un modèle de conception qui organise le code en le découpant en couches distinctes :

* Couche model : contient le traitement le stockage et mise à jour des données
* Couche vue : contient l’interaction avec les utilisateurs et présentation des données
* Couche contrôleur : constitue le lien entre les utilisateurs et l’application et contrôle leur action

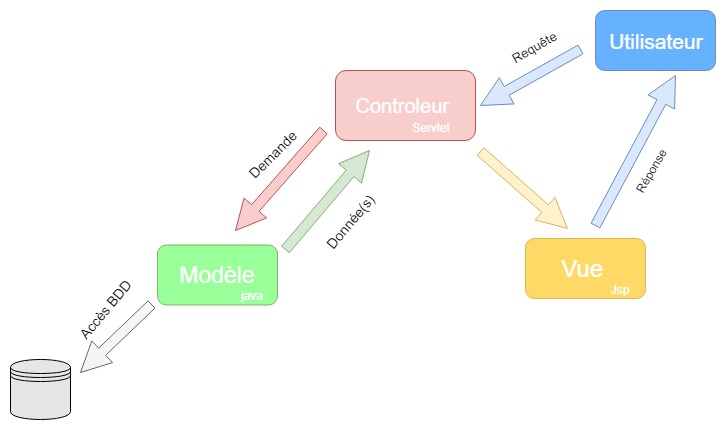
Nous avons utilisé le MVC2 qui est lourd à mettre en place car chaque vue a son propre contrôleur mais il est très pratique, facilite la maintenance et garantie l’unicité du point d’entrée de l’application.

Figure ‎3.1 Architecture MVC

### Langages utilisés

#### downloadJava Enterprise Edition

Nous avons choisi le JEE comme langage de programmation réputé pour son architecture et développement simplifié, grands choix de serveurs, outils, composants et support d'un large éventail d'exigences de sécurité.

#### HTML et CSS

HTML5 est un langage de balisage utilisé dans la création des pages web implémentées au niveau de notre projet sous forme de fichiers JSP. Ces fichiers JSP s’exécutent au niveau du serveur, elles permettent de manipuler des variables java et de transmettre des données au servlets.

CSS3 est un langage complémentaire de HTML qui permet de gérer l’apparence de la page web en manipulant le style (font, couleur, dimension …)

#### SQL

Un langage d’exploitation des bases de données qui permet de rechercher, d'ajouter, de modifier ou de supprimer des données ainsi de créer des procédures stockées. Il permet aussi de gérer des « triggers » ou des procédures exécutées après des évènements, ce langage a été d’une importance primordiale dans la création de plusieurs fonctionnalités de notre site web.

#### XML

Langage da balisage générique qui permet de structurer les données, il est souple, facile à utiliser et il existe beaucoup d’outils permettant la manipulation des fichiers écrits dans ce langage.

Nous avons utilisé ce langage dans la manipulation de base de données au niveau site web (dans le téléchargement et classification) pour rendre la procédure assez générique et accessible à tous les membres de la communauté.

### Outils utilisés

#### Eclipse Jee Oxygène

Est un environnement de développement intégré permettant la réalisation des projets multiplateforme gratuitement, il intègre tous les outils nécessaires pour faciliter le processus de développement et déploiement d’un projet, comme assistance d’écriture du code (autocomplete, génération automatique de portions de code), interface de débogage, etc.

#### download (1)Apache Tomcat 9

Serveur d’application ou conteneur de pages web permet de compiler et exécuté les servlets et les JSP. Il est gratuit, très léger et simple à utiliser avec l’IDE Eclipse car il s’y intègre facilement.

#### MySQL Workbench 6.3 CE

Nous avons utilisé MySQL Workbench pour la création de notre base de données à cause de son interface graphique qui permet la modélisation des tables et les liaisons facilement et intuitivement et surtout la simplicité de la génération des tables à partir d’un schéma de base de données.

#### downloadNotepad ++

Un éditeur de texte générique multilingue disponible pour Windows, il propose un system de coloration pour plusieurs langages de programmation, il offre aussi une grande sélection de plugins. Nous avons utilisé cet outil pour gérer nos fichiers textes, SQL et XML, il nous a servis énormément surtout pour le remplissage de notre base de données, notamment à l’aide de sa fonctionnalité de sélection et modification de plusieurs lignes à la fois.

## Site web et base de données

Le site web est le point d’accès à toutes les fonctionnalités que nous avons développées, que ce soit pour la base de données ou pour le classificateur. Ce site est censé inciter ses utilisateurs à contribuer pour le remplissage de notre base de données et/ou exploiter les différentes fonctionnalités proposées, généralement :

1. Téléchargement d’une base de données instanciée selon des paramètres.
2. Classification d’une scène en introduisant ses différents objets.
3. Tester notre classificateur sur une base de données XML d’une structure bien définie.

En ce qui suit, nous allons détailler l’implémentation du site web avec ses différentes parties.

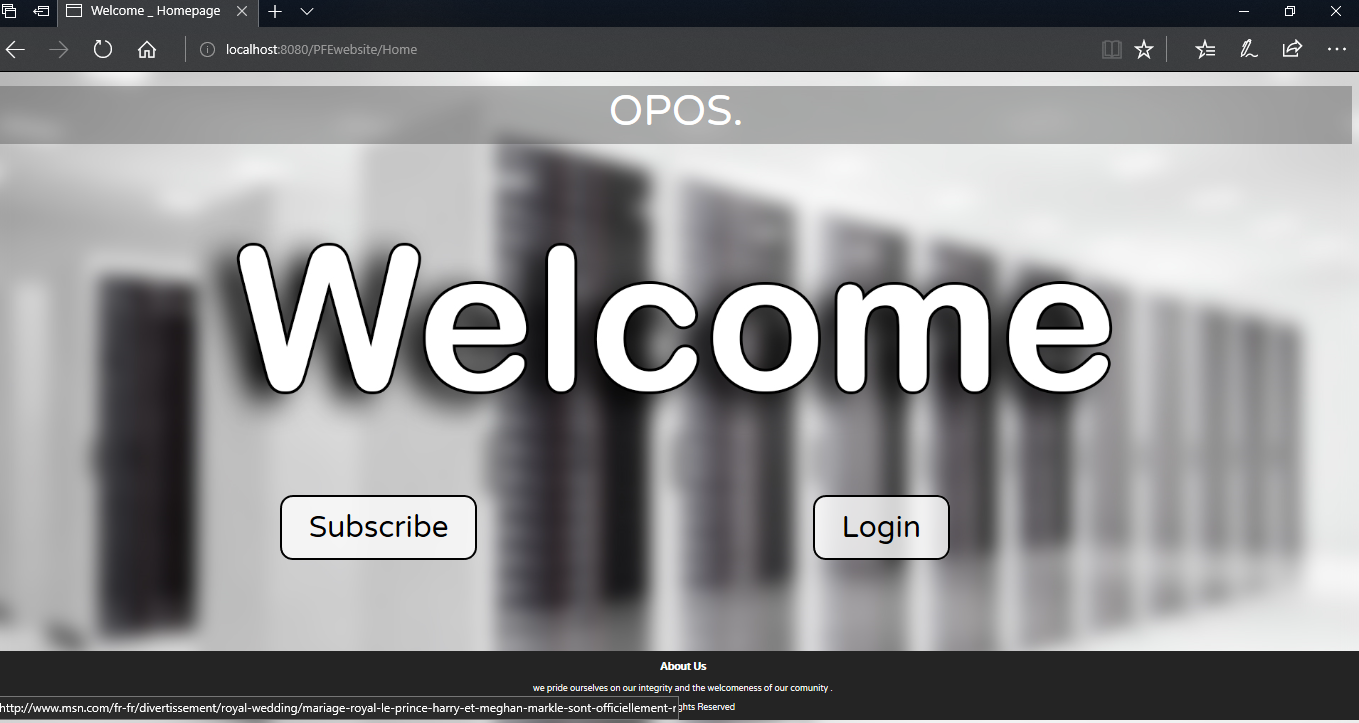
Un client doit créer un compte et/ou s'authentifier pour avoir accès aux fonctionnalités implémentées dans le site, la figure 3.2 montre la première page d’accueil d’où le visiteur peut s’authentifier ou crée un compte.

Figure ‎3.2 Page d'accueil du site web.

### Partie utilisateur

Après son authentification, un utilisateur a accès aux fonctionnalités suivantes :

La figure 3.3montre la page d’accueil d’un utilisateur.

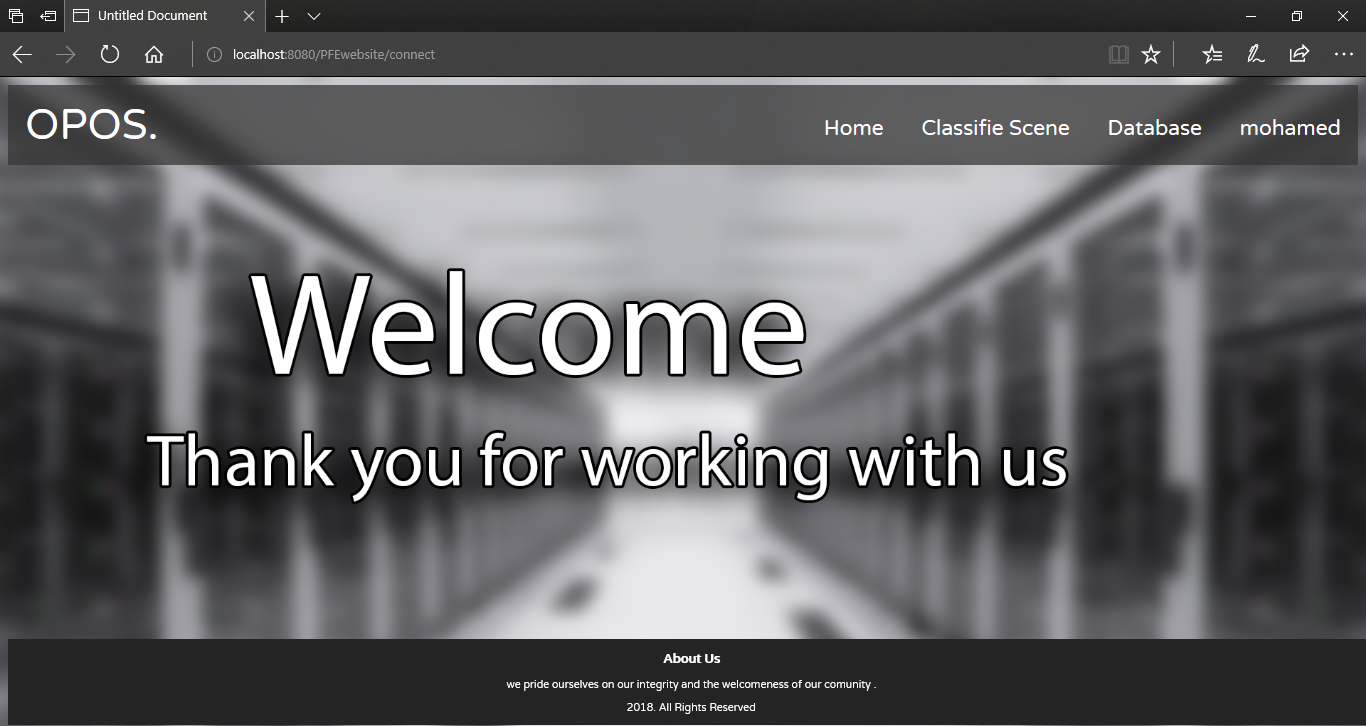


Figure ‎3.3 Page d'accueil utilisateur.

#### Access à la base de données « Database »

Tout utilisateur peut visualiser le contenue de la base de données en passant la souris sur l’onglet « Database » deux fonctionnalités s’affichent (voir la figure 3.4).

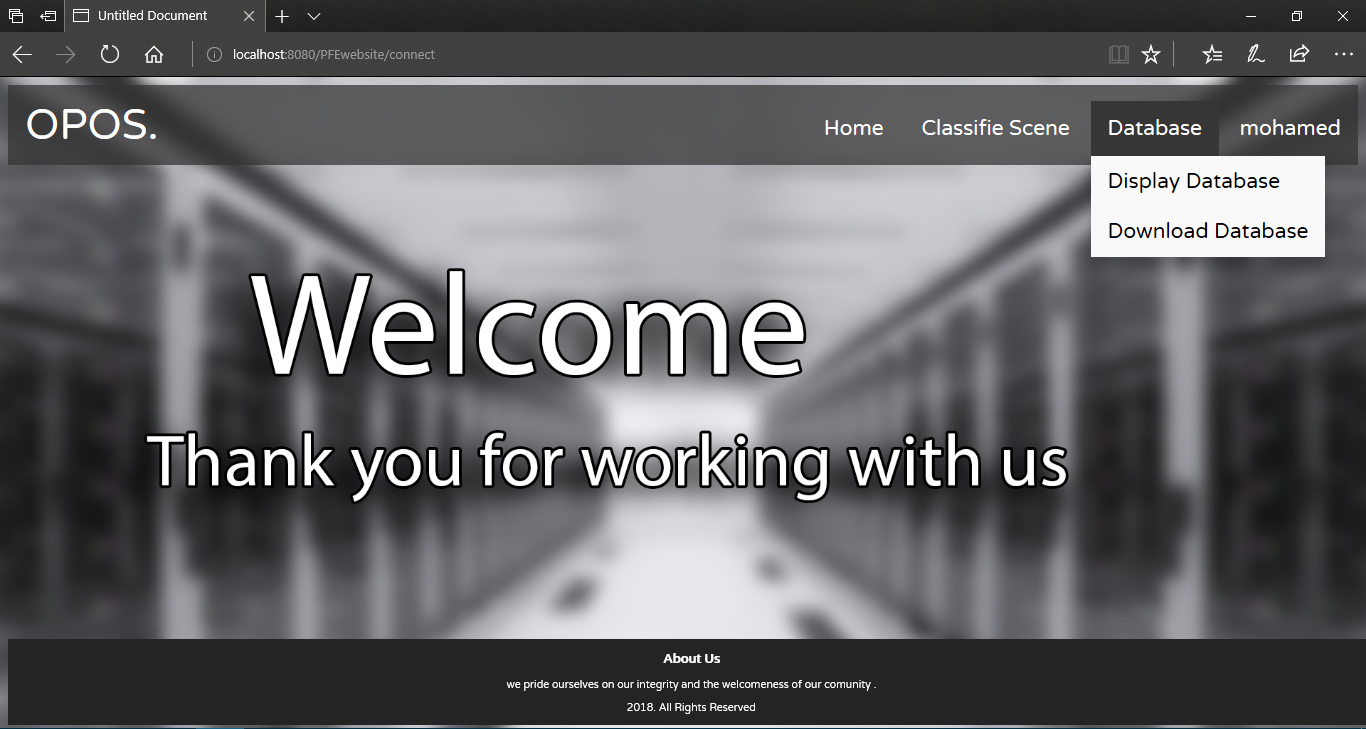


Figure ‎3.4Fonctionnalités de "Database".

En cliquant sur « Display Database », le site affichera la liste de toutes les catégories de scènes et proposera aux utilisateurs la possibilité de donner leurs opinions sur le contenu. En cliquant sur une catégorie de scènes, une liste d’objets associés sera affichée en deux sections :

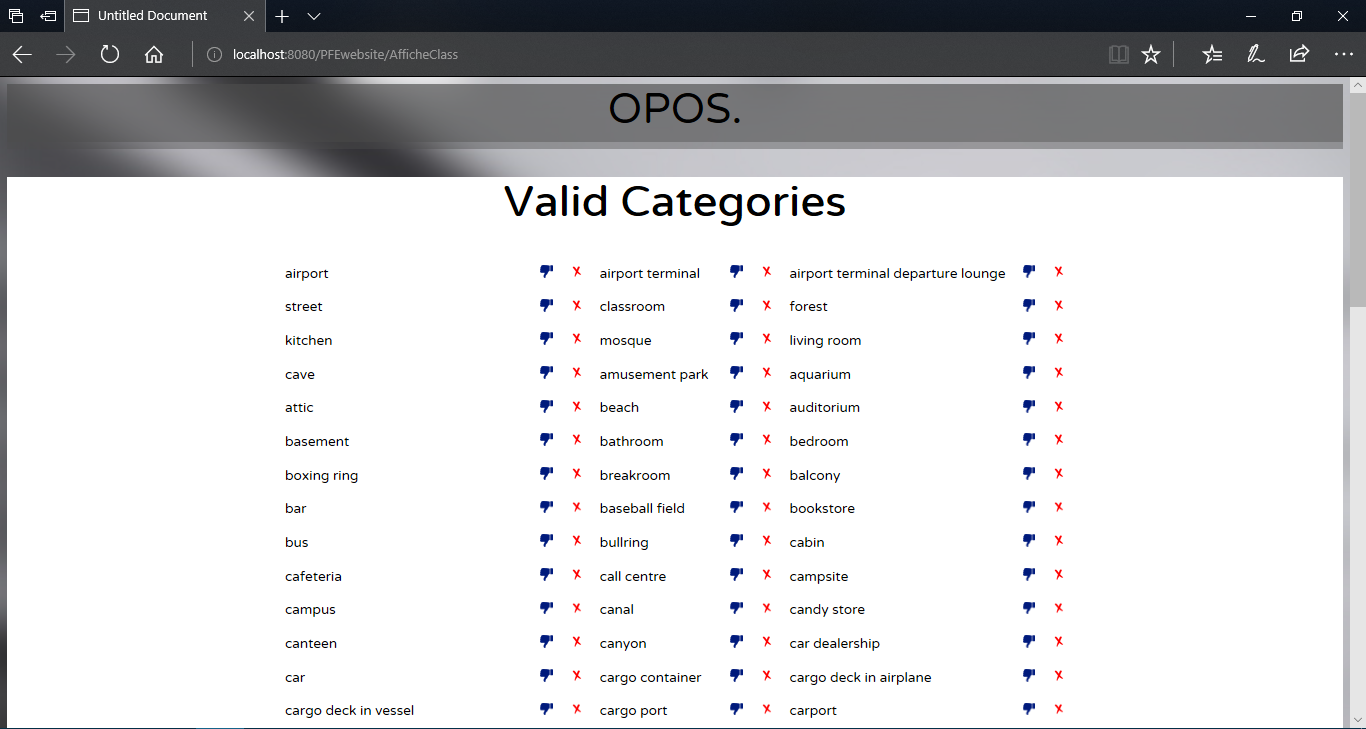
1. Les scènes/objets valides avec deux opinions montrés dans la figure 3.5:

Figure ‎3.5 Listes des catégories de scènes valides.

* 1. L’utilisateur peut signaler une classe de scènes/objet et demandera alors sa suppression de la base de données.
  2.  L’utilisateur peut exprimer son désagrément de la classe de scènes/objet, ceci aboutira à l’invalidation de la classe de scènes/objet.

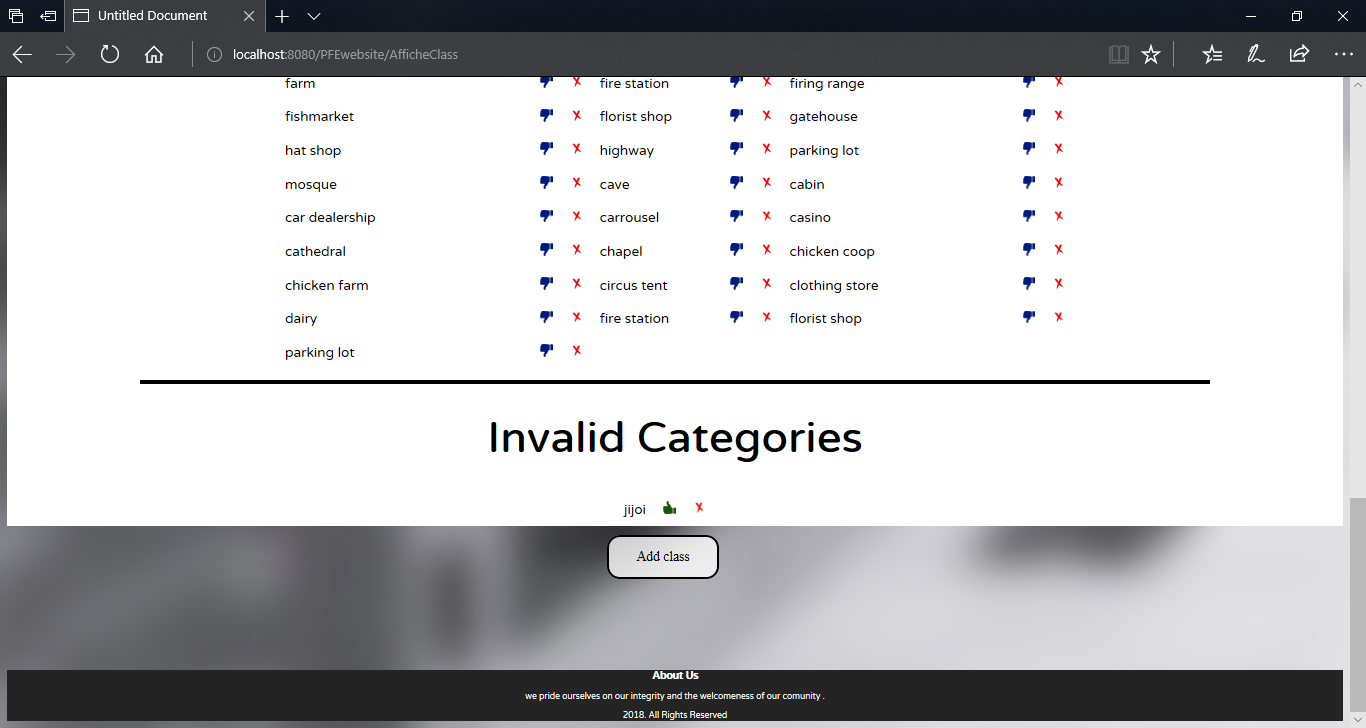
1. Les scènes non valides (voir la figure 3.6) pour lesquelles les utilisateurs peuvent suggérer :

Figure ‎3.6 Listes catégories de scènes invalides.

* 1.  Nous avons pensé que la suppression doit être disponible dans cette section pour éliminer les intrus et les erreurs avant leur introduction (validation par erreur ou par mal pensé) à la base de données.
  2. L’utilisateur pourra supporter l’ajout de la classe de scènes/objet dans la base de données i.e. sa validation.

Les opinions seront traitées après que suffisamment de membres ont donnés leurs avis, ces derniers sont biaisés parle grade de leur propriétaire. Cela a été détaillé dans la section ‎2.4 « travail communautaire ».

#### Ajout d’une classe de scènes / objet

L’utilisateur pourra aussi ajouter une classe de scènes/objet depuis la page d’affichage des scènes/objets.

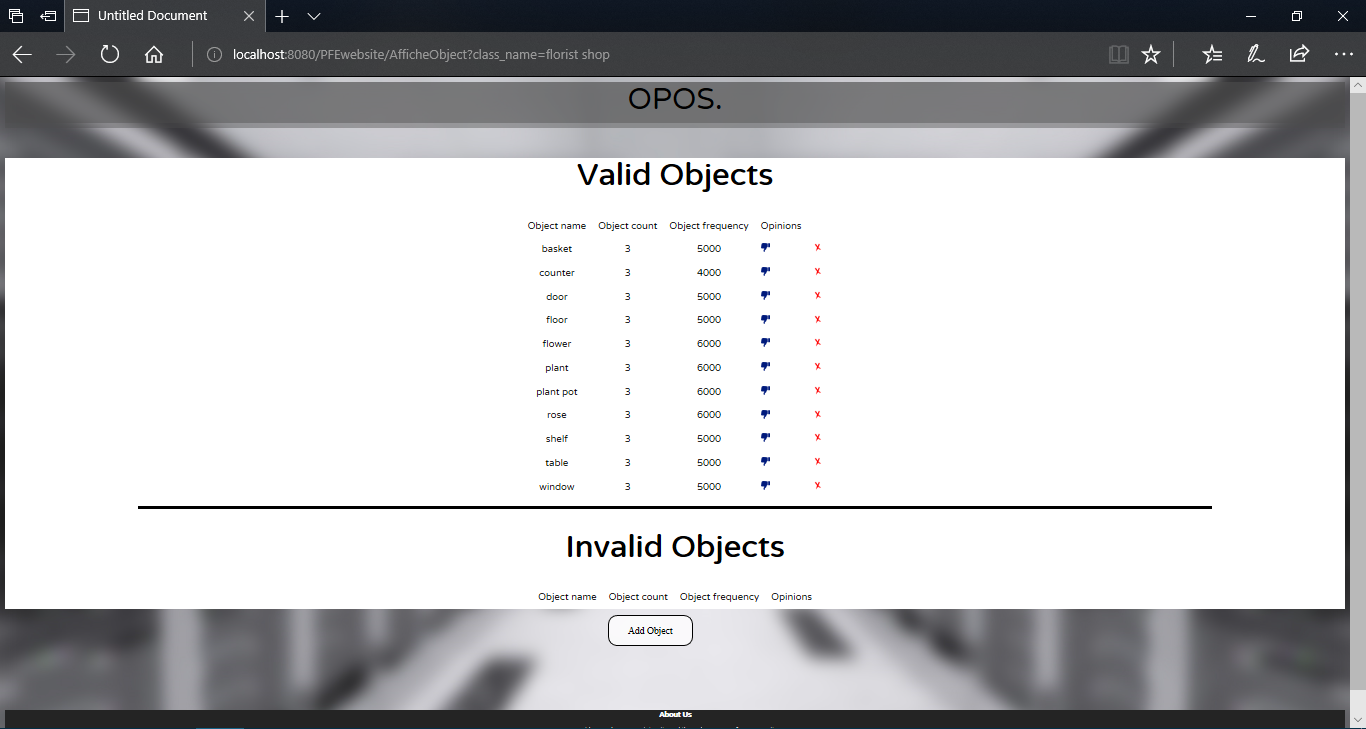


Figure ‎3.7 Listes des objets valides/invalides.

La figure 3.7 montre la liste des objets valides et invalides suivit d’un bouton « Add Object » qui permet d’afficher le formulaire d’ajout d’un objet (voir la figure 3.8), même démarche pour ajouter une catégorie de scènes.

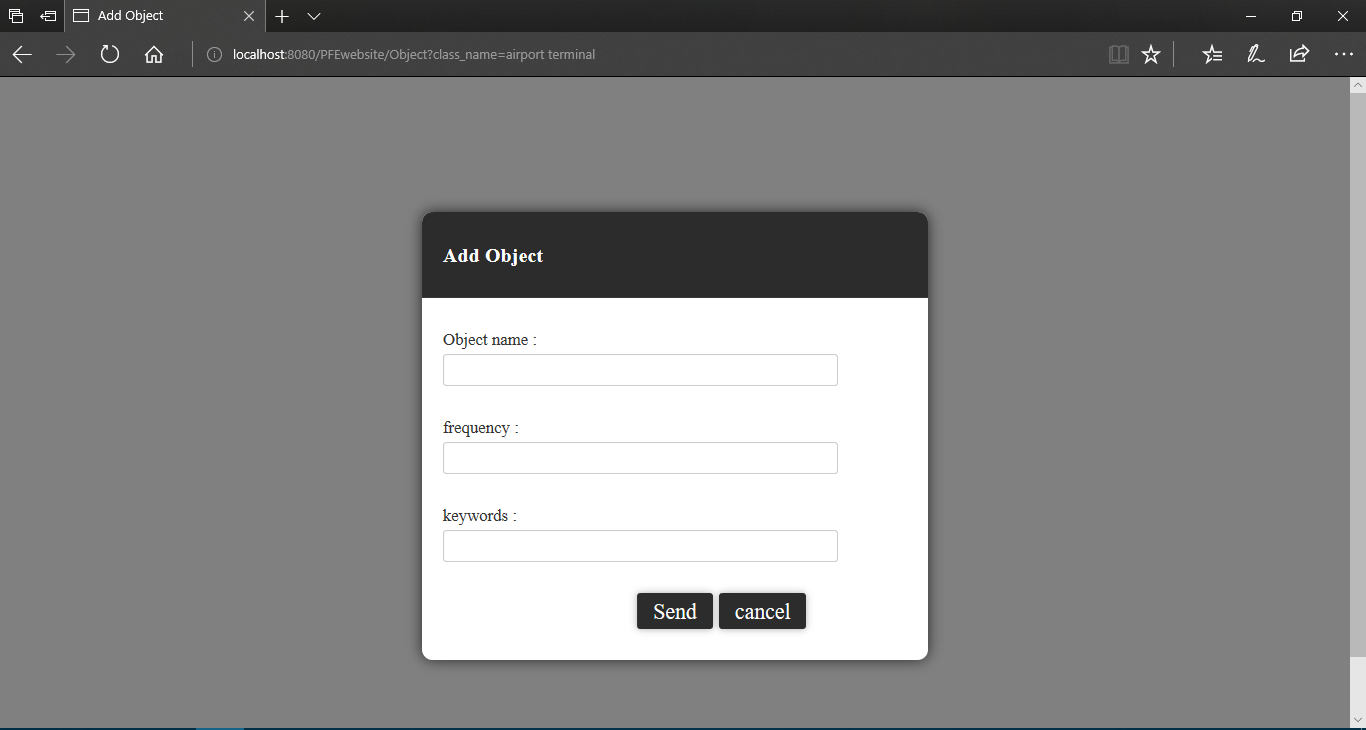


Figure ‎3.8 Formulaire d'ajout d'un Object.

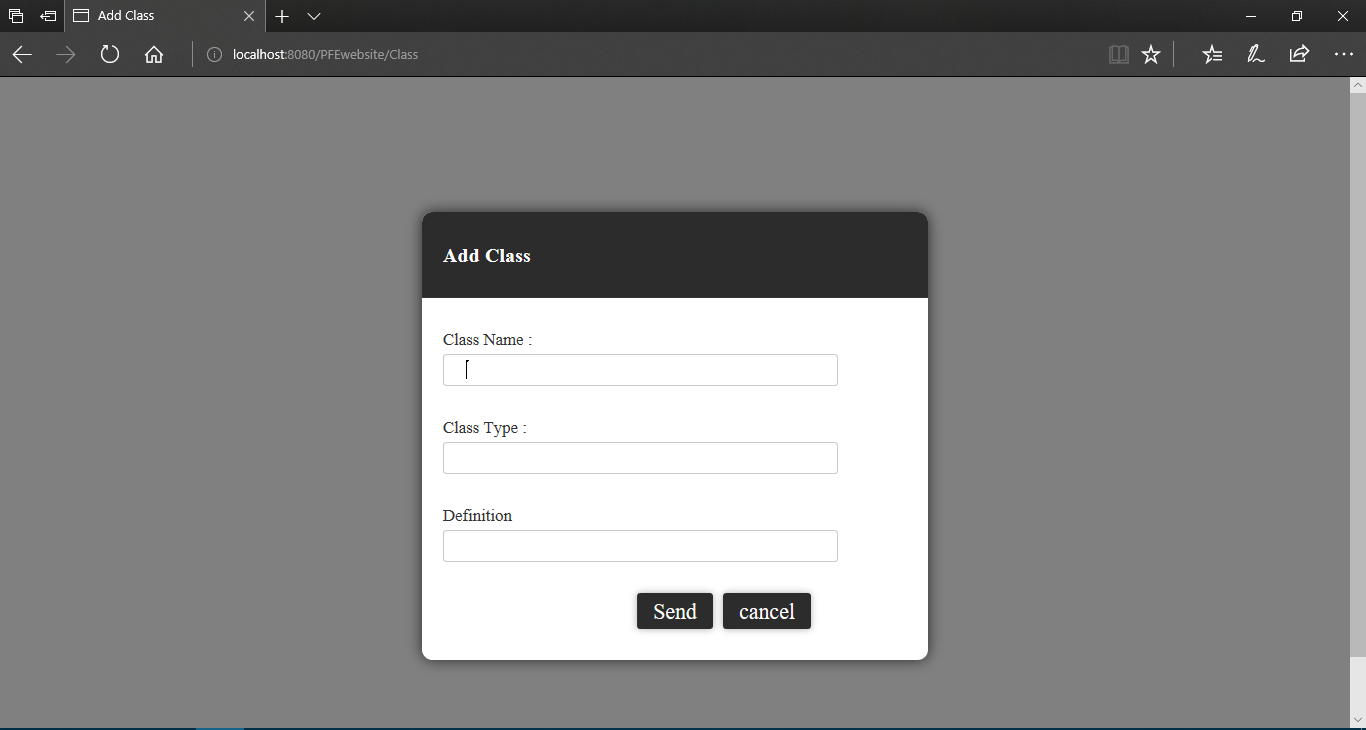


Figure ‎3.9 Formulaire d'ajout d'une catégorie de scène.

En remplissant les champs nécessaires, le nouvel élément introduit sera ajouté à la liste des objets/classes de scènes invalides.

#### Téléchargement de la base de données

Un utilisateur peut commander une instance de la base de données en cliquant sur « Download database » selon des paramètres mentionné dans ‎2.2.2.

La figure 3.10montre la page d’insertion des paramètres de téléchargement et la figure 3.11 montre le téléchargement du fichier XML contenant l’instance générée de la base de données.

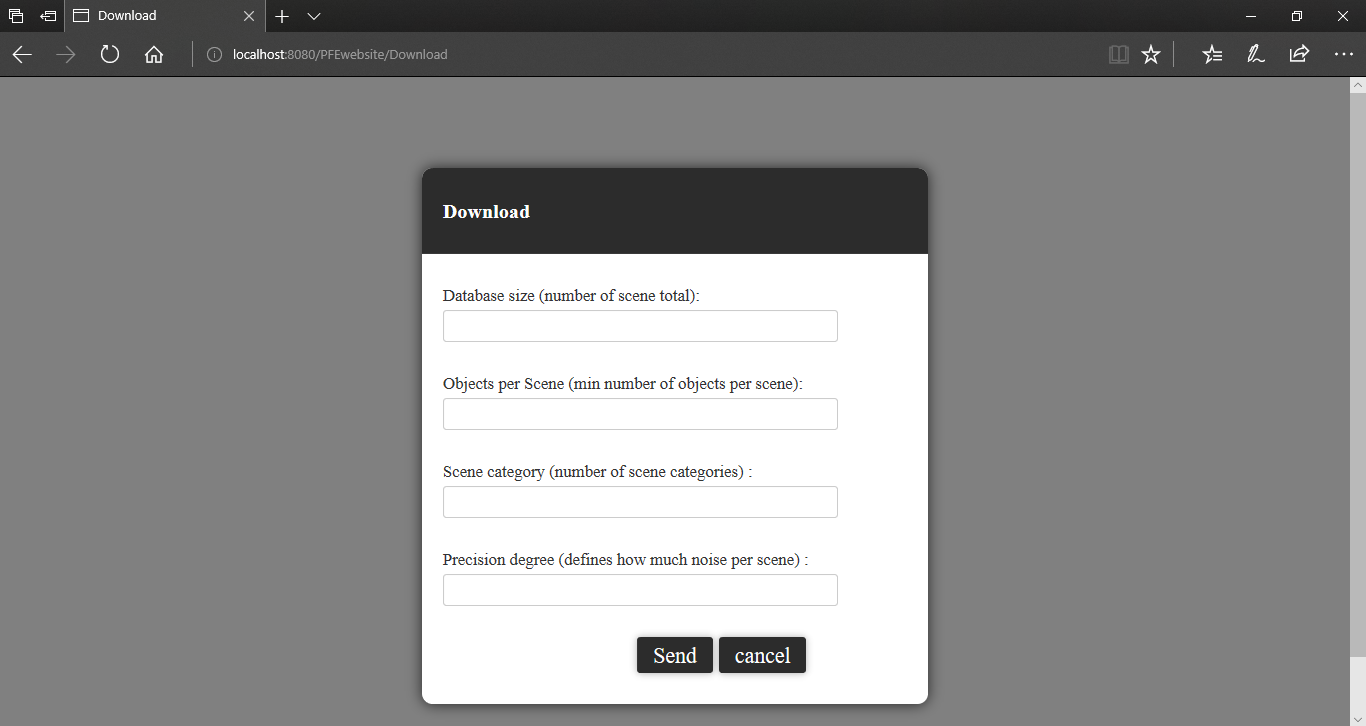


Figure ‎3.10 Formulaire d'insertion des paramètres pour téléchargement

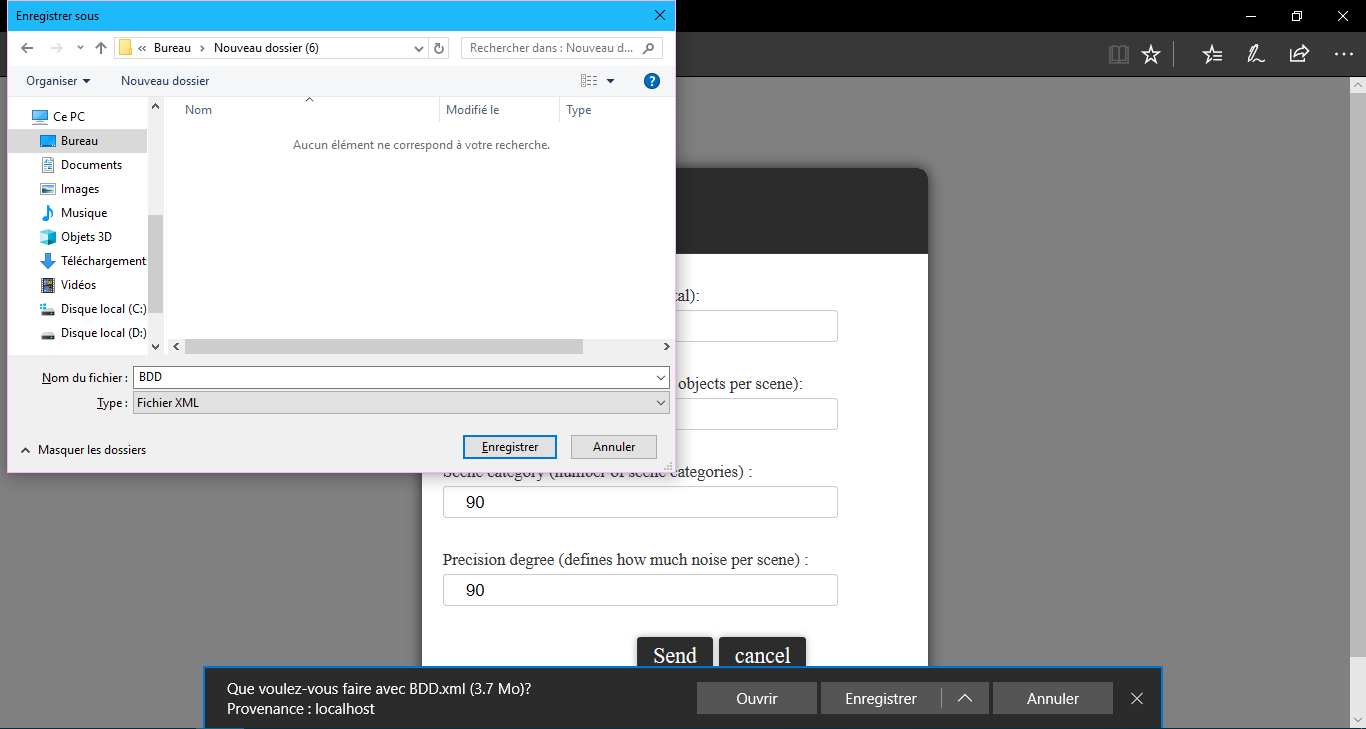


Figure ‎3.11 Téléchargement d'une instance

#### Classification d’une scène

L’utilisateur peut tester la classification d'une scène, cela en ouvrant la page « classify scene » où il peut remplir un formulaire (voir la figure 3.12) avec les noms d’objets et leurs quantités dans la scène.

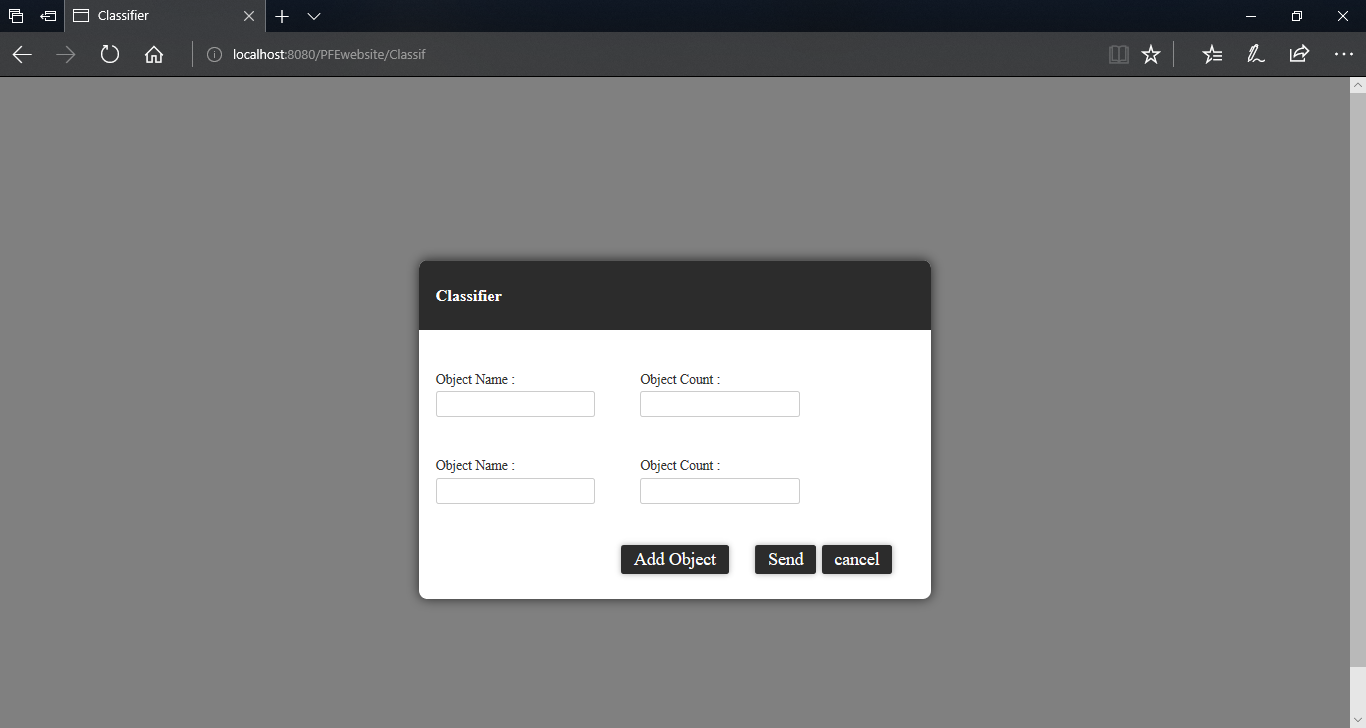


Figure ‎3.12 Formulaire d'insertion d'objet pour classification.

### Partie administrateur

L'interface d'administrateur (voir la figure 3.13)offre les mêmes fonctionnalités que l'interface utilisateur en plus des options ci-après :

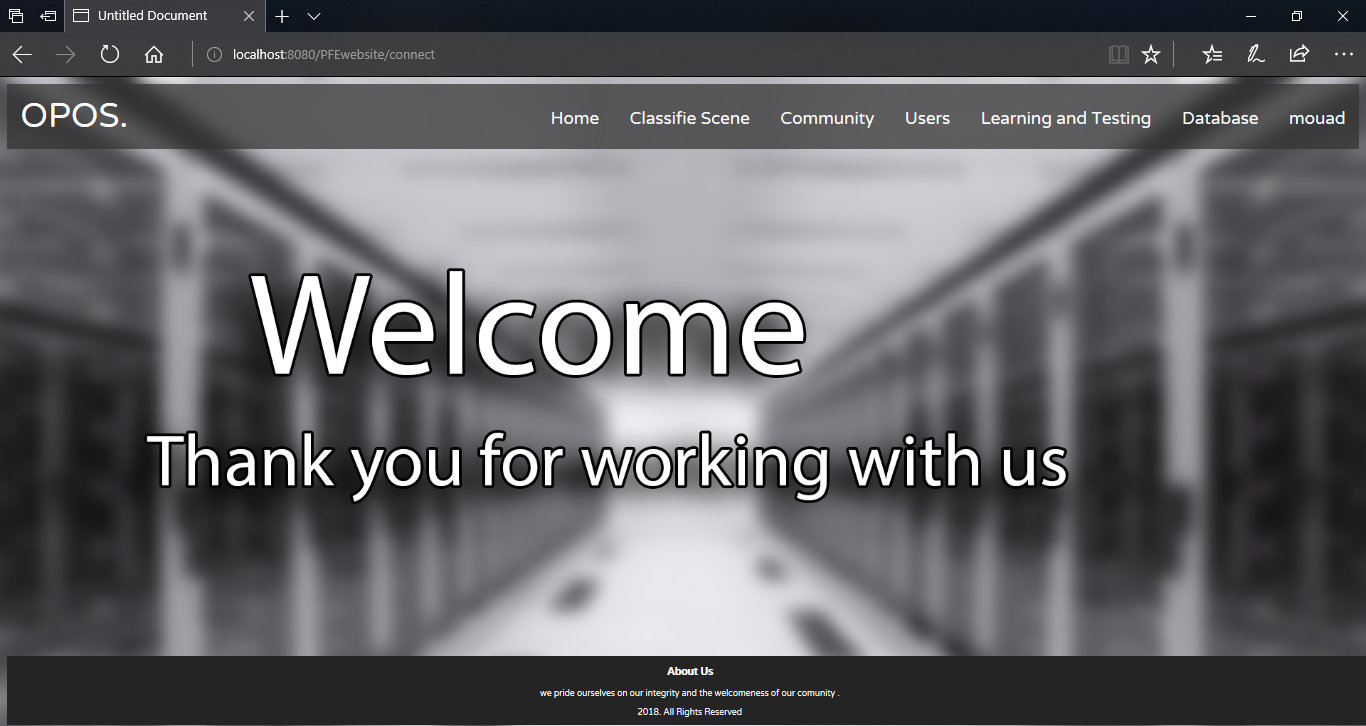


Figure ‎3.13 Pages d'accueil Administrateur.

#### Gestion des utilisateurs

Un administrateur peut gérer les utilisateurs, cela en se rendant dans la page « Users » (voir la figure 3.14) d’où il peut visualiser la liste des utilisateurs, leurs grades, leurs points de vie et s’ils sont connectés ou pas.

Il a la possibilité d’agir sur un utilisateur avec deux actions :

1. Modifier le grade d’un utilisateur en insérant la nouvelle valeur du grade dans la page « modifie grade ».
2. Il peut aussi supprimer un utilisateur depuis la page « delete user ».

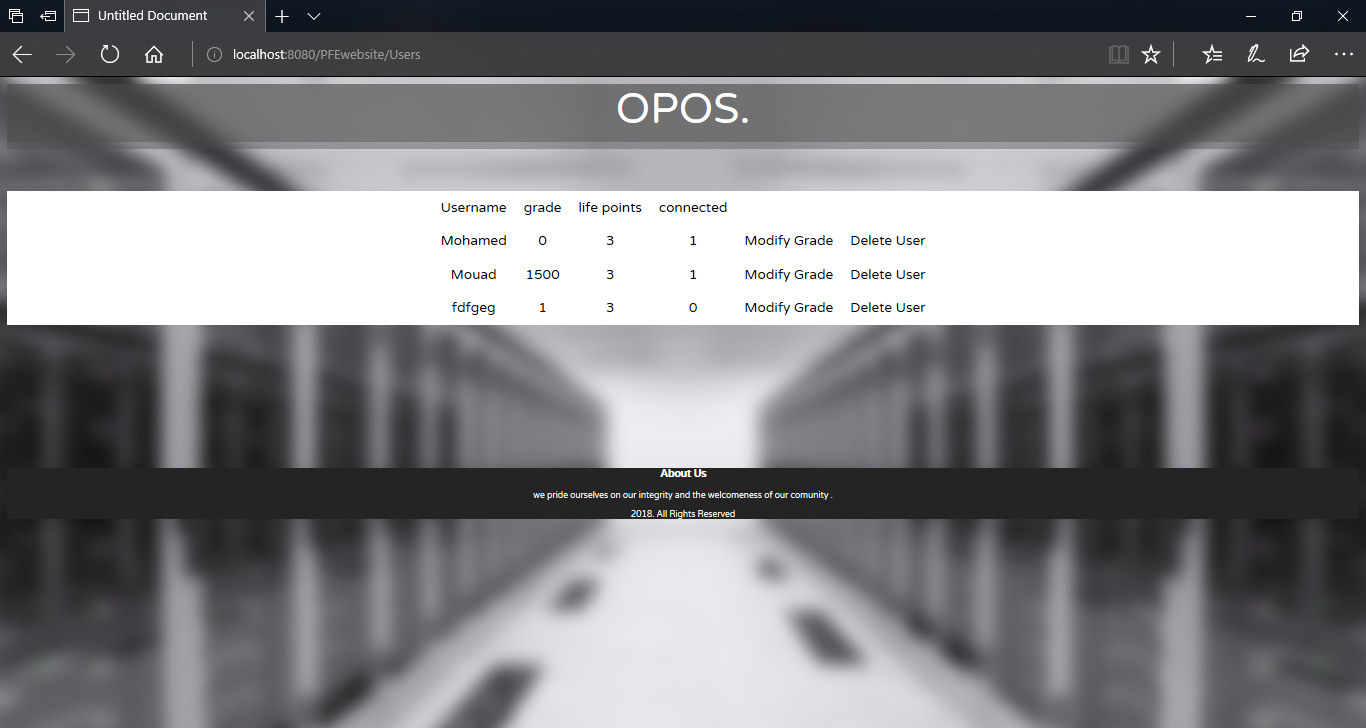


Figure ‎3.14 La liste d'utilisateurs.

#### Lancement de l’apprentissages et test

Dans la page montré par la figure 3.15, l’administrateur peut paramétrer le classificateur avec les paramètres suivants :

|  |  |
| --- | --- |
| paramètre | Description |
| Learn | Permet de lancer l’apprentissage |
| Save | Permet de sauvegarder les données de l’apprentissage si ce dernier est lancé en sélectionnant la case à cocher « learn » |
| Load | Permet de charger les données sauvegardées de l’apprentissage du classificateur. Ceci est primordial car cela va éviter de refaire l’apprentissage déjà fait d’une base de données existante. |
| Test | Permet de lancer le test du classificateur sur labase de données depuis laquelle il a été appris, et visualiser les résultats. |
| Learn from | Offre deux choix :  1- Lancer la classification depuis une base de données implémentée dans le site, à savoir : une instance de notre base de données, SUN397 [1], LabelMe16 [11], MIT Indoor67 [4], etc.  2- Lancer la classification depuis une base de données extérieure (non implémentée dans le site) définie par un lien (chemin du fichier xml). |
| Show classifier data | Ajout la possibilité d'extraction des données du classificateur et permettre aux administrateurs l'exploitation du contenu pour la maintenance |
| Optimize classifier | Permet l’optimisation du classificateur est optimisé avec le seuil d’intolérance (défini dans ‎2.3.1.5) ou non. |

Tableau ‎3.1 Les paramètres de lancement d’apprentissage et test

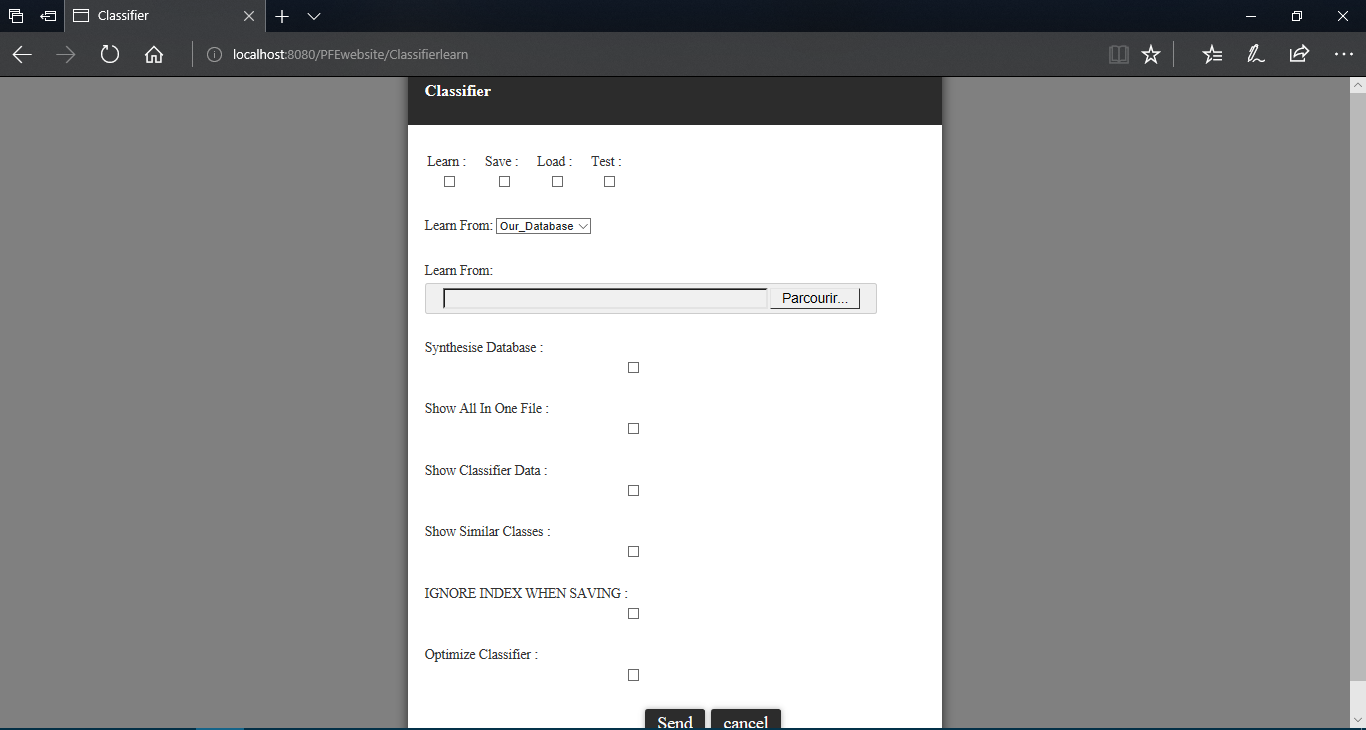


Figure ‎3.15 Lancement de l’apprentissage.

### Implémentation de la base de données

Nous présentons dans la figure 3.16 le schéma UML de notre base de données :

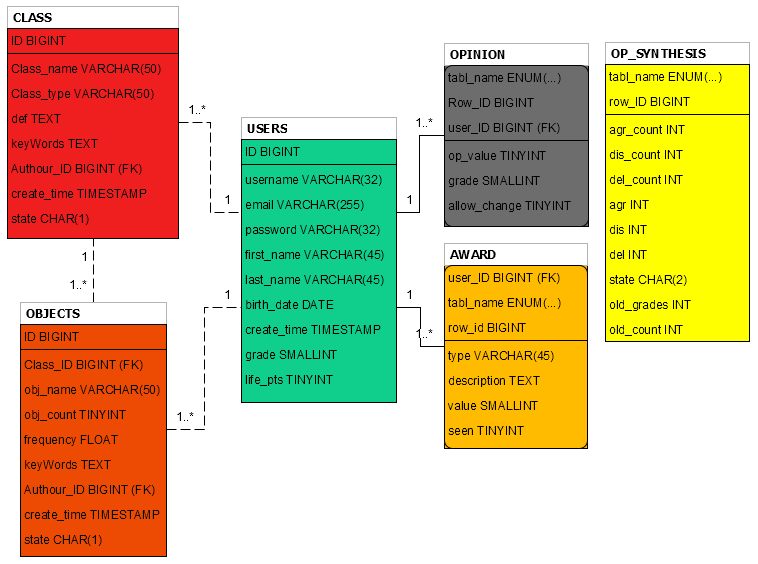


Figure ‎3.16 Schéma UML de notre base de données LRIA\_BD.

Les classes de scènes sont stockées dans la table « CLASS » et les objets liées à celle-ci dans la table « OBJECTS ». Toute opinion d’un utilisateur concernant un objet où une classe de scènes est placée dans la table « OPINION » qui est analysée à chaque fois elle est mise à jour, la synthèse de ces analyses sont ensuite enregistrées dans la table « OP\_SYNTHESIS » dans la quelle résident les procédures de prise de décision de notre système de validation de données. Nous gardons aussi un historique des gratifications et pénalités effectués par le système dans la table « AWARD ».

## Résultats de classification

Nous avons testé notre méthode de classification de scènes en deux étapes :

1. Premièrement,en testant sur les paramètres du classificateur pour trouver celles pour lesquelles nous obtenons les meilleurs résultats.
2. Deuxièmement, en utilisant les meilleurs paramètres sélectionnés, nous testant notre méthode sur plusieurs bases de données reconnues, puis nous présenterons nos résultats en parallèle avec celles d’autres méthodes de classification de scènes dans l’état de l’art.

Les bases de données utilisées dans les tests sont les suivantes :

1. Une instance de notre base de données, notée LRIA\_BD112.
2. La base de données SUN397[1].
3. La base de données LabelMe16[11].
4. La base de données MIT Indoor67[4].

La figure 3.17montrele nombre de catégories de scènes et le nombre d’instances de scènes pour chacune de ces bases de données, ce schéma est présenté dans une échelle logarithmique :

Figure ‎3.17 Dimensions des bases de données utilisées dans les tests.

Nous illustrons aussi dans la figure 3.18 la distribution des instances de scènes dans leurs catégories avec des quartiles, cela pour les bases de données SUN397 [1], LabelMe16 [11] et MIT Indoor67 [4].

Figure ‎3.18 Distribution des instances de scènes dans les bases de données SUN397 [1], LabelMe16 [11] et MIT Indoor67 [4].

D’après la figure 3.18, on peut voir par exemple que 50% des catégories de scènes dan SUN397 [1] ne contiennent pas plus que 16 instances de scènes, ce qui veut dire que, dans le meilleur des cas, notre classificateur va apprendre depuis 14 instances et est testé sur les 4 qui restent. Cela peut nuire à la qualité des résultats de classification à cause du manque de données d’apprentissage et le risque que les données utilisés dans les tests différent fortement de celles d’apprentissage.

Dans ce qui suit, nous présenterons les résultats des tests.

### Tests des paramètres du classificateur

Dans ce qui suit, nous évaluons les résultats avec la « sensibilité » (sensitivity) moyenne, cette mesure peut être interprétée comme la précision ou le rapport des scènes correctement classifiées dans chaque catégorie de scènes, tel que pour chacune de ces catégories

|  |  |  |
| --- | --- | --- |
|  |  | ‎3.1 |

Avec TP et FP sont les valeurs de « vrai positif » et « faux négatif » respectivement.

Les tests ont été faits sur 20% des scènes dans chaque catégorie et pour toutes les bases de données. Aussi, nous avons appliqué des restrictions sur le nombre de scènes dans les autres bases de données pour accélérer les tests dans cette partie, vue qu’on s’intéresse ici uniquement à tester les paramètres. Dans le test de chaque paramètre, nous varions ce dernier en fixant tous les autres.

#### Cardinalité maximale des ensembles d’objets

Ce paramètre affecte lourdement le temps d’apprentissage et la taille des données du classificateur appris. Pour une cardinalité maximale de *n*, le nombre d’ensembles d’objets générée peut être approché a :

|  |  |  |
| --- | --- | --- |
|  |  | ‎3.3 |

Où « *C »*est le nombre de catégories de scènes, « S » le nombre de scènes moyen dans ces derniers et « O » le nombre d'objets moyen dans les scènes.

Ce nombre (formule ‎3.3) est de complexité de (car est un polynôme de degré k), à cause de cela, nous nous sommes limités à tester ce paramètre dans un intervalle de 1 jusqu'à 5.

Figure ‎3.19Résultats des tests pour différentes valeurs de la cardinalité maximale.

Il est clair d’après la figure 3.19que ce paramètre améliore dramatiquement les résultats dans la majorité des cas, mais comme lesrésultats semblent se stagner après la valeur 3, nous choisissons cette dernière comme pivot pour ce paramètreafin de gagner en termes de complexité temporelle et spatiale.

Nous concluons que, comme nous l’espérions, que la mise en valeur des ensembles d'objets joue un rôle indispensable dans la qualité de la classification.

#### Seuil d’intolérance

Nous rappelons que le seuil d’intolérance (définit dans la section‎2.3.1.5) est un paramètre du classificateur utilisé pour son optimisation en terme d’espace et de temps d’apprentissage, cela en éliminant les objets ayant une fréquence inférieure sa valeur.

Figure ‎3.20 Résultats des tests pour différentes valeurs du « seuil d’intolérance ».

D’après la figure 3.20, nous pouvons bien constater que ce paramètre, bien qu’ilsoit très favorable pour l’optimisation, semble dégrader la qualité de la classification, mais cela dépend de la base de données utilisée. L’intérêt de ce paramètre est l’élimination des objets rares qui n’identifient pas réellement leur classe de scènes, l’existence de ces objets est relative aux jeux de données utilisés, ce qui explique la différence des résultats obtenus pour les différentes bases de données.

Nous suggérons alors de choisir un seuil d'intolérance dans l'intervalle [0.0, 0.15] comme les résultats y sont plus au moins stables.

#### Exposant de cardinalité

L’Exposant de cardinalité noté *n* (définit dans ‎2.3.2) est un paramètre de notre formule de masse proposée (voir la formule ‎2.2), il sert à favoriser les ensembles d’objets qui ont les plus grandes cardinalités qui peuvent être plus « indicatifs ».

Figure ‎3.21 Résultats des tests pour différentes valeurs de la « puissance ».

Il est à noter que d'après la figure 3.21 que ce paramètre fait accroitre (même légèrement) les résultats. Ces derniers se stabilisent haut delà de la valeur 7 sur la quelle tombe notre choix.

#### Facteur d'unicité

Ce paramètre appeléα est utilisé pour mettre en valeur les ensembles d’objets propres à une classe de scènes (voir la formule ‎2.3) en élevant leur valeur de masse. Nous rappelons aussi que le cas oùcoïncideavec la celui de la formule ‎2.2 où la notion d’unicité n’était pas prise en charge.

Figure ‎3.22 Résultats des tests pour différentes valeurs du « facteur d’unicité ».

Il clair que d’après la figure 3.22 que l’effet de ce paramètre diffère selon la base de données d’apprentissage, cela peut s’expliquer parle « taux de ressemblance des catégories de scènes »qui certainement différent pour chacune de ces derniers, car plus il y a des catégories similaires, plus elles deviennent confondues, et donc il faut mettre en valeur leurs éléments propres pour mieux les identifier (d’où l’intérêt de ce paramètre). Par exemple, pour la base de données Indoor67 [4] contenant que des scènes d’intérieurs (indoor), ce paramètre raffine toujours les résultats d’avantage, car ce type de scènes présente relativement plus de similarités en termes d’objets.

Mais dans tous les cas, l’introduction de ce paramètre arrive à donner de meilleurs résultats, i.e. la formule‎2.3apporte une amélioration par rapport à la formule ‎2.2. Le choix de ce paramètre reste ouvert suivant la base de données d’étude. Nous choisissons les valeurs suivantes :

1. 12 pour la base de données LabelMe16[11].
2. 30 pour notre base de données.
3. 700 pour les deux bases de données SUN397 [1]et MIT Indoor67[4].

### Comparaison avec quelques méthodes de l’état de ~~l'état~~ l’art

Nous utilisant dans cette section les meilleurs paramètres choisis dans ‎3.4.1, et nous présentons nos résultats en parallèle avec celles obtenus pour d’autres méthodes de classification de scènes rapportées dans la littérature.

Nous commençons par tester les résultats sur différentes instances de notre base de données LRIA\_BD112 avec 112 catégories (un minimum de 5 objets par scènes) de scènes et une « précision » de 50% (voir les paramètres de téléchargement dans la section ‎2.2.2).

Figure ‎3.23 Résultats des tests sur différentes instances de LRIA\_BD de différentes tailles.

~~On peut saisir~~ Il est clair que d’après les résultats de la figure 3.23 que la taille de la base de données LRIA\_BD112 affecte positivement les résultats obtenus, car plus il y a d’instances de scènes, plus l’apprentissage est plus riche, et ainsi la qualité de classification est meilleure.

Nous obtenons une précision de 97.4% pour 2000 instances de scènes.

Les résultats montrés dans les tableaux 3.~~2~~3, 3.34 et 3.45 concernent respectivement la base de données MIT Indoor67[4], SUN397[1] et LabelMe16[11].

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Méthode | (DPM) [29] | (GC) [29] | (SP) [29] | (All)  [29] | **Méthode proposée** | Méthode proposée dans [28] | (OB) [30] |
| Sensibilité [%] | 31.15 | 34.88 | 29.89 | 43.75 | **55,48** | 25.13 | 37.6 |

Tableau ‎3.3~~2~~ Comparaison des résultats sur la base de données MIT Indoor67[4] avec l’état de l’art.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Méthode | Sensibilité [%] | | SUN(HOG) [1] | 27.20 | | SPMSM [39] | 28.20 | | OTC [31] | 34.56 | | **Méthode proposée** | **35,86** | | Meta-classes [32] | 36.80 | | SUN(MKL) [1] | 38.00 | | Zhou et al. [33] | 38.00 | | KCNF [34] | 40.80 | | DeCAF [35] | 40.94 | | Méthode dans  [36] | 47.20 | | OTC+HOG2x2 [37] | 49.60 | | fc7-VLAD [38] | 51.98 | | fc7-FV [38] | 53.00 | | fc8-FV [39] | 54.40 | | DSP [32] | 59.78 | | fc8+fc7+places [39] | 61.72 |   Tableau ‎3.4 Comparaison des résultats sur la base de données SUN397[1]Indoor avec l’état de l’art. | |  |  | | --- | --- | | méthode | Sensibilité[%] | | GIST [42] | 54.0 | | BoW [43] | 55.0 | | SPM [41] | 59.0 | | Object Bank-SVM [40] | 68.0 | | Object Bank-LR [40] | 76.0 | | Wang and Al [44] | 76.0 | | Méthode proposée dans [45] | 83.0 | | Dixit et al [46] | 86.9 | | SPMSM [47] | 87.5 | | Kernel Descriptor [48] | 87.3 | | **Méthode proposée** | **87,8** | | BoW+SPM (Codebook=512) [47] | 88.6 | | MFS [48] | 88.9 | | Object Bank [49] | 89.8 | | KCNF [50] | 89.8 | | Extended MFS [48] | 89.9 |   Tableau ‎3.54 Comparaison des résultats sur la base de données LabelMe16[11]avec l’état de l’art. |

Pour la base de données MIT Indoor67 [4], nous obtenons une sensibilité de 55.48%. Parmi les autres méthodes que nous citons, la meilleure est (All) [29] de Pandey et al.avec une sensibilité de 43.75%, pour cette méthode ils ont utilisé une combinaison de 3 méthodes : des pyramides spatiales (SP) + GIST-color +leur approche basée sur les DPM (deformablepart-based models) qui sont capable de capturer des éléments visuels fréquents et des objets saillants (évidents), en combinaison avec des caractéristiques globales de l’image.

Par contre pour la base de données SUN397 [1], les résultats obtenus n’étaient pas aussi prometteurs, nous obtenons une précision de 35.86%. Gong, Y. et al. [38] arrivent à une précision de 53.0% pour leur méthode fc7-FV, ils ont utilisé un schéma qu’ils appellent « multi-scale orderless pooling » (MOP-CNN), ce schéma extrait les activations CNN pour les patchs locaux à plusieurs niveaux d'échelle, effectue une mise en pool VLAD sans ordre de ces activations à chaque niveau séparément et concatène le résultat.

Finalement, pour la base de données LabelMe16 [11], notre approche engendre une précision de 87.8%.D’après la figure 3.18, cette base de données propose la distribution la plus riche de scènes, ce qui explique la qualité des résultats. Une des plus récentes méthodescitées pour LabelMe16 [11] est Kernel Descriptor [48] de Li et al(2017) avec une précision de 87.3%.Li et al.[48] ont proposés deux autres méthodes qui dépassent la nôtre : MFS et Extended MFSavec une précision de 88.9% et 89.9% respectivement, Li et al. [48] ont utilisé des « représentations de patches discriminatives » à l’aide de réseaux de neurones et une architecture hybride dans laquelle le SM(semantic manifold) est construite au-dessus de CNNs(deep learning) multi-échelles.

## Conclusion

Dans ce chapitre, nous avons présenté les outils que nous avons utilisés pour la réalisation des différentes parties de notre travail que nous avons ~~aussi~~ abordés au sein de ce chapitre, à savoir, notre site web SCAFFOLD, base de données LRIA\_BD et classificateur de scènes. Nous avons aussi présenté les résultats que nous avons obtenus en testant notre approche de classification de scènes sur plusieurs bases de données et nous avons également comparé ces résultats avec l’état de l’art.

Les résultats obtenus montrent que la méthode que nous avons proposé, malgré sa simplicité, offre une qualité très acceptable pour la classification de scènes ~~via~~ ~~leurs~~ basée sur objets.

Conclusion générale

A travers ce travail, notre objectif était de contribuer à l’avancement de la recherche dans le domaine reconnaissance de scènes, en proposant une base de données nommée LRIA\_BD fondée sur le travail collaboratif d’une communauté en ligne via un site web que nous appelons SCAFFOLD, cela dans le but d’enrichir le jeux de données de cette base de données et ainsi garantir sa cohérence en introduisant un système de validation de données. Cette base de données comporte des catégories de scènes, et dans chacune, des objets caractérisés par leurs fréquence et nombre d’occurrence dans cette catégorie de scènes.

La base de données LRIA\_BD peut être instanciée en plusieurs bases de données selon des paramètres établis, notamment la taille de la base de données enfant, nombre de catégories de scènes et le nombre minimal d’objets par scène.

Nous avons aussi présenté dans ce mémoire une méthode ~~proposée~~ pour la classification de scènes, basée sur la théorie des fonctions de croyance. La méthode que nous propos~~ant~~ons ~~concise~~, étant donnée une liste d'objets d’une scène, de proposer une classification de cette scène avec un degré de croyance ou certitude et ce pour remédier aux limites des probabilités concernant le problème de..... .

Pour cela, nous avons conçu un classificateur de scènes capable, depuis une base de données sous contenant des scènes étiquetées, de faire un apprentissage en générant à partir de chaque scène, toutes combinaisons d'objets de cardinalités inférieurs à un seuil, puis extraire les fréquences de ces objets et d’autres informations pour leurs attribuer des masses.

Par la suite, nous avons testé la qualité de la méthode proposée, d’abord en variant les paramètres du classificateur notamment la cardinalité maximale des ensembles d'objets, ensuite, en sélectionnant les meilleurs paramètres déduits. Nous avons également testé notre approche sur trois bases de données parmi les plus répondues, à savoir MIT Indoor67 [4], LabelMe16 [11] et SUN397 [1].Également, nous comparons nos résultats avec d’autres méthodes de classification de scènes rapportées dans la littérature.

Finalement, la qualité de la plupart des résultats obtenus certifient l’aptitude de la méthode proposée, malgré sa simplicité, pour une bonne classification de scènes en se basant uniquement sur ~~la détection~~ objets.

Perspectives au moins Une.....

bibliographie

|  |  |
| --- | --- |
| [1] | Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., & Torralba, A. (2010). Sun database: Large-scale scene recognition from abbey to zoo. IEEE conference on Computer vision and pattern recognition, 3485-3492. |
| [2] | «SUN Database» [En ligne]. Available: https://groups.csail.mit.edu/vision/SUN/. |
| [3] | «MIT Indoor» [En ligne]. Available: http://web.mit.edu/torralba/www/indoor.html. |
| [4] | A. Quattoni, and A.Torralba , Recognizing Indoor Scenes. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009. |
| [5] | Torralba, A., B.C. Russell, and J. Yuen, “LabelMe: Online Image Annotation and Applications.” Proceedings Of the IEEE 98.8 (2010) : 1467-1484. Copyright © 2010, IEEE. |
| [6] | «LabelMe,» [En ligne]. Available: http://labelme.csail.mit.edu/Release3.0/. |
| [7] | «Wikipédia,» [En ligne]. Available: https://www.wikipedia.org. |
| [8] | «Wiktionary,» [En ligne]. Available: https://en.wiktionary.org. |
| [9] | «Wiktionary,» [En ligne]. Available: https://www.wiktionary.org/. |
| [10] | «Google Translate,» [En ligne]. Available: https://translate.google.com. |
| [11] | B. C. Russell, A. Torralba, K. P. Murphy, W. T. Freeman, LabelMe: a database and web-based tool for image annotation. International Journal of Computer Vision, pages 157-173, Volume 77, Numbers 1-3, May, 2008. |
| [12] | «Vandalism on Wikipedia,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Vandalisme. |
| [13] | A. Vailaya, A. Jain, and H. Zhang, “On image classification: city vs. landscapes,” Pattern Recog., vol. 31, pp. 1921–1935, 1998. |
| [14] | E. Chang, K. Goh, G. Sychay, and G. Wu, “Cbsa: Content-basedsoft annotation for multimodal image retrieval usingbayespointmachines,” IEEE Trans. on Circuits and Systems for Video Technology,vol. 13, pp. 26–38, 2003. |
| [15] | I. Ulrich and I. Nourbakhsh, “Appearance-based place recog. fortopological localization,” in IEEE Int. Conf. on Rob. And Automation(ICRA), 2000. |
| [16] | A. Oliva and A. Torralba, “Modeling the shape of the scene: a holistic representation of the spatial envelope,” Int. Journal of Comp. Vision, vol. 42, pp. 145–175, 2001. |
| [17] | C. Siagian and L. Itti, “Rapid biologically-inspired scene classification using features shared with visual attention,” IEEE Trans. on PatternAnalysis and Machine Intell., vol. 29, no. 2, pp. 300–312, 2007. |
| [18] | L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” IEEE Trans. on Pattern Analysisand Machine Intell., vol. 20, no. 11, pp. 1254–1259, 1998. |
| [19] | A. Vailaya, M. Figueiredo, A. Jain, and H. Zhang, “Content-based hierarchical classification of vacation images,” in IEEE Int. Conf. onMultimedia Computing and Systems (ICMCS), 1999. |
| [20] | M. Szummer and R. Picard, “Indoor-outdoor image classification,” in IEEE Int. Conf. on Comp. Vision (ICCV), Workshop on Content-basedAccess of Image and Video atabases, 1998. |
| [21] | S. Paek and S. Chang, “A knowledge engineering approach for image classification based on probabilistic reasoning systems,” in IEEE Int.Conf. on Multimedia and Expo (ICME), 2000. |
| [22] | A. Mojsilovic, J. Gomes, and B. Rogowitz, “Isee: Perceptual features for image library navigation,” in SPIE Human vision and electronicimaging Conf., 2002. |
| [23] | C. Fredembach, M. Schroder, and S. Susstrunk, “Eigenregions for image classification,” IEEE Trans. on Pattern Analysis and MachineIntell., vol. 26, no. 12, pp. 1645–1649, 2004. |
| [24] | L. Fei-Fei and P. Perona, “A bayesian hierarchical model for learning natural scene categories,” in IEEE Int. Conf. on Comp. Vision andPattern Recog. (CVPR), 2005. |
| [25] | A. Bosch, A. Zisserman, and X. Mu˜noz, “Scene classification via plsa,” in European Conf. on Comp. Vision (ECCV), 2006. |
| [26] | S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in IEEE Int. Conf. on Comp. Vision and Pattern Recog. (CVPR), 2006. |
| [27] | T. Denœux, Introduction to Belief Functions. Lecture notes, Université de Technologie de Compiègne, Compiègne, France, 2011. |
| [28] | Sadeghi, Mohammad Amin, and Ali Farhadi. "Recognition using visual phrases." In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. (2011):1745-1752. |
| [29] | Pandey, Megha, and Svetlana Lazebnik. "Scene Recognition and Weakly Supervised Object Localization with Deformable Part-based Models." 2011 International Conference on Computer Vision, (2011). |
| [30] | Li, Li-Jia, Hao Su, Yongwhan Lim, and Li Fei-Fei. "Object Bank: An Object-Level Image Representation for High-Level Visual Recognition." International Journal of Computer Vision 107, no. 1 (2013) |
| [31] | Grauman, K., & Darrell, T. (2005). The pyramid match kernel: Discriminative classification with sets of image features. Tenth IEEE International Conference on Computer Vision, (2),1458-1465. |
| [32] | Bergamo, A., & Torresani, L. (2014). Classemes and other classifier-basedfeatures for efficient object categorization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(10), 1988-2001. |
| [33] | Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., & Oliva, A. (2014). Learning deepfeatures for scene recognition using places database. Advances in neural information processing systems, 487-495. |
| [34] | Song, X., Jiang, S., Herranz, L., Kong, Y., & Zheng, K. (2016). Categoryco-occurrence modeling for large scale scene recognition. Pattern Recognition, (59), 98-111. |
| [35] | Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. (2014). Decaf: A deep convolutional activation feature for generic visual recognition. International conference on machine learning, 647-655. |
| [36] | Sánchez, J., Perronnin, F., Mensink, T., & Verbeek, J. (2013). Image classification with the fishervector: Theory and practice. International journal of computer vision, 105(3), 222-245. |
| [37] | Margolin, R., Zelnik-Manor, L., & Tal, A. (2014). Otc: A novel local descriptor for scene classification. European Conference on Computer Vision, 377-391. |
| [38] | Gong, Y., Wang, L., Guo, R., & Lazebnik, S. (2014). Multi-scale order less pooling of deep convolutional activation features. European conference on computer vision, 392-407. |
| [39] | Elbaşi, E. (2013). Fuzzylogic-based scenario recognition from video sequences. Journal of applied research and technology, 11(5), 702-707. |
| [40] | Li, L. J., Su, H., Fei-Fei, L., & Xing, E. P. (2010). Object bank: A high-level image representation for scene classification and semantic feature sparsification. In Advances in neural information processing systems, 1378-1386. |
| [41] | Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In Proceeding CVPR '06 Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2169-2178. |
| [42] | Oliva, A., & Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. International journal of computer vision, 42(3), 145-175. |
| [43] | Csurka, G., Dance, C., Fan, L., Willamowski, J., & Bray, C. (2004). Visual categorization with bags of keypoints. In Workshop on statistical learning in computer vision (ECCV), 1-22. |
| [44] | Chong, W., Blei, D., & Li, F. F. (2009). Simultaneous image classification and annotation. In Computer Vision and Pattern Recognition (CVPR), 1903-1910. |
| [45] | Maji, S., & Berg, A. C. (2009). Max-margin additive classifiers for detection. In IEEE 12th International Conference on Computer Vision (ICCV), 40-47. |
| [46] | Dixit, M., Rasiwasia, N., & Vasconcelos, N. (2011). Adapted gaussian mixtures for image classification. In Computer Vision and Pattern Recognition (CVPR), 937-943 |
| [47] | Kwitt, R., Vasconcelos, N., & Rasiwasia, N. (2012). Scene recognition on the semantic manifold. In European Conference on Computer Vision (ECCV), 359-372. |
| [48] | Song, X., Jiang, S., & Herranz, L. (2017). Multi-Scale Multi-Feature Context Modeling for Scene Recognition in the Semantic Manifold. In IEEE Transactions on Image Processing, 26(6), 2721-2735. |
| [49] | Li, L. J., Su, H., Lim, Y., & Fei-Fei, L. (2014). Object bank: An object-level image representation for high-level visual recognition. International journal of computer vision, 107(1), 20-39. |
| [50] | Song, X., Jiang, S., Herranz, L., Kong, Y., & Zheng, K. (2016). Category co-occurrence modeling for large scale scene recognition. In Pattern Recognition, 59, 98-111. |

**Choisisser le meme format pour vos réferencs**