

Projet de Multimedia Information Retrieval

Rapport
I-ILIA-014

Hugo Venturoso
Yorick Estievenart
Groupe 16

Année académique 2022-2023
Sciences de l'Informatique
Janvier 2022



Faculté
des Sciences

UMONS
Université de Mons

Table des matières

1	Introduction	2
2	Manuel d'utilisation	2
2.1	Installation et exécution	2
2.2	Description de l'application	2
2.2.1	Écran d'indexation	2
2.2.2	Écran de recherche	3
3	Données utilisées	4
4	Motivation et justification	4
4.1	Choix des descripteurs	4
4.1.1	Combinaison de descripteurs	5
4.1.2	SIFT	5
5	Evaluation	6
5.1	Comparaison des descripteurs	6
5.2	Performances du système	7
5.3	Précision du moteur de recherche	7
6	Conclusion	9

1 Introduction

L'objectif principal de ce projet est de développer une application pour indexer une base de données d'images et effectuer une recherche par rapport à une image de requête. Ce rapport présente une description de cette application avec un manuel d'installation et d'utilisation. Il présente également les données utilisées tout au long du projet pour le tester et l'évaluer. Dans les sections suivantes, les motivations et justifications de nos choix pour réaliser cette application seront présentées et ce rapport se terminera par une évaluation du système.

2 Manuel d'utilisation

2.1 Installation et exécution

Pour pouvoir utiliser l'application, il faut préalablement installer les librairies nécessaires au bon fonctionnement de celle-ci. Pour cela, il est possible d'utiliser l'environnement conda créé lors des travaux pratiques "Mir_Classic_Features" ou celui qui est fourni dans le code source de notre application. Pour utiliser ce dernier, il suffit de se mettre à la source de notre application et, si vous êtes sous windows, d'exécuter la commande suivante :

```
conda env create -f conda_env_windows.yml
```

Sinon, il faudra exécuter la commande suivante :

```
conda env create -f conda_env_mac.yml
```

Ensuite, il suffit d'activer cet environnement :

```
activate project_mir
```

Pour exécuter notre application, il faut se mettre dans le dossier *src* et exécuter la commande suivante. L'interface graphique sera dès lors lancée.

```
python main.py
```

2.2 Description de l'application

Au lancement de l'application, l'écran de menu s'affiche et vous disposez de trois options. Vous pouvez cliquer sur le bouton "Indexation" pour être dirigé vers la page permettant d'indexer la base de données avec les descripteurs choisis. Vous pouvez également cliquer sur le bouton "Recherche" pour effectuer une recherche sur une image choisie comme requête. Ces deux pages sont décrites plus en détail dans les sections suivantes. Enfin, un bouton "Quitter" vous permet de mettre fin à l'exécution de l'application.

2.2.1 Écran d'indexation

L'écran d'indexation peut être visualisé à la figure 1. Ce design est tiré des travaux pratiques, étant plutôt simple et efficace, nous avons choisi de le garder. Voici les différentes fonctionnalités offertes par cette fenêtre :

- **Choix de descripteurs (simples)** : Pour indexer en utilisant un descripteur, la procédure est simple. Tout d'abord, il faut charger la base de données, puis cocher le descripteur souhaité. Enfin, il suffit d'appuyer sur le bouton "Calculer les descripteurs" et le calcul est effectué. Le résultat est stocké dans le dossier *output/[Nom du descripteur]*.

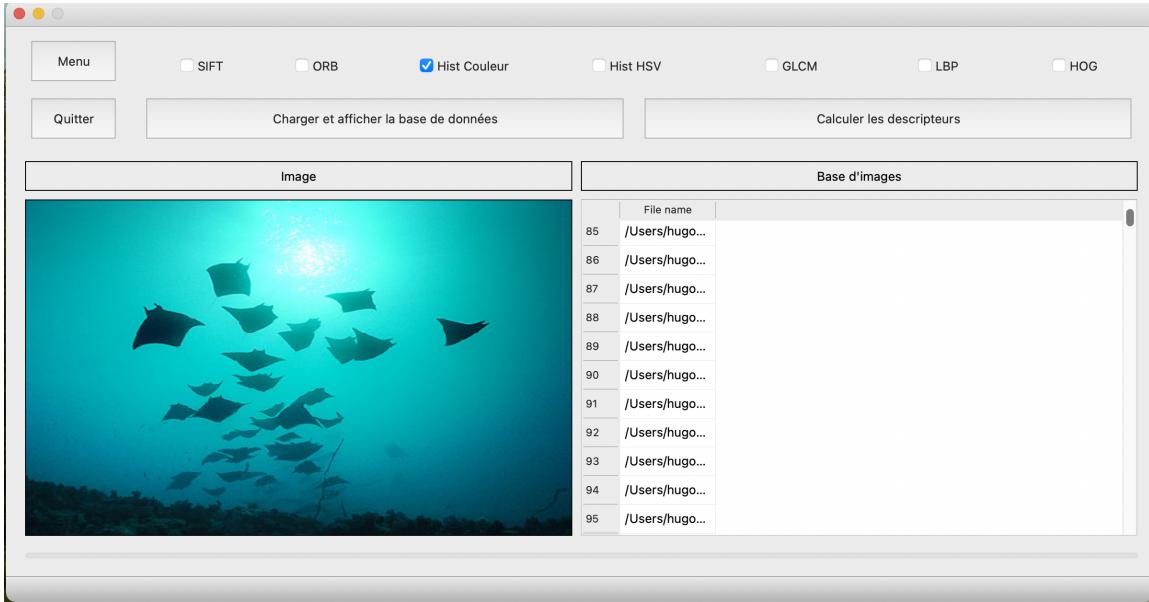


FIGURE 1 – Exemple d'affichage de la page d'indexation.

- **Choix de descripteurs (combinés)** : Si plusieurs descripteurs sont cochés en même temps, le résultat sera la combinaison de ceux choisis. Les détails du fonctionnement de cette combinaison et sa logique intrinsèque seront expliqués dans la suite de ce rapport. Le résultat est stocké dans le dossier *output/[descripteur1 à descripteurN]*, séparés par des *underscore*.

Ainsi, si nous voulons indexer l'ensemble de la base de données sans combiner les descripteurs, il suffit de les cocher un par un et d'exécuter le calcul pas à pas pour chacun des descripteurs souhaités.

2.2.2 Écran de recherche

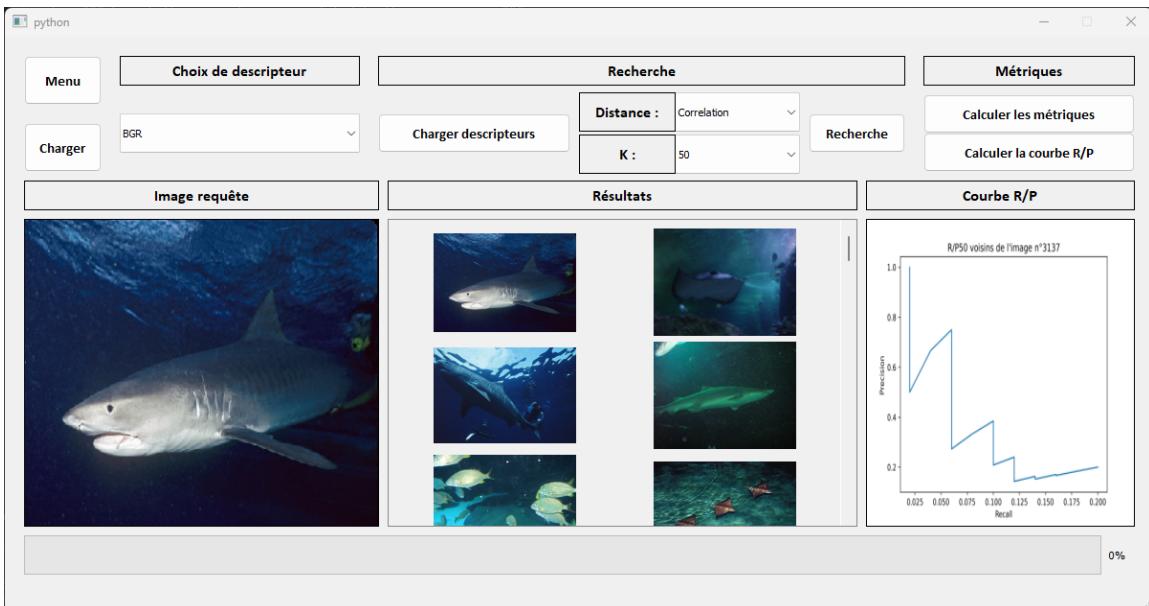


FIGURE 2 – Exemple d'affichage de la page de recherche.

L'écran de recherche peut être visualisé sur la figure 2. Ce design suit celui utilisé lors des travaux

pratiques. Sur cet écran, nous pouvons remarquer différents boutons :

- **Charger** : permet de sélectionner une image dans la base de données qui sera utilisée comme image de recherche.
- **Charger descripteurs** : permet de charger les descripteurs créés dans la partie indexation. Pour ce faire, vous devez avoir fait un choix de descripteurs dans le coin supérieur gauche.
- **Distance** : Une fois que les descripteurs ont été choisis et qu'une image de requête a été sélectionnée, plusieurs choix seront affichés dans cette zone pour permettre la sélection d'une fonction de calcul de similarité.
- **K** : lorsqu'une image de requête a été choisie, cette zone permet de choisir le nombre d'images à récupérer lors de la recherche. Notez que la valeur qui n'est pas un nombre entier correspond au nombre maximum d'images pour la classe concernée.
- **Recherche** : permet de rechercher les images les plus similaires à l'image de requête selon la fonction de similarité.
- **Calculer les métriques** : permet d'obtenir les cinq métriques demandées (Rappel (R), Précision (P), Average Precision (AP), Mean Average Precision (MaP) et R-Precision).
- **Calculer la courbe R/P** : permet de calculer et d'afficher la courbe rappel-précision.

3 Données utilisées

Pour évaluer notre système, nous avons travaillé sur la base de données "Animaux" contenant 5 classes d'animaux. Chaque classe est elle-même divisée en 6 autres classes, soit un total de 30 classes. Cette base de données contient 4505 images. Les images utilisées comme requêtes pour tester le système de recherche sont citées dans la table 1.

Indice requête	Classe	Sous-classe	Images
R1	Araignées	Tarantula	0_5_araignees_tarantula_795
R2	Araignées	Gardenspider	0_4_araignees_gardenspider_631
R3	Araignées	Wolfspider	0_1_araignees_wolfspider_259
R4	Chiens	Siberianhusky	1_0_chiens_Siberianhusky_849
R5	Chiens	Chihuahua	1_3_chiens_Chihuahua_1315
R6	Chiens	Labradorretriever	1_1_chiens_Labradorretriever_1054
R7	Oiseaux	Greatgreyowl	2_2_oiseaux_greatgreyowl_2092
R8	Oiseaux	Robin	2_4_oiseaux_robin_2359
R9	Oiseaux	Bluejay	2_3_oiseaux_bluejay_2232

TABLE 1 – Images requêtes utilisées pour évaluer le système.

Il est important de noter que nous considérons comme pertinente une image qui appartient à la même sous-classe. En effet, il nous a semblé logique que si l'image requête est un Chihuahua, obtenir une image de Labrador ne semble pas pertinent même si les deux images appartiennent à la même classe. Nous considérons donc toutes les sous-classes et pas uniquement les classes.

4 Motivation et justification

4.1 Choix des descripteurs

Pour le choix des descripteurs, nous avons choisi de n'implémenter que ceux que nous avons vus et analysés pendant le cours et les sessions pratiques. Dans cette section, nous préciserons

quelques détails intéressants à noter en rapport avec eux.

4.1.1 Combinaison de descripteurs

Nous avons implémenté la possibilité de combiner différents descripteurs entre eux. En théorie, il est donc possible de combiner tous les descripteurs dans un seul fichier de sortie. Cependant, cela n'est pas forcément pertinent. Il est plus pertinent de combiner des descripteurs qui sont déjà similaires les uns aux autres, citons par exemple BGR et HSV : Ces deux descripteurs fonctionnent de manière complémentaire. La représentation BGR est utilisée pour représenter les couleurs d'une image en utilisant la valeur numérique de chaque canal de couleur. Cependant, cette représentation peut être sensible à la variation de l'éclairage et de la luminosité de l'image. La représentation HSV, d'autre part, sépare la couleur de la luminosité et de la saturation. La valeur de teinte (Hue) décrit la couleur pure, la saturation (Saturation) décrit la pureté de la couleur, et la valeur (Value) décrit la luminosité. Cette représentation est plus robuste aux variations d'éclairage et de luminosité. Combiner ces deux représentations pourrait donc permettre d'avoir un descripteur en sortie plus efficace.

D'un point de vue théorique donc, la combinaison de descripteurs semble être intéressante. Cependant, en pratique, cette amélioration potentielle n'est pas présente. Cela est lié au fait que les descripteurs utilisent des approches différentes pour extraire les caractéristiques de l'image, ils peuvent donc ne pas être complémentaires. Combiner BGR et HSV semble plausible, mais cela est moins évident pour certaines combinaisons. Aussi, il existe différentes façons de combiner les descripteurs, chacune ayant ses propres avantages et inconvénients. L'une des approches les plus simples est de concaténer les descripteurs ensemble, c'est-à-dire de les combiner en un seul vecteur. Cependant, cette approche entraîne une perte d'informations car elle ne tient pas compte des relations entre les descripteurs. C'est cette approche qui a été utilisée dans ce projet.

4.1.2 SIFT

Sur les 4505 images de la base de données, 10 n'ont pas pu être indexées par SIFT. Cela signifie qu'aucun point clé intéressant n'a pu être extrait de ces images. Cela peut être dû à plusieurs raisons, notamment au fait que l'image est soit trop simple, soit trop floue et que par conséquent aucun point clé n'est extrait. Une analyse rapide nous permet de comprendre quelles sont les images qui posent problème. Une image problématique peut être vue dans la figure 3. Les 9 autres images problématiques sont similaires.



FIGURE 3 – Photo posant problème.

5 Evaluation

5.1 Comparaison des descripteurs

Avant de pouvoir évaluer notre système, nous avons dû choisir les descripteurs les plus performants en général. Notons que nous considérons qu'un descripteur est meilleur qu'un autre s'il présente un meilleur compromis rappel/précision. Ces deux métriques sont rappelées dans la section suivante. Pour comparer ces descripteurs, nous avons examiné les courbes de rappel/précision pour une image de requête (R4) prise arbitrairement en utilisant la distance euclidienne, sauf pour SIFT et ORB où nous avons utilisé la force brute. Pour cette image, nous avons affiché les 100 images les plus pertinentes.

La raison pour laquelle nous n'avons pas testé plus de fonctions de distance est simplement due à des contraintes de temps. Sinon, nous aurions dû tester tous les descripteurs pour plusieurs images de requête ainsi que pour plusieurs fonctions de distance, ce qui aurait pris un temps considérable. Les résultats obtenus sont visibles sur la figure 4. Dans cette figure, nous pouvons voir que le descripteur VGG16 fournit les meilleurs résultats puisque l'idéal se trouve dans la zone supérieure droite de chaque courbe. Les descripteurs LBP et ORB fournissent également des valeurs satisfaisantes. Ces trois descripteurs seront donc considérés dans la suite comme nos trois meilleurs descripteurs.

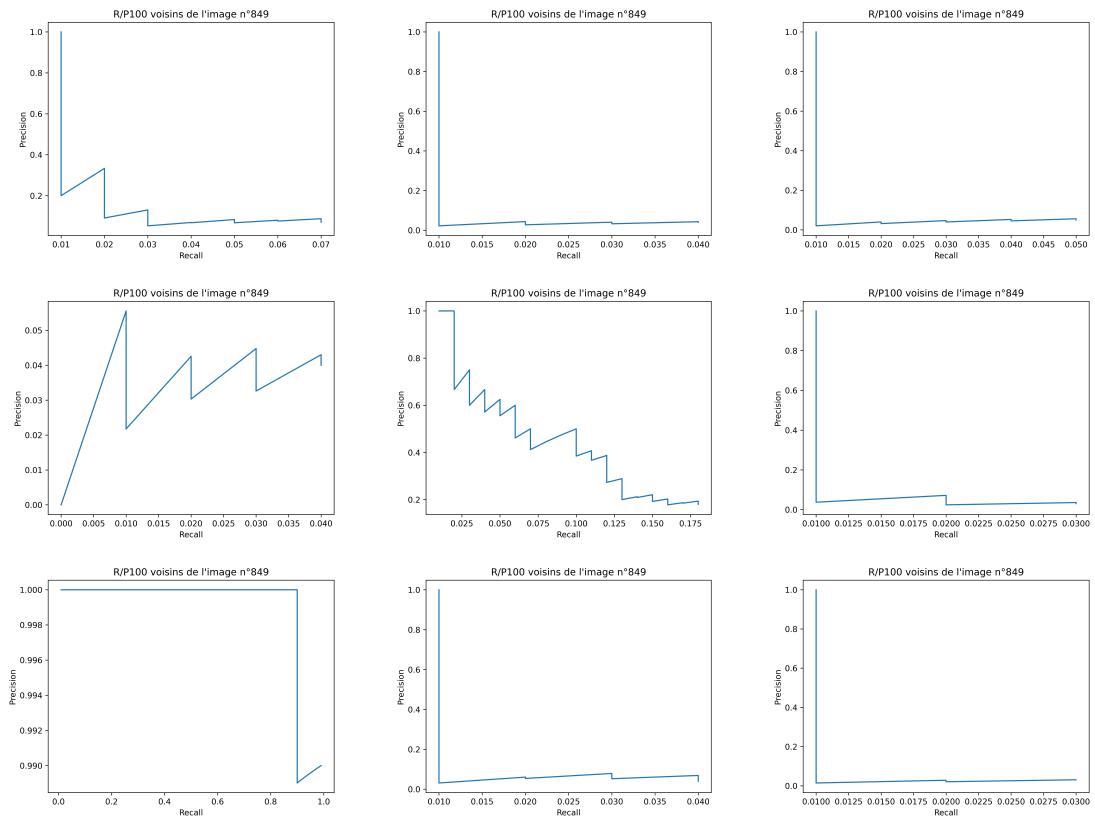


FIGURE 4 – Comparaison des courbes de rappel/précision pour chaque descripteurs (dans l'ordre de gauche à droite et de bas en haut : BGR, HSV, GLCM, LBP, ORB, HOG, VGG16, SIFT, BGR_HSV)

Une autre chose que nous avons voulu voir par rapport aux descripteurs est leur taille en mémoire. En effet, compte tenu de la taille de la base de données qui contient 4505 images, une question pertinente est l'espace utilisé par les descripteurs en mémoire. Ces différentes tailles sont visibles dans le tableau 2. Nous pouvons voir que SIFT et ORB prennent une quantité considérable

d'espace et ne sont pourtant pas les meilleurs descripteurs. VGG16 est beaucoup plus léger et fournit pourtant les meilleurs résultats.

Descripteurs	Espace mémoire
BGR	85.7 MB
GLCM	754 KB
HOG	680 MB
HSV	85.7 MB
LBP	28.6 MB
ORB	1.59 GB
SIFT	2.92 GB
VGG16	111 MB
BGR_HSV	171 MB

TABLE 2 – Espace mémoire utilisé par les descripteurs.

5.2 Performances du système

La première chose que nous avons voulu évaluer par rapport à notre système concerne les descripteurs créés lors de l'indexation et utilisés lors de la recherche. Nous avons voulu voir, pour les 3 meilleurs d'entre eux, le temps d'exécution nécessaire pour les indexer ainsi que pour effectuer la recherche d'une image. Les résultats peuvent être visualisés dans le tableau 3. Le temps de recherche moyen par image est calculé sur un total de 10 requêtes et en affichant les 100 images les plus pertinentes. Les temps sont exprimés en secondes. Nous pouvons voir que VGG16 est très rapide pour la recherche mais prend un temps considérable pour l'indexation. À contrario, LBP et ORB sont nettement plus rapides pour indexer la base de données. Au vu de ces résultats, nous pouvons donc considérer que LBP offre le meilleur compromis entre temps d'indexation et temps de recherche.

Meilleurs descripteurs	Nom des descripteurs	Temps d'indexation (s)	Temps de recherche moyen par image (s)
Descripteur N°01	VGG16	2340	0.84828517
Descripteur N°02	LBP	97.67	1.74434592
Descripteur N°03	ORB	76.21	12.66270269

TABLE 3 – Mesures de performances d'indexation et recherche.

5.3 Précision du moteur de recherche

Nous avons ensuite voulu évaluer notre système par rapport à la précision obtenue pour la recherche des images utilisées comme requêtes. Pour ce faire, nous avons examiné les 50 et 100 premières images retournées par le moteur de recherche pour chaque image utilisée comme requête décrite dans le tableau 1. Pour chacune de ces images, nous les comparons à 4 métriques qui sont le Rappel (R), la Précision (P), l'Average Precision (AP), et la Mean Average Precision (Map). Les formules pour ces 4 métriques sont données ci-dessous.

$$R = \frac{\text{Nombre d'images pertinentes récupérées}}{\text{Nombre total d'images pertinentes}} \quad (1)$$

$$P = \frac{\text{Nombre d'images pertinentes récupérées}}{\text{Nombre total d'images récupérées}} \quad (2)$$

$$AP = \frac{1}{N} \sum_{i=0}^N P_i \text{ où } N = \text{Nombre d'images récupérées} \quad (3)$$

$$MAP = \frac{1}{R} \sum_{i=0}^R AP_i \text{ où } R = \text{Nombre de requête effectuée} \quad (4)$$

Notez que notre système fournit également une cinquième métrique, appelée R-Précision, mais que nous n'affichons pas ici. Ceci est dû au fait que chaque sous-classe contient environ 150 images. Il y a donc environ 150 images pertinentes pour une image de requête. Mais nous n'affichons que les 50 et 100 premières. La R-Précision est donc équivalente à la Précision. Nous avons également utilisé 2 fonctions de calcul de distance différentes pour pouvoir faire une comparaison. Il s'agit de la distance euclidienne et de la distance chi carré et elles ont été choisies arbitrairement. Les résultats de l'évaluation de notre système sont visibles dans la table 4 en utilisant la distance euclidienne et dans la table 5 en utilisant la distance chi carré. Le descripteur utilisé pour effectuer la recherche est celui qui se trouve en haut du tableau présenté dans la section 5.3, c'est-à-dire VGG16.

Indice requête	R		P		AP		MaP	
	T50	T100	T50	T100	T50	T100	T50	T100
R1	1	0.94	1	0.94	1	0.990580	1	0.990580
R2	0.36	0.34	0.36	0.34	0.366407	0.354076	0.683203	0.672328
R3	0.94	0.86	0.94	0.86	0.935090	0.917230	0.767166	0.753962
R4	1	0.99	1	0.99	1	0.998951	0.825374	0.815210
R5	0.98	0.91	0.98	0.91	0.997914	0.978480	0.859882	0.847864
R6	0.96	0.85	0.96	0.85	0.975874	0.941472	0.879214	0.863465
R7	1	1	1	1	1	1	0.896469	0.882970
R8	1	1	1	1	1	1	0.909410	0.897599
R9	1	1	1	1	1	1	0.919476	0.908977

TABLE 4 – Mesures de performance de recherche avec les Top 50 et les Top 100 pour chaque mesure (en utilisant la fonction de distance euclidienne).

Sur ces deux tables, on peut voir que ce descripteur fournit de très bons résultats. Il faut noter que les images sont recherchées les unes après les autres. Il est donc normal que MaP soit égal à AP au début, et diverge lors de la recherche des autres images. Nous pouvons également constater que le rappel et la précision donnent les mêmes valeurs, ce qui est logique puisque nous regardons les valeurs à la fin. Par conséquent, le nombre total d'images pertinentes est équivalent au nombre total d'images récupérées, et donc le rappel et la précision ont la même formule.

Indice requête	R		P		AP		MaP	
	T50	T100	T50	T100	T50	T100	T50	T100
R1	1	0.94	1	0.94	1	0.990619	1	0.990619
R2	0.42	0.38	0.42	0.38	0.369966	0.371836	0.684983	0.681227
R3	0.86	0.81	0.86	0.81	0.888537	0.877440	0.752834	0.746632
R4	1	1	1	1	1	1	0.814625	0.809974
R5	0.98	0.93	0.98	0.93	0.997470	0.983994	0.851194	0.844778
R6	0.96	0.83	0.96	0.83	0.975624	0.938539	0.871932	0.860404
R7	1	1	1	1	1	1	0.890228	0.880347
R8	1	1	1	1	1	1	0.903949	0.895303
R9	1	1	1	1	1	1	0.914621	0.906936

TABLE 5 – Mesures de performance de recherche avec les Top 50 et les Top 100 pour chaque mesure (en utilisant la fonction de distance chi carré).

6 Conclusion

Dans ce rapport, nous avons fourni une description de notre application d’indexation et de recherche multimédia. Nous avons décrit le processus d’installation et d’exécution de l’application, ainsi que les fonctionnalités de l’écran d’indexation et de l’écran de recherche. Nous avons également discuté des données utilisées pour alimenter le système. Nous avons ensuite souligné que la combinaison de descripteurs est prometteuse d’un point de vue théorique mais compliqué à mettre en pratique. Enfin, nous avons évalué les performances du système en comparant les descripteurs et en mesurant la précision du moteur de recherche.