



Université Hassan 1^{er}

Faculté des sciences et techniques-Settat

MST IMSC



Département d'informatique et mathématique

RECHERCHE D'IMAGES BASÉE SUR LE CONTENU

Vision par ordinateur

Réalisé par :

HAYAR Hamza.

Responsable :

Mr. GADI Taoufiq

Année universitaire : 2020/2021

Table des matières

1.	Introduction	5
1.1	Recherche d'images basée sur le contenu	5
1.2	Contexte du CBIR	6
1.2.1	Extraction des caractéristiques de la couleur	6
1.2.2	Les descripteurs de textures	7
1.2.3	Les descripteurs de formes	7
1.3	Facteurs de Motivation.....	7
1.3.1	Facteurs sociétaux.....	7
1.3.2	Facteurs industriels	8
1.3.3	Médicale	8
1.4	Objective.....	8
1.4.1	Pré-traitement des données	9
1.4.2	Extraction de caractéristiques de couleur	9
1.4.3	Extraction de caractéristiques de texture	9
1.4.4	Extraction de caractéristiques de forme	10
1.4.5	Comparer la similarité des images et récupérer des images similaires.....	10
2	Développement.....	10
2.1	Outils.....	10
2.2	ARCHITECTURE DE BASE	11
2.3	Techniques.....	12
3	Implémentation	13
3.1	GÉNÉRALE	13
3.2	Histogramme de couleur	14
3.3	Filtre Gabor en tant qu'extraction de caractéristiques de texture	14
3.4	Extraction de caractéristique de forme avec « Canny ».....	16
3.5	Le descripteur d'histogramme de bord (EHD).....	17
3.6	Calcul de la distance euclidienne entre deux images.....	18
3.7	Calcul des valeurs de similarité	18
4	Test et discussion	20
4.1	General	20
4.2	Interface graphique	20
4.3	Test	21

4.3.1	Image requête de test	21
4.3.2	Test d'algorithme 1 (Couleur HSV)	22
4.3.3	Test d'algorithme 2 (Gabor moyenne)	22
4.3.4	Test d'algorithme 3 (Gabor standard)	22
4.3.5	Test d'algorithme 4 (Texture EHD)	22
4.3.6	Test d'algorithme 5 (Contour canny)	23
4.3.7	Résultat final	23
4.4	Limitations de ce système CBIR	23
4.4.1	Le modèle HSV	23
4.4.2	Limitations du filtre Gabor	24
5	Conclusion	24

Tableau de figures

Figure 1 Représentation de l'espace couleur HSV	7
Figure 2 Architecture de système CBIR	12
Figure 3 schéma d'indexation basé sur Histogramme	14
Figure 4 Les paramètres de Gabor	15
Figure 5 L'image d'entrée et la sortie finale de la fonction myGabor().	16
Figure 6 Descripteur d'histogramme de contour	17
Figure 7 Implémentation de fonction qui calcule la distance	18
Figure 8 Implémentation d'algorithme qui calcule la valeur de similarité	19
Figure 9 GUI	20
Figure 10 GUI - traitement sur l'image requête	20
Figure 11 Gui - Recherche à dix images similaires	21
Figure 12 Images de test	21

Liste de tableaux

Tableau 1 Test d'algorithme basé sur couleur HSV.....	22
Tableau 2 Test d'algorithme basé sur Gabor moyenne	22
Tableau 3 Test d'algorithme basé sur Gabor standard	22
Tableau 4 Test d'algorithme basé sur EHD	23
Tableau 5 Test d'algorithme basé sur détection de contour (canny)	23
Tableau 6 Résultat finale du test.....	23

1 Introduction

1.1 Recherche d'images basée sur le contenu

Avec le développement actuel de la technologie multimédia et des appareils numériques, les bases de données d'images numériques se développent rapidement. Ces bases de données fournissent une assistance vers de multiples domaines et secteurs, notamment dans la recherche médicale. L'immense quantité d'images stockées dans ces bases de données a créé des défis croissants pour les systèmes informatiques pour rechercher et récupérer des images pertinentes rapidement et avec précision. Traditionnellement, la récupération d'images basée sur du texte (TBIR) a été utilisée pour résoudre le problème de récupération d'images. Cependant, cela demande beaucoup de travail car une intervention humaine est nécessaire pour annoter manuellement chaque image avec des mots-clés. S'assurer que tous les détails de l'image ont été correctement indexés ou étiquetés avec les mots-clés pertinents est une corvée. En outre, l'utilisation de texte pour identifier les images n'est pas aussi précise ni fiable lorsqu'il s'agit de récupérer des images par sujet. Il existe également le cas où une image peut avoir été balisée dans une langue différente et ces mots-clés devraient être traduits dans la langue de la requête de texte pour la correspondance. Le manque de contexte dans les images ou les mots-clés que l'utilisateur utilise ou saisit est également un problème dans TBIR. Par exemple, si un utilisateur demandait le mot-clé « pomme » en attendant des images relatives à la société de technologie Apple, mais recevait des images de pommes réelles à moins qu'il ne tape plus de détails. Pour résoudre les principaux problèmes de TBIR dans le problème de la récupération d'images, la récupération d'images basée sur le contenu (CBIR) est introduite.

Rechercher d'images basée sur le contenu visuel est potentiellement une technique très puissante. Imaginez un système dans lequel l'utilisateur peut interroger le système pour récupérer toutes les images fixes et les trames vidéo contenant un certain type d'IED, après quoi le système répond en présentant exactement ces images. De même, imaginez le même système, mais où l'utilisateur choisit plutôt d'interroger le système en fournissant un exemple d'image d'un navire prise par une caméra haute résolution, après quoi le système fournit toutes les images de la base de données dans laquelle le navire est présent, ainsi que informations concernant ses emplacements précédemment documentés. Nous sommes convaincus que ces types de techniques peuvent présenter un grand intérêt pour les forces

armées suédoises. Certains des objectifs de ce projet sont donc d'avoir un aperçu de ce qu'est l'état de l'art du CBIR, quels types de systèmes sont disponibles sur le marché commercial ou en tant qu'outils open source déjà aujourd'hui, quelles sont les principales limitations actuelles de CBIR sont, et quel type de fonctionnalité qui peut être attendu dans un proche avenir (défini comme quelques années à partir du présent).

1.2 Contexte du CBIR

La récupération d'images basée sur le contenu est fortement basée sur l'extraction et la comparaison ultérieure des caractéristiques visuelles des images. Ces caractéristiques des images peuvent être classées en trois catégories :

- Fonctionnalités de bas niveau (liées à la vision humaine)
- Fonctionnalités de niveau intermédiaire (liées à l'objet)
- Fonctionnalités de haut niveau (liées à la sémantique)

Les caractéristiques de bas niveau sont des composants bruts de base de l'image qui se composent généralement de la couleur, de la texture, des bords, de l'orientation du dégradé, etc. Ces caractéristiques de bas niveau peuvent être traitées ultérieurement pour détecter des caractéristiques de niveau supérieur telles que des objets et des descriptions sémantiques peuvent même être générées.

1.2.1 Extraction des caractéristiques de la couleur

Les informations sur la couleur sont les caractéristiques physiques inférieures et intuitives. Parce que la couleur est robuste aux effets du bruit, de la taille et de l'orientation de l'image, la caractéristique de couleur est donc généralement utilisée dans la recherche d'images basée sur le contenu. La quantification de la couleur est étroitement liée à l'espace couleur. De nombreux types d'espaces couleur ont été proposés et utilisés pour la recherche d'images. Cependant, les différents espaces de couleur ont des applications différentes de décider quel type d'espace couleur est le plus approprié pour notre algorithme de recherche d'images. L'espace couleur HSV peut bien imiter la perception humaine des couleurs.

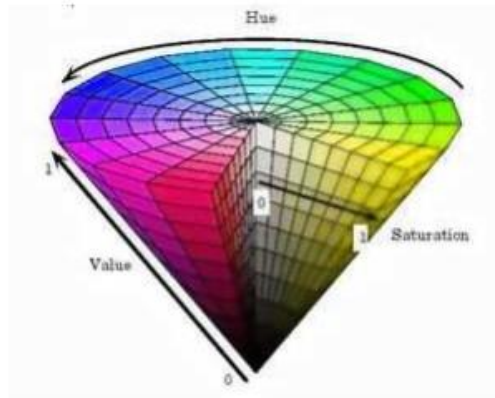


Figure 1 Représentation de l'espace couleur HSV

1.2.2 Les descripteurs de textures

La texture est une caractéristique visuelle importante qui décrit la propriété de surface d'un objet et sa relation avec l'environnement environnant. La texture d'une image est caractérisée par des différences de luminosité avec des fréquences élevées dans le spectre de l'image. Il existe deux catégories dans lesquelles les textures sont placées, les méthodes statistiques et les méthodes structurelles. Les méthodes structurelles sont couramment appliquées sur des textures régulières où d'autres méthodes statistiques, car leur nom fait référence à la vérification de la distribution statistique de l'intensité de l'image

1.2.3 Les descripteurs de formes

Les formes sont plus compliquées que les caractéristiques de couleur et de texture. Ils doivent passer par un processus d'identification de région ou de contour et être segmentés en objets et arrière-plan significatifs avant d'appliquer des descripteurs de forme. Cependant, il est plus difficile sur le plan informatique d'effectuer des comparaisons de forme, par conséquent, la contribution des descripteurs de forme en général est rare.

1.3 Facteurs de Motivation

1.3.1 Facteurs sociétaux

C'est un truisme d'observer que les images sont actuellement utilisées dans tous les domaines de la vie. L'influence de la télévision et des jeux vidéo dans la société d'aujourd'hui est évidente pour tous. La raison la plus courante pour stocker, transmettre et afficher des images est probablement à des fins récréatives, bien que cette catégorie comprenne une grande variété d'attitudes et de styles d'interaction différents, allant de regarder passivement le dernier épisode d'un feuilleton à l'analyse active des clichés d'une star du

tennis dans le l'espoir d'améliorer son propre jeu. Les images sont de plus en plus utilisées pour véhiculer des informations, dans des domaines aussi divers que la cartographie, la météo et les achats par correspondance, et pour persuader ou transmettre une humeur, comme dans la publicité. Ils peuvent aussi être appréciés à part entière, en tant qu'œuvres d'art.

1.3.2 Facteurs industriels

Dans le domaine de l'utilisation professionnelle de l'image, la situation est assez différente. Bien qu'il existe certainement des différences de style entre les ingénieurs de conception individuels, par exemple, la nature du processus de conception impose un certain nombre de contraintes incontournables dans lesquelles tous les ingénieurs doivent travailler. Il est donc possible de généraliser dans une certaine mesure la manière dont les images sont utilisées par les différentes professions. Étant donné que ce rapport concerne principalement le stockage et la récupération d'images, il est logique de limiter notre discussion en se concentrant sur les utilisations qui impliquent d'une manière ou d'une autre des collections d'images stockées.

1.3.3 Médicale

Les professions médicales et apparentées utilisent et stockent des informations visuelles sous forme de radiographies, d'échographies ou d'autres images numérisées, à des fins de diagnostic et de surveillance. Il existe des règles strictes sur la confidentialité de ces informations. Les images sont conservées avec les dossiers de santé des patients qui sont, pour la plupart, des fichiers manuels, stockés par identifiant unique. Les informations visuelles, à condition qu'elles soient rendues anonymes, peuvent être utilisées à des fins de recherche et d'enseignement. Une grande partie des efforts de recherche liés aux images est entreprise dans le domaine de la physique médicale. Les aspects préoccupants comprennent des systèmes de traitement d'image efficaces qui aident le praticien à détecter et à diagnostiquer les lésions et les tumeurs et à suivre la progression.

1.4 Objective

L'objectif de ce projet était de développer un système de récupération d'images basé sur le contenu, qui trie une grande base de données d'images en utilisant les caractéristiques des images plutôt que des métadonnées ajoutées précédemment et une

partie dans laquelle l'utilisateur peut améliorer la qualité de l'image requête avec multiples des outils de traitement d'image. Étant donné une image de requête, le système renvoie des images de la base de données similaires à la requête.

1.4.1 Pré-traitement des données

La récupération d'images basée sur le contenu implique le traitement d'une grande bibliothèque d'un ensemble de données disponible. Du fait qu'un grand référentiel d'images peut être disponible dans toutes sortes de tailles dimensionnelles et de formats de codage, des méthodes de prétraitement doivent d'abord être imposées à chacune de ces images. Cette étape est nécessaire pour appliquer une certaine norme de format pour chacune des images avant que les méthodes d'extraction de caractéristiques ne soient appliquées. Les méthodes de pré-traitement appliquées à l'image sont :

- Redimensionner les images à une résolution prédéterminée.
- Conversion d'images en niveaux de gris dans le même format d'espace colorimétrique (tel que RVB) avant un traitement ultérieur.

1.4.2 Extraction de caractéristiques de couleur

La caractéristique de couleur est la caractéristique la plus générale extraite d'une image car elle est robuste et invariante aux rotations et aux mises à l'échelle. Les objectifs de cette étape seront simples et directs.

- Implémentez un extracteur de caractéristiques de couleur basé sur les occurrences de chaque pixel de couleur des images.
- Avoir une sortie vectorielle de caractéristiques finales qui peut être utilisée comme un comparateur précis pour la similitude d'image.

1.4.3 Extraction de caractéristiques de texture

Les caractéristiques de texture d'une image sont considérées comme un élément essentiel pour que le système ait des sorties précises. Les objectifs de cette extraction sont les suivants :

- Exécutez les images dans l'extraction de caractéristiques de texture qui identifiera les textures

- Assurez-vous que l'extraction des caractéristiques de texture prend en compte les échelles et l'orientation.
- Avoir une sortie qui décrit les textures des images.

1.4.4 Extraction de caractéristiques de forme

La forme est une fonctionnalité importante et la plus puissante utilisée pour la classification, l'indexation et les récupérations d'images. Les objectifs de cette extraction sont les suivants :

- Exécutez les images dans l'extraction de caractéristiques de forme qui identifiera les formes.
- Pour des images d'entrée données, les caractéristiques de forme sont extraites à l'aide d'une méthode de détection de contour (canny).

1.4.5 Comparer la similarité des images et récupérer des images similaires

Un système de récupération d'images efficace ne serait pas considéré comme complet sans une méthode fiable pour comparer les indices de similarité entre l'image de requête et toutes les images de l'ensemble de données. Notre objectif est d'affiner les résultats de sortie finaux, en affichant uniquement les dix images les plus similaires. Dans ce projet, les objectifs que on a fixé pour cet aspect comprennent :

- Implémentez une méthode précise de calcul et de mesure de la similarité entre l'image de requête et toutes les images d'un ensemble de données, à l'aide des caractéristiques visuelles extraites.
- Incluez un processus qui fait la fusion de plusieurs algorithmes et choisi les dix images plus proches à l'image de la requête.

2 Développement

2.1 Outils

MATLAB est un environnement de développement de calcul numérique et de visualisation hautes performances. Il fournit un environnement interactif avec des centaines de fonctions intégrées pour le calcul technique, les graphiques et l'animation. Il offre également une extensibilité facile avec son propre langage de programmation de haut niveau. Le nom MATLAB signifie (MATrix Laboratory).

MATLAB est un programme efficace pour le traitement de données vectorielles et matricielles. Il contient des fonctions prêtes pour les manipulations matricielles et la visualisation d'images et permet à un programme d'avoir une structure modulaire. En raison de ces faits, MATLAB a été choisi comme logiciel de prototypage.

Avantages de MATLAB :

- MATLAB fournit de nombreuses fonctions pour le traitement d'images et d'autres tâches. La plupart de ces fonctions sont écrites dans le langage MATLAB et sont lisibles publiquement sous forme de fichiers texte brut.
- MATLAB permet d'assurer une précision numérique maximale dans le résultat final.
- MATLAB est un langage de programmation scientifique et fournit un support mathématique et numérique solide pour la mise en œuvre d'algorithmes avancés.
- Les algorithmes de traitement d'images disponibles sous MATLAB sont susceptibles d'être plus avancés que ceux disponibles à partir d'autres applications de traitement d'images.

2.2 ARCHITECTURE DE BASE

Le CBIR tire nombre de ses méthodes du domaine du traitement d'images et de la vision par ordinateur, et est considéré par certains comme un sous-ensemble de ce domaine. Il diffère de ces domaines principalement par l'accent mis sur la récupération d'images présentant les caractéristiques souhaitées à partir d'une collection de taille significative. Le traitement d'images couvre un domaine beaucoup plus large, comprenant l'amélioration, la compression, la transmission et l'interprétation d'images. Bien qu'il existe des zones grises (telles que la reconnaissance d'objets par analyse de caractéristiques), la distinction entre l'analyse d'image traditionnelle et le CBIR est généralement assez nette. Un exemple peut rendre cela clair. De tels systèmes peuvent être utilisés de deux manières. Premièrement, l'image devant la caméra peut être comparée à l'enregistrement de la base de données d'un seul individu pour vérifier son identité. Dans ce cas, seules deux images sont appariées, un processus que peu d'observateurs appelleraient CBIR. Deuxièmement, la base de données entière peut être consultée pour trouver les images les plus proches. Il s'agit d'un véritable exemple de CBIR.

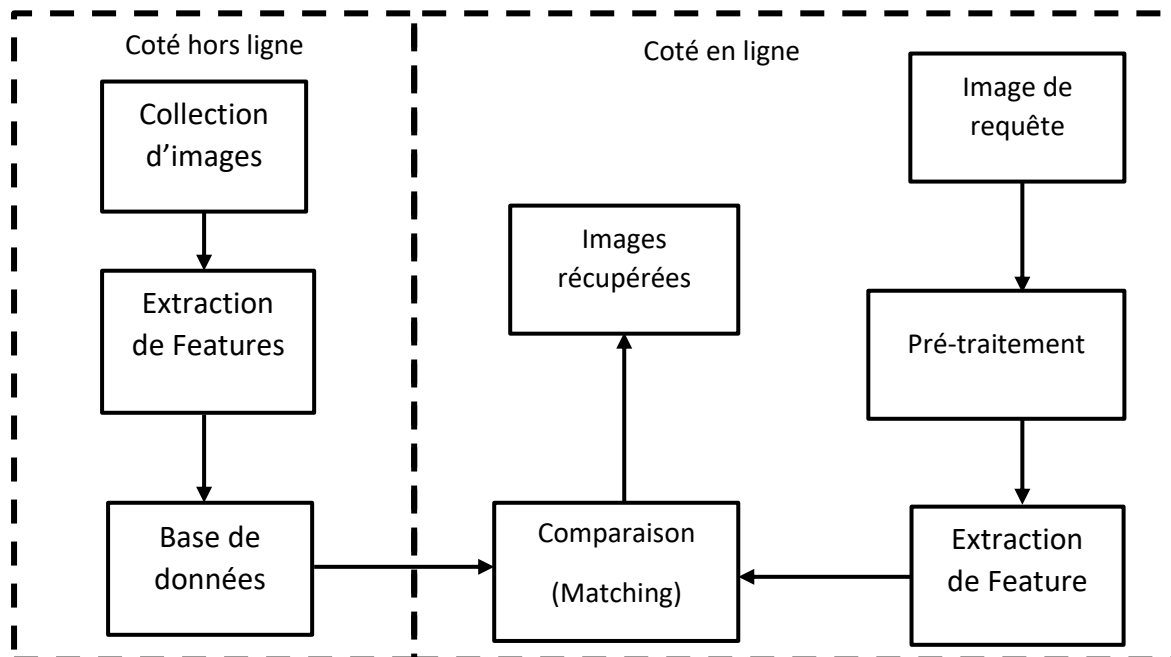


Figure 2 Architecture de système CBIR

r

la base de caractéristiques (telles que la couleur, la texture et la forme) qui peuvent être automatiquement extraites des images elles-mêmes. Les caractéristiques utilisées pour la récupération peuvent être primitives ou sémantiques, mais le processus d'extraction doit être principalement automatique.

Etapes principales :

- La première étape de la récupération d'images consiste à faire un traitement sur l'image, adapter la luminosité et le contraste, appliquer des filtres pour améliorer la qualité et pour bien visualiser les caractéristiques d'image.
- La deuxième étape c'est d'extraire les caractéristiques de l'image, il existe assez de méthodes et d'algorithmes que l'on peut utiliser dans cette étape.
- La troisième étape, représenter la base de données et l'image de la requête dans un format utile pour la comparaison. Lors de la recherche d'images similaires, toutes les informations sur les images ne sont pas nécessaires ou pertinentes.

2.3 Techniques

La précision de la recherche des images similaires, principalement basée sur l'extraction de caractéristiques d'images. Des caractéristiques d'image plus distinguées donneront de meilleurs résultats dans le processus d'indexation et de récupération. Dans ce projet, nous avons extrait la caractéristique de forme en utilisant les algorithmes canny et

gabor, pour la texture on utilise **Descripteur de texture homogène** (HTD) et on extrait les caractéristiques de couleurs à l'aide des composantes RGB ou HSI. Les représentations de forme peuvent être basées sur des bords ou des régions. Les caractéristiques fournissent des informations numériques sur une image, qui ne changent pas même lorsque la position, la taille et la direction des objets sont modifiées. On n'oublie pas la dernière étape de comparaison entre les images indexées dans la base de données et l'image requête, alors pour calculer la similarité entre les images on utilise la distance euclidienne, on met en ordre les images selon ses distances euclidiennes.

3 Implémentation

3.1 GÉNÉRALE

Dans cette partie, nous décrivons les détails de l'implémentation des algorithmes et les étapes de son fonctionnement pour le CBIR. Dans ce projet on va implémenter six algorithmes d'indexation différents pour la recherche d'image et on a la possibilité de fusionner le résultat de deux algorithmes ou plusieurs, ainsi qu'on a la possibilité d'améliorer la qualité de l'image à l'aide des fonctions de Matlab.

Alors que nous actualisons une implémentation CBIR pour ce projet, nous avons décidé du flux et de la conception d'implémentation suivants.

- **Avoir une base de données** : j'ai trouvé une base de données qui contient un grand ensemble de données d'images destinées à des fins de récupération d'images. L'ensemble de données comprend 500 images de test explorées sur le Web à basse résolution
- **Lecture d'une image de requête en tant qu'entrée de l'utilisateur** : j'ai récupéré l'une des images du grand ensemble de données et l'ai définie comme image de requête. Cela agit comme un test au début du système.
- **Extraction de caractéristiques** : pour l'extraction de caractéristiques de forme, on utilise les algorithmes canny et gabor, pour la texture on utilise **Descripteur de texture homogène** (HTD) et on extrait les caractéristiques de couleurs à l'aide des composantes RGB ou HSI.

- **Correspondance des caractéristiques** : une fois les caractéristiques extraites, les résultats similaires sont utilisés pour comparer, puis les valeurs de similitude sont calculées et comparées les unes aux autres
- **Vérification des résultats** : Une fois les étapes précédentes exécutées, les valeurs de similarité sont récupérées et les images similaires de la base de données sont triées en conséquence.

3.2 Histogramme de couleur

Un histogramme des couleurs décrit la répartition des couleurs dans l'ensemble d'une image ou d'une scène vidéo ou dans une région spécifiée. En tant que caractéristique au niveau des pixels, l'histogramme est invariant par rapport à la rotation, la translation et la mise à l'échelle d'un objet. Dans le même temps, l'histogramme ne capture pas d'informations sémantiques et deux images avec des histogrammes de couleurs similaires peuvent posséder des contenus totalement différents. Un espace colorimétrique quantifié HSI (ou RVB) est généralement utilisé pour représenter la couleur afin de rendre la recherche partiellement invariante aux contraintes non pertinentes telles que l'éclairage et les points de vue des objets.

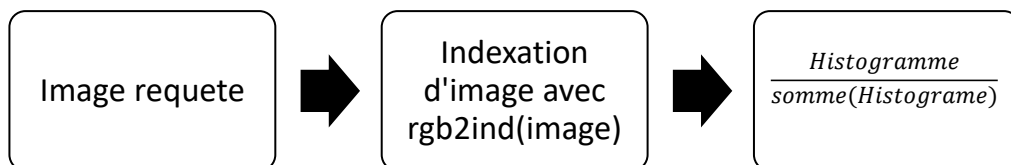


Figure 3 schéma d'indexation basé sur Histogramme

3.3 Filtre Gabor en tant qu'extraction de caractéristiques de texture

Les filtres de Gabor sont constitués d'ondelettes de Gabor qui est un groupe de fonctions exponentielles complexes modulées par un signal de Gauss. La raison de ce choix est que le filtre de Gabor prend en compte toutes les différentes échelles et orientations.

Une image est considérée comme étant formée en superposant une série d'ondes sinusoïdales de différentes fréquences orientées dans différentes directions. Chaque pixel de l'image indique à l'utilisateur l'intensité d'une telle onde. Par conséquent, le travail de Gabor

Filtre est d'isoler les informations spécifiques. Avec une telle méthode, Gabor Filtre améliore réellement les différentes textures de chaque pixel et fait ressortir la texture.

Le filtre Gabor doit prendre en compte quelques paramètres pour que chaque texture fonctionne. Il y a la fréquence/longueur d'onde d'accord, qui établit quel type de filtre sinus le filtre répondra le mieux. Pour que cela fonctionne, nous avons joué avec le nombre pour assurer que le bon type est choisi. Le paramètre suivant est l'angle central qui dans le code de l'équipe est écrit Pi. Ces ondes peuvent avoir n'importe quelle direction, il est donc important que les vagues à un angle spécifique soient bien choisies. Il y a aussi un paramètre de rapport d'aspect qui doit être pris en compte, il précise l'ellipticité du support de la fonction de Gabor. La bande passante vient après cela, c'est la bande passante de fréquence spatiale à demi-réponse d'un filtre de Gabor. Enfin, il y a un paramètre qui fait attention à l'orientation des vagues. Il spécifie l'orientation de la normale aux bandes parallèles d'une fonction de Gabor et les valeurs valides sont comprises entre 0 et 360.

Les nouvelles coordonnées du pixel lorsqu'on applique la fonction de Gabor est :

$$g(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + 2\pi W_x\right)$$

Dans notre implémentation du filtre Gabor, `myGabor()`, nous avons tenté d'ajuster les valeurs des paramètres du filtre Gabor afin d'obtenir une meilleure précision de sortie. Cependant, il est vrai qu'il n'est toujours pas parfait et ne donne pas nécessairement des résultats excellents ou précis tout le temps. Notre préréglage final pour les paramètres de Gabor est le suivant :

```
% preset gabor filter arguments
gamma = 1;           % (aspect ratio)
psi = 0.1;           % (phase)
theta = 90;          % (orientation)
bw = 2.8;             % (bandwidth)
lambda = 3.5;         % (wavelength)
pi = 180;             % (pi)
```

Figure 4 Les paramètres de Gabor

Avant d'appliquer le filtre de Gabor, quelques étapes de prétraitement sont appliquées sur l'image. Étant un filtre en niveaux de gris, bien sûr, la première méthode de prétraitement appliquée à l'image serait la méthode intégrée `rgb2gray()`. Après l'avoir convertie en niveaux de gris, l'image est ensuite redimensionnée à une résolution de 270 par 480 avec la méthode `imresize()`. Ensuite, `im2double()` est appliqué, ce qui double effectivement la précision de l'image d'entrée. Enfin, la fonction Gabor peut enfin être appliquée à l'image.

Exemple



Figure 5 L'image d'entrée et la sortie finale de la fonction `myGabor()`.

La sortie donnera une double matrice 240×480 . Les fonctions intégrées `mean()` et `std()` sont ensuite appliquées pour obtenir les vecteurs de caractéristiques de moyenne et d'écart-type, les deux ayant une taille de vecteur condensée de 1×480 . Ces vecteurs de caractéristiques seront ensuite utilisés pour calculer la similarité avec l'image de la requête à l'aide des calculs de distance euclidienne.

3.4 Extraction de caractéristique de forme avec « Canny »

La détection des contours est le nom d'un ensemble de méthodes mathématiques qui visent à identifier les points d'une image numérique auxquels la luminosité de l'image change brusquement ou de manière plus formelle, présente des discontinuités. Le point auquel la luminosité de l'image change brusquement est généralement organisé en un ensemble de segments de lignes courbes appelés bords. Le but de la détection des contours en général est de réduire considérablement la quantité de données dans une image, tout en préservant les propriétés structurelles à utiliser pour le traitement ultérieur de l'image.

Détection des contours de Canny, l'une des méthodes de détection des contours développées par John F Canny (JFC) en 1986.

Dans cet algorithme, le bord de définition optimal est détecté sur la base de certains critères qui incluent la recherche de bords en minimisant le taux d'erreur, en créant des bords proches des bords réels pour maximiser l'emplacement et en marquant les bords une seule fois lorsqu'un seul bord existe pour une réponse minimale. Le filtre optimal qui répond aux trois critères ci-dessus peut être approché en utilisant la dérivée première d'une fonction gaussienne :

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma^2} + \frac{y^2}{\sigma^2}\right)\right)$$

Dans la première étape, l'image est convoluée avec un filtre gaussien. Ensuite, le gradient de l'image est mesuré en alimentant l'image convoluée. L'opération de convolution 2-D est décrite par l'équation suivante

$$l'(x, y) = g(k, 1) * l(x, y)$$

$g(k, 1)$: Filtre

$l(x, y)$: Image original

3.5 Le descripteur d'histogramme de bord (EHD)

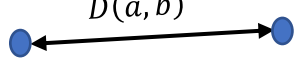
Le descripteur d'histogramme de bord (EHD) contient la distribution spatiale de cinq types de filtres, c'est-à-dire f_1, f_1, f_2, f_3, f_4 . f_1 Est pour l'horizontale, f_2 est pour la verticale, f_3 est pour 45° , f_4 est pour 135° , f_5 est pour non directionnel. Des filtres peuvent être appliqués pour récupérer des images ayant une signification sémantique similaire. La composition de l'EDH avec d'autres descripteurs tels que les performances du descripteur d'histogramme de couleur YCbCr peut être améliorée. L'EHD signifie la distribution des bords locaux dans l'image.



Figure 6 Descripteur d'histogramme de contour

3.6 Calcul de la distance euclidienne entre deux images

Toutes les valeurs qui ont été calculées sont stockées dans la ligne désignée au début. Par exemple, dans notre implémentation, la ligne 1 est pour toutes les valeurs de l'histogramme des couleurs, la ligne 2 est pour la moyenne du filtre de Gabor et la ligne 3 est l'écart type du filtre de Gabor, ligne 4 indexation de contour. Une fois que les valeurs des deux extractions de caractéristiques ont été calculées, les matrices vectorielles de caractéristiques de l'image de requête et de chaque image de l'ensemble de données sont transmises à la fonction de distance euclidienne. Dans la fonction, une formule simple est utilisée pour obtenir la distance.

$$D(a, b) = \sqrt{\sum_{i=1}^n (b_i - a_i)^2}$$


À partir de l'équation ci-dessus, la différence entre deux vecteurs de caractéristiques est calculée puis mise au carré. Une fois au carré, la sommation est récupérée et enfin la racine carrée de la valeur entière produira la distance précise entre les deux images. Plus la valeur de distance est élevée, plus les deux images doivent être dissemblables.

```
function result = euclideanDistance(feature1, feature2)

    % Euclidean distance for comparing the feature vector
    score_euclidean = sqrt(sum((feature1-feature2).^2));

    result = score_euclidean;
end
```

Figure 7 Implémentation de fonction qui calcule la distance

Les valeurs sont renvoyées dans la fonction principale. La matrice euclidienneDistances qui stocke les résultats des calculs de distance euclidienne est mise à jour en conséquence. Les autres étapes sont expliquées ci-dessous.

3.7 Calcul des valeurs de similarité

Après avoir calculé les distances euclidiennes, nous avons en fait déjà toutes les informations nécessaires pour trier et filtrer les images dissemblables. Cependant, ces valeurs de distance euclidienne n'ont pas de plage maximale définie. Pour visualiser correctement ce concept, rappelez-vous que la distance euclidienne de deux images

complètement similaires serait de 0, ce qui est logique car les deux images n'ont aucune différence, quelles que soient les méthodes d'extraction de caractéristiques utilisées pour obtenir le vecteur de caractéristiques. Les deux images seraient considérées comme 100% similaires.

Quand l'utilisateur sélectionne plus qu'un algorithme pour la recherche le système va fusionner les résultats des algorithmes de cette façon :

$$Df(a, b) = \sum_{i=1}^s \frac{D_i(a, b)}{S}$$

$Df(a, b)$: distance finale

$D_i(a, b)$: Distance d'algorithme i

S : nombre des algorithmes sélectionnées

```
maxDist = euclideanDistances(length(euclideanDistances),firstPassFilter);

len = length(euclideanDistances);
for ii = 1:len
    eDist = euclideanDistances(ii,firstPassFilter);
    result = 1 - (eDist/maxDist);
    similarityValues(ii,firstPassFilter) = result;
end
```

Figure 8 Implémentation d'algorithme qui calcule la valeur de similarité

Cependant, lorsque la distance euclidienne de deux images dissemblables est calculée, en fonction de la méthode d'extraction de caractéristiques, elles produisent une plage de valeurs sans limite supérieure apparente. Deux images complètement similaires donneraient toujours une valeur de 0, ce qui correspond à une similitude de 100 %. Mais le problème auquel on a été confrontés est le suivant : qu'est-ce qui constitue une similitude à 0 % ?

Dans le but d'une mise en œuvre simple, on a opté pour l'utilisation de la valeur de distance euclidienne maximale comme valeur de référence pour 0% de similarité. Cela signifie qu'une valeur de similitude de 0 % ne signifierait pas vraiment que les deux images sont complètement différentes, et toutes les valeurs de similitude seront calculées proportionnellement à l'image la plus dissemblable dans l'ensemble de données.

4 Test et discussion

4.1 General

Premièrement on va importer une image de test et faire un traitement pour améliorer sa qualité, puis on va tester la recherche avec tous les algorithmes et discuter les résultats.

4.2 Interface graphique

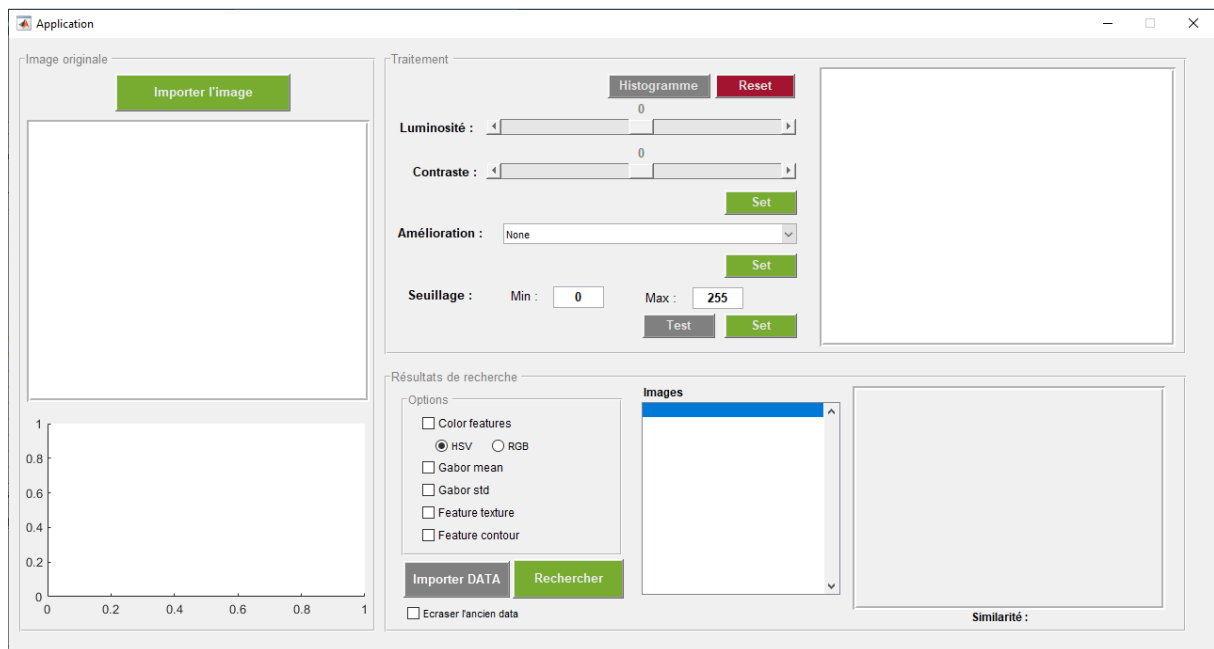


Figure 9 GUI

L'interface d'utilisateur principale a une conception simple et directe. En partant de la droite, l'utilisateur pourra télécharger une image de requête et les images de base de données pour récupérer des images similaires, ainsi qu'une section optionnelle pour améliorer la qualité d'image.

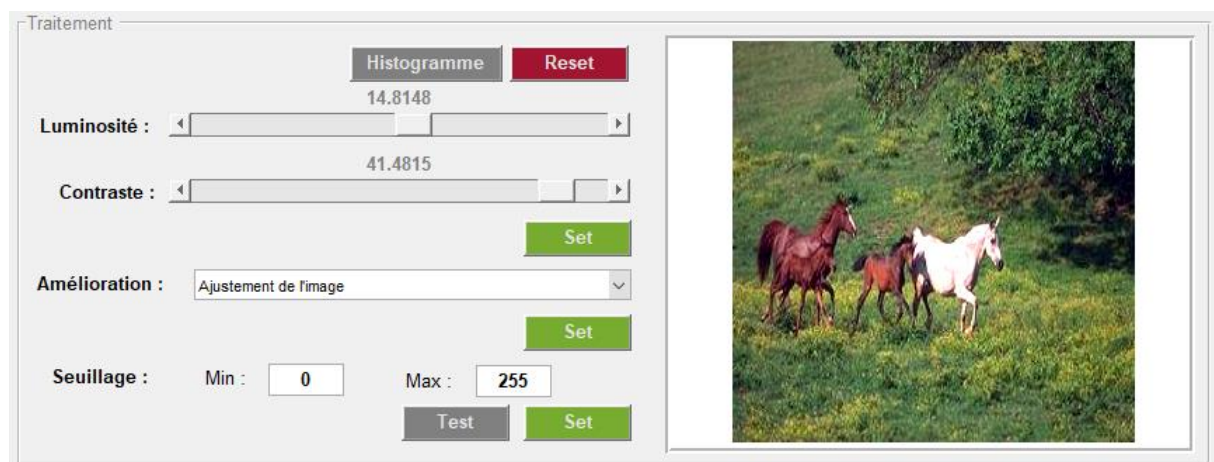


Figure 10 GUI - traitement sur l'image requête

L'utilisateur à l'aide des outils qui se trouvent dans section de traitement peut changer la valeur de luminosité et de contraste ainsi qu'il peut améliorer l'image par sélectionnés un des trois options de la menus déroulants : Ajustement d'image, Egalisation d'histogramme et histogramme adaptative.

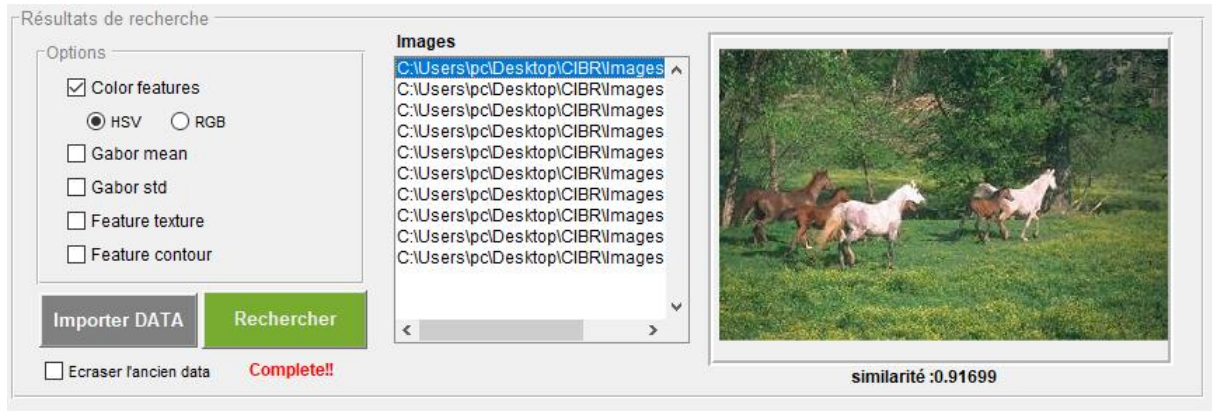


Figure 11 Gui - Recherche à dix images similaires

L'utilisateur pourra sélectionner les descripteurs de caractéristiques souhaités à utiliser pour filtrer les images pour. Quatre options : Descripteur de couleur, Gabor (moyenne), Gabor (std dev), descripteur de texture (EHD) et descripteur de contour (Canny). Dans la même zone le système va afficher le résultat de recherche dans le menu à coté et l'utilisateur peut visualiser les images similaires et la valeur de similarité par sélectionner un de ces images.

4.3 Test

4.3.1 Image requête de test

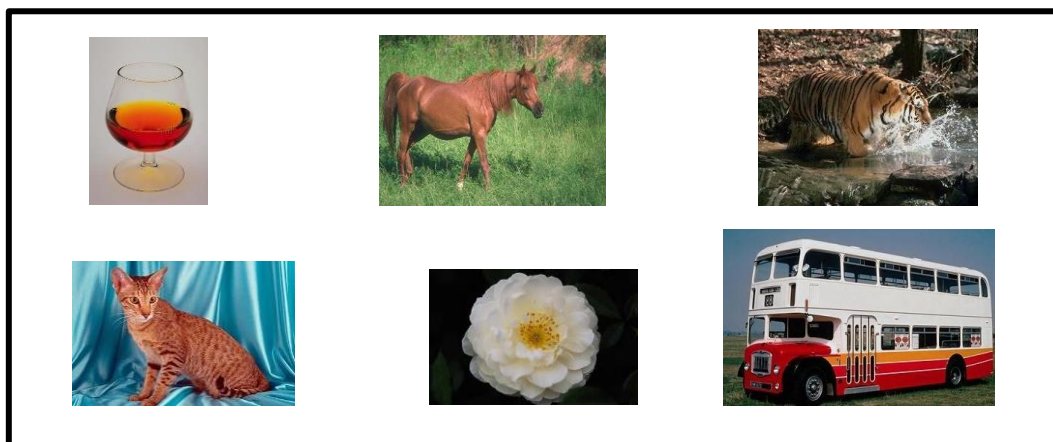


Figure 12 Images de test

On va utiliser ces deux images dans les tests suivant, on va tester sur tous les algorithmes et voir combien des résultats corrects on aura.

L'utilisateur peut choisir 2 algorithmes en même temps, pour ce test on va juste tester chaque algorithme individuellement.

4.3.2 Test d'algorithme 1 (Couleur HSV)

Image requête	Nombre des résultats correctes	Nombre des résultat faux	Erreur
Jus	10	0	0
Chevale	9	1	0.1
Tigre	10	0	0
Chat	9	1	0.1
Rose	8	2	0.2
Bus	10	0	0

Tableau 1 Test d'algorithme basé sur couleur HSV

4.3.3 Test d'algorithme 2 (Gabor moyenne)

Image requête	Nombre des résultats correctes	Nombre des résultat faux	Erreur
Jus	8	2	0.2
Chevale	5	5	0.5
Tigre	3	7	0.7
Chat	2	8	0.8
Rose	3	7	0.7
Bus	3	7	0.7

Tableau 2 Test d'algorithme basé sur Gabor moyenne

4.3.4 Test d'algorithme 3 (Gabor standard)

Image requête	Nombre des résultats correctes	Nombre des résultat faux	Erreur
Jus	9	1	0.1
Chevale	10	0	0
Tigre	5	5	0.5
Chat	3	7	0.7
Rose	3	7	0.7
Bus	2	8	0.8

Tableau 3 Test d'algorithme basé sur Gabor standard

4.3.5 Test d'algorithme 4 (Texture EHD)

Image requête	Nombre des résultats correctes	Nombre des résultat faux	Erreur
Jus	8	2	0.2
Chevale	10	0	0
Tigre	5	5	0.5
Chat	4	6	0.6
Rose	5	5	0.5
Bus	8	2	0.2

Tableau 4 Test d'algorithme basé sur EHD

4.3.6 Test d'algorithme 5 (Contour canny)

Image requête	Nombre des résultats correctes	Nombre des résultat faux	Erreur
Jus	6	4	0.4
Chevale	10	0	0
Tigre	6	4	0.4
Chat	5	5	0.5
Rose	10	0	0
Bus	10	0	0

Tableau 5 Test d'algorithme basé sur détection de contour (canny)

4.3.7 Résultat final

Algorithme	Erreur
Couleur HSV	0.08
Gabor moyenne	0.72
Gabor standard	0.56
Texture EHD	0.4
Contour canny	0.18

Tableau 6 Résultat finale du test

4.4 Limitations de ce système CBIR

4.4.1 Le modèle HSV

La raison pour laquelle le modèle HSV est une limitation du système est due aux avantages que le modèle couleur HSV a fournis à l'industrie du traitement d'images. Étant donné que HSV se concentre uniquement sur la teinte, la saturation et l'intensité, il ne récupère pas complètement la couleur de l'image. S'il y a respectivement deux couleurs dans chaque image qui ont une valeur de saturation ou d'intensité similaire, mais que le schéma de couleurs n'est pas similaire, il peut être placé dans la même plage l'un que l'autre. Cela fait que les valeurs de similarité sont élevées même si ce n'est pas le cas lorsque nous les regardons.

Mais malgré cela l'algorithme qui basé sur HSV restent efficace et avec d'erreur inférieur cela parce qu'on avait travaillé avec une base de données n'est très grande.

4.4.2 Limitations du filtre Gabor

L'une des principales raisons des résultats relativement imprécis est due au filtre Gabor. Le filtre de Gabor que j'ai implémenté est basé sur mon propre compréhension de base de la fonction, puisque je n'ai pas utilisé la fonction intégrée du filtre de Gabor. La mise en œuvre du filtre Gabor est simple et ne prend pas en compte beaucoup de détails mineurs qui peuvent avoir un impact sur la précision des images. L'autre raison est due au fait que la moyenne et l'écart type ont été calculés avec un algorithme simple, qui a prouvé que ce n'est pas le plus précis. La moyenne et l'écart type sont des entités séparées au lieu d'être ensemble. Par conséquent, lorsqu'il s'agit d'utiliser un filtre de Gabor, nous avons deux options qui ne sont pas similaires.

5 Conclusion

On a présenté un système de recherche et de récupération d'images dans ce projet. Concernant la taille de la base de données, on peut dire que notre système a fourni de bons résultats. En utilisant plus de mesures de performance, nous pouvons certainement affiner plus de fonctionnalités et éventuellement fournir aux utilisateurs les meilleures options de récupération en tant que paramètres par défaut, un peu comme le fait la recherche Google par image, les fonctionnalités, les options et les mesures de distance sont toutes cachées mais définies à l'écart. Les résultats de la recherche seront aussi pertinents que possible pour la requête.

De nombreuses améliorations pourraient être apportées à chaque extraction de caractéristiques, et malgré les nombreux efforts, il existe encore de nombreuses extractions de caractéristiques qui n'ont pas atteint leur plein potentiel. Par conséquent, il est très important que les chercheurs soient pleinement informés et gardent une trace de tout nouvel article qui a fonctionné sur CBIR