

Rapport de projet

**Indexation et recherche d'images par le
contenu en se basant sur
«Edge Histogram Distance »**

Réalisé par :

ABDELJAOUED Chiheb

BENCHAAABEN Achref

Encadré par :

Mr. Tebourbi Riadh

Année universitaire : 2019/2020

Table des matières

Introduction.....	1
I. Recherche d'images par le contenu :	2
1. Historique	2
2. Architecture du système d'indexation et recherche d'images	3
3. Aperçu sur les méthodes d'extraction de caractéristiques visuelles	4
a. Description globale de l'image	4
b. Caractéristiques globales vs locales.....	6
II. Solution proposée	7
1. Edge histogram descriptor.....	7
2. Kmeans	8
3. Architecture de la solution proposée	8
4. Résultats	9
Conclusion et perspectives.....	10

Introduction

Le domaine de l'image numérique est un domaine en pleine expansion. Depuis quelques années, avec l'explosion d'Internet et aussi le développement à grande échelle de la photographie numérique, il n'est pas rare d'avoir des bases d'images numériques massives, que ce soit des bases ciblées pour un domaine d'activité professionnelle (journalisme, tourisme, éducation, musées, ...) ou tout simplement pour les particuliers qui accumulent d'immenses bases de photographies numériques (souvenirs, voyages, famille, événements, ...).

En fait, l'automatisation de la recherche d'images devient un enjeu crucial pour un bon nombre d'applications et qui met en évidence l'importance de l'indexation de l'image à partir de l'analyse de son contenu visuel et qui permet de réduire les ambiguïtés ou lacunes des techniques maintenant classiques d'indexation par le texte. Elle devient même la seule solution lorsque le volume d'images rend l'annotation textuelle irréalisable.

La recherche par contenu visuel dans les bases de données massives d'images est une problématique émergente née de l'intersection de deux domaines de recherche en informatique : l'analyse d'images et les bases de données.

D'où l'idée de ce projet qui consiste à développer un système pour indexer l'image et de retrouver des images similaires lors d'une requête en fonction du contenu des images. En développant ce système, nous avons aussi développé une interface qui permet à l'utilisateur de faire la requête et d'afficher les résultats.

I. Recherche d'images par le contenu :

1. Historique

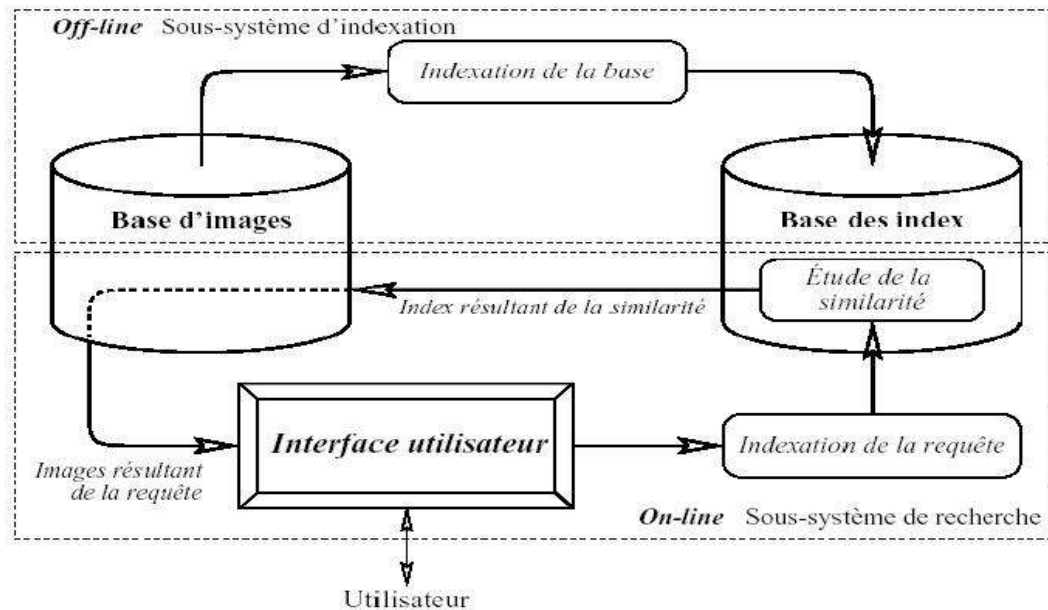
Dans les applications pratiques des bases d'images numériques que l'on retrouve aujourd'hui, que ce soit pour les chaînes de télévision, les journaux, les musées et mêmes pour les moteurs de recherche sur Internet qui proposent des solutions de recherche d'images, l'indexation et la recherche de ces images se font en se basant majoritairement sur **des annotations textuelles** ou **des éléments de texte** qu'on peut rattacher aux images.

Dans plusieurs cas (chaînes de télévision, journaux,), l'archivage des images et des séquences vidéo ne se fait qu'au prix d'une étape d'annotation manuelle à l'aide de mots-clés. Cette indexation représente une tâche longue et répétitive pour l'humain, surtout avec les bases d'images qui deviennent aujourd'hui de plus en plus grandes.

Le besoin de méthodes d'indexation et de recherche directement basée sur le contenu de l'image n'est donc plus à démontrer. Le premier prototype de système a été proposé en 1970 et ce système a attiré l'attention de beaucoup de chercheurs. Quelques systèmes deviennent des systèmes commerciaux tels que QBIC (Query By Image Content), CIREs (Content Based Image REtrieval System), ... Les systèmes d'indexation et recherche d'images par le contenu permettent de rechercher les images d'une base d'images en fonction de leurs caractéristiques visuelles. Ces caractéristiques, sont : la couleur, la texture, la forme et tout autre caractéristique de l'image qu'on peut imaginer.

2. Architecture du système d'indexation et recherche d'images

La figure suivante présente l'architecture du système de recherche d'images par le contenu.



Ce système s'exécute en deux étapes : l'étape d'indexation et l'étape de recherche :

- ❖ **Étape d'indexation** : des caractéristiques sont automatiquement extraites à partir de l'image et stockées dans un vecteur numérique appelé descripteur visuel. Grâce aux techniques de la base de données, on peut stocker ces caractéristiques et les récupérer rapidement et efficacement.
- ❖ **Étape de recherche** : le système prend une requête l'utilisateur et lui donne le résultat correspond à une liste d'images ordonnées en fonction de la similarité entre leur descripteur visuel et celui de l'image requête en utilisant une mesure de distance.

3. Aperçu sur les méthodes d'extraction de caractéristiques visuelles

Les méthodes de description de l'image dépendent des applications considérées. Il existe deux classes de descripteurs d'images : les descripteurs globaux et les descripteurs locaux.

a. Description globale de l'image

En indexation d'images, l'apparence visuelle globale d'une image est résumée par trois attributs de bas niveau : la couleur, la texture et la forme. Ces attributs ont été intégrés dans la norme **MPEG-7**.



La couleur

La couleur est l'un des éléments les plus représentatif du contenu d'une image, il est très souvent le premier descripteur qui est employé pour la recherche d'images. Il existe de nombreuses possibilités d'attributs pour caractériser la couleur tels que l'histogramme, les moments.

L'histogramme

Les histogrammes sont faciles et rapides à calculer, et robustes à la rotation et à la translation. Cependant l'utilisation d'histogrammes pour l'indexation et la recherche d'images pose des problèmes :



Ils ne possèdent pas d'informations spatiales sur les positions des couleurs



Ils sont sensibles à de petits changements de luminosité, ce qui est problématique pour comparer des images similaires, mais acquises dans des conditions différentes.



Ils sont inutilisables pour la comparaison partielle des images (objet particulier dans une image), puisque calculés globalement sur toute l'image)

Plusieurs travaux récents ont été effectués pour améliorer ces problèmes tel que l'utilisation d'autres espaces de codage des couleurs qui se rapprochent plus de la perception d'humaine :

L'espace RVB est un espace de couleur utilisé couramment, par facilité, dans tous les systèmes de vision automatique, mais il n'est pas forcément le mieux adapté. En effet, les trois composantes RVB sont très dépendantes les unes des autres. Un simple changement d'éclairage dans la scène modifie les trois composantes, alors que les objets de la scène n'ont pas changé de couleur, mais se sont simplement assombris ou éclairés.

L'espace TSV (Teinte Saturation Valeur - en anglais HSV) est le plus utile pour la segmentation et la reconnaissance et il a été prouvé un espace très fort dans le système de recherche des images. Parce qu'il s'agit d'une représentation plus physique de la couleur. Dans cet espace, on peut séparer pour un pixel : l'intensité du pixel (valeur) et la couleur du pixel (teinte et saturation).



Les moments statistiques

La méthode d'histogramme utilise la distribution complète de la couleur. On doit stocker de nombreuses données. Au lieu de calculer la distribution complète, dans les systèmes de recherche d'images, on calcule seulement des dominantes caractéristiques de couleur tels que l'espérance, la variance et d'autres moments. En fait, les recherches prouvent que les méthodes utilisées des moments statistiques marchent plus vite et donnent des résultats meilleurs que les méthodes d'histogrammes.

L'espérance, la variance, les moments d'ordre 3 peuvent également être calculés sur chaque composante couleur par les formules suivantes :

$$E_i = \frac{1}{N} \sum_{j=1}^N p_{ij} \quad \delta_i = \left(\frac{1}{N} \sum_{j=1}^N (p_{ij} - E_i)^2 \right)^{1/2} \quad s_i = \left(\frac{1}{N} \sum_{j=1}^N (p_{ij} - E_i)^3 \right)^{1/3}$$

Où E_i est l'espérance, δ_i est la variance, S_i est le troisième moment qui correspond à une composante couleur i (i est R, V, B de l'espace RVB ou T, S, V de l'espace TSV).



La texture

La texture est la répétition d'éléments de base construits à partir de pixels qui respectent un certain ordre. Le sable, l'eau, l'herbe, la peau sont autant d'exemples de textures. L'aléatoire joue un rôle particulier dans les textures. On peut distinguer deux types extrêmes de textures, entre lesquels se positionnent toutes les textures :

Les textures régulières, dans lesquelles la périodicité du motif est évidente : grilles, murs, tissus... Elles peuvent être décrites par des approches fréquentielles (spectres de

Fourrier, ondelettes de Gabor) ou des approches structurelles dans lesquelles on associe un motif et des règles de placement sur un pavage régulier.

Les textures aléatoires, pour lesquelles la distribution des intensités n'est l'objet d'aucune régularité apparente : sable, nuage, herbe... Elles peuvent être décrites par des lois statistiques sur les distributions.

Au niveau sémantique, les textures peuvent apporter une information précieuse. En effet, elles permettent de différencier des parties d'images dont les descripteurs de couleurs sont identiques. Par exemple, le ciel (texture unie ou nuageuse) et la mer (texture des vagues).



La forme

La description de la forme ou de la structure de l'image requiert en général deux niveaux de traitement. Elle nécessite en premier lieu une segmentation de l'image en régions ou au moins une extraction des contours. Ensuite, outre l'utilisation de caractéristiques simples comme la longueur ou la courbure des formes extraites, une technique simple consiste à caractériser la distribution de l'orientation des gradients sur les contours, sous la forme d'un histogramme par exemple.

b. Caractéristiques globales vs locales

A la base, les caractéristiques présentées précédemment sont calculées de façon globale sur l'image. Cependant, un système basé uniquement sur des caractéristiques globales ne peut pas donner les résultats désirés.

Soit une image composée de plusieurs objets ayant des caractéristiques, couleurs et textures, très différentes, le vecteur de caractéristiques globales extrait à partir de l'image entière perd les informations locales (les objets) et ne produit qu'une moyenne grossière du contenu de cette image. Par contre, l'analyse uniquement basée sur des caractéristiques locales risque de perdre le sens global de l'image, en submergeant celui-ci dans un flot de petits détails inutiles.

Par conséquent, un compromis doit être trouvé, différent selon les applications et selon les requêtes individuelles, entre caractéristiques globales et caractéristiques locales.

II. Solution proposée

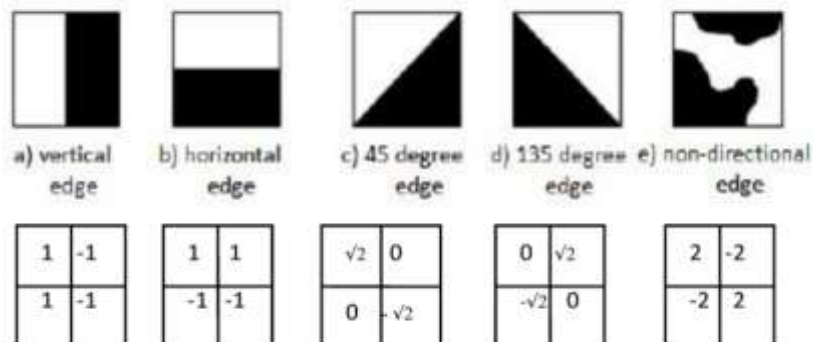
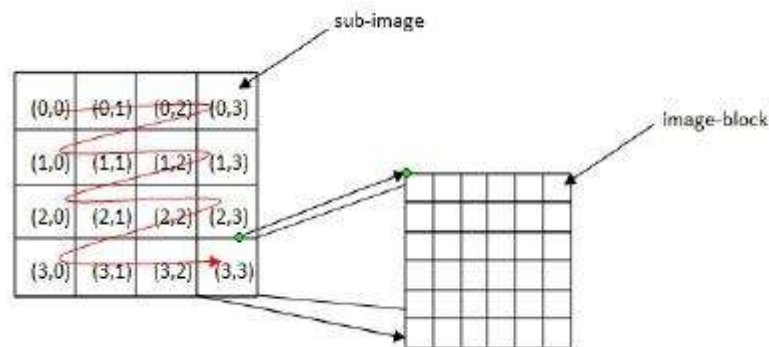
Dans ce projet, Pour la partie indexation, nous avons choisi comme descripteur "Edge histogram", et "MongoDB" pour le stockage des données et la recherche.

Pour la partie Recherche, on a utilisé l'algorithme de classification non-supervisé "Kmeans".

1. Edge histogram descriptor

C'est l'un des méthodes les plus utilisées pour la détection de forme. Il représente fondamentalement la fréquence relative d'occurrence de 5 types d'arêtes dans chaque zone locale appelée sous-image ou bloc d'image. La sous-image est défini en partitionnant l'espace image en 4x4 blocs qui ne se chevauchent pas, comme le montre la figure 1. Ainsi, la partition de l'image crée définitivement 16 blocs de taille égale Indépendamment de la taille de l'image d'origine.

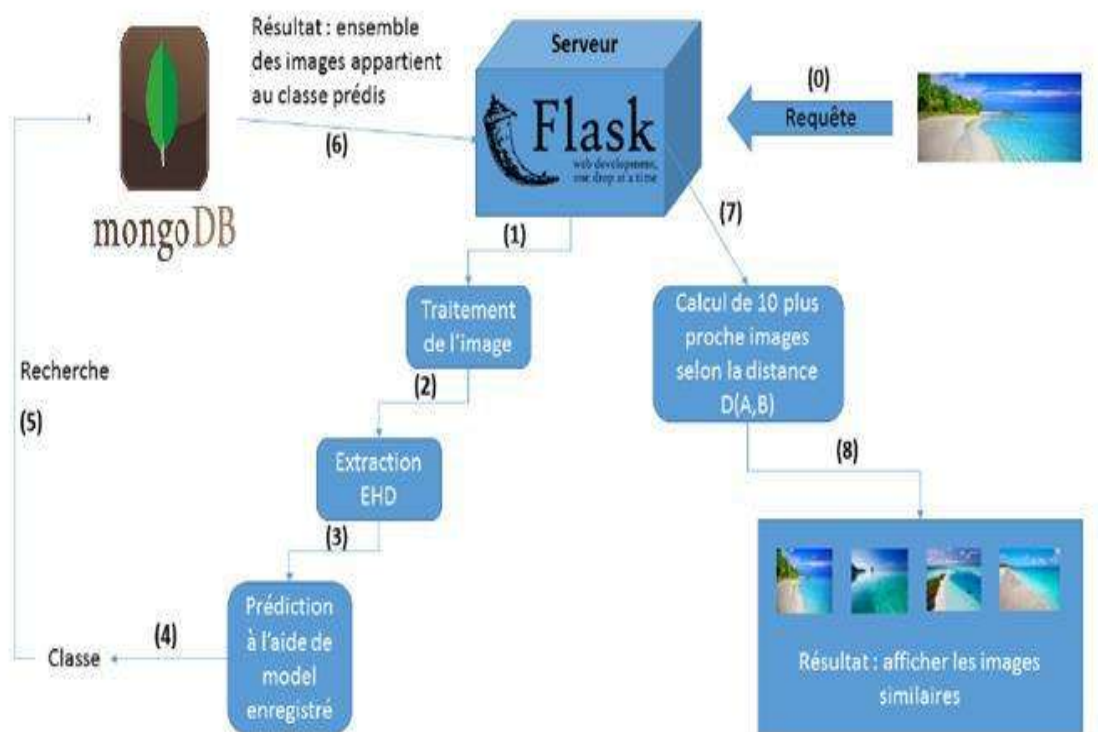
Pour définir la caractéristique du bloc d'image, nous générons ensuite un histogramme de la distribution des bords pour chaque bloc d'image. Les bords du bloc d'image sont classés en 5 types : verticale, horizontale, diagonale de 45 degrés, 135 degrés Bords diagonaux et non directionnels, comme illustré à la figure 2. Ainsi, l'historgramme de chaque bloc d'image représente la répartition relative des 5 types de bords dans la Sous-image correspondante.



2. Kmeans

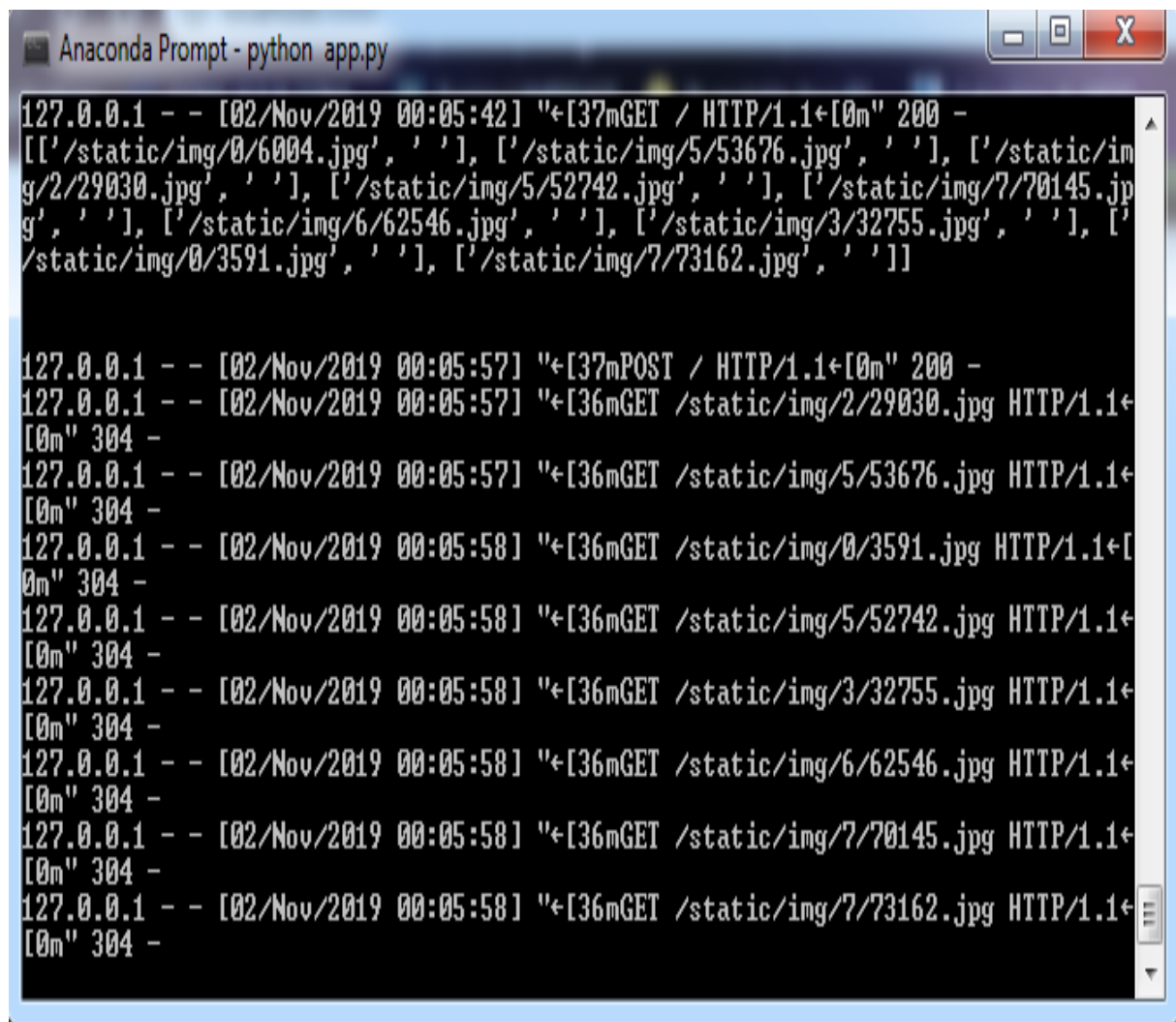
Étant donnés des points et un entier k , le problème est de diviser les points en k groupes, souvent appelés clusters, de façon à minimiser une certaine fonction. Cette fonction est la distance d'un point à la moyenne des points de son cluster.

3. Architecture de la solution proposée



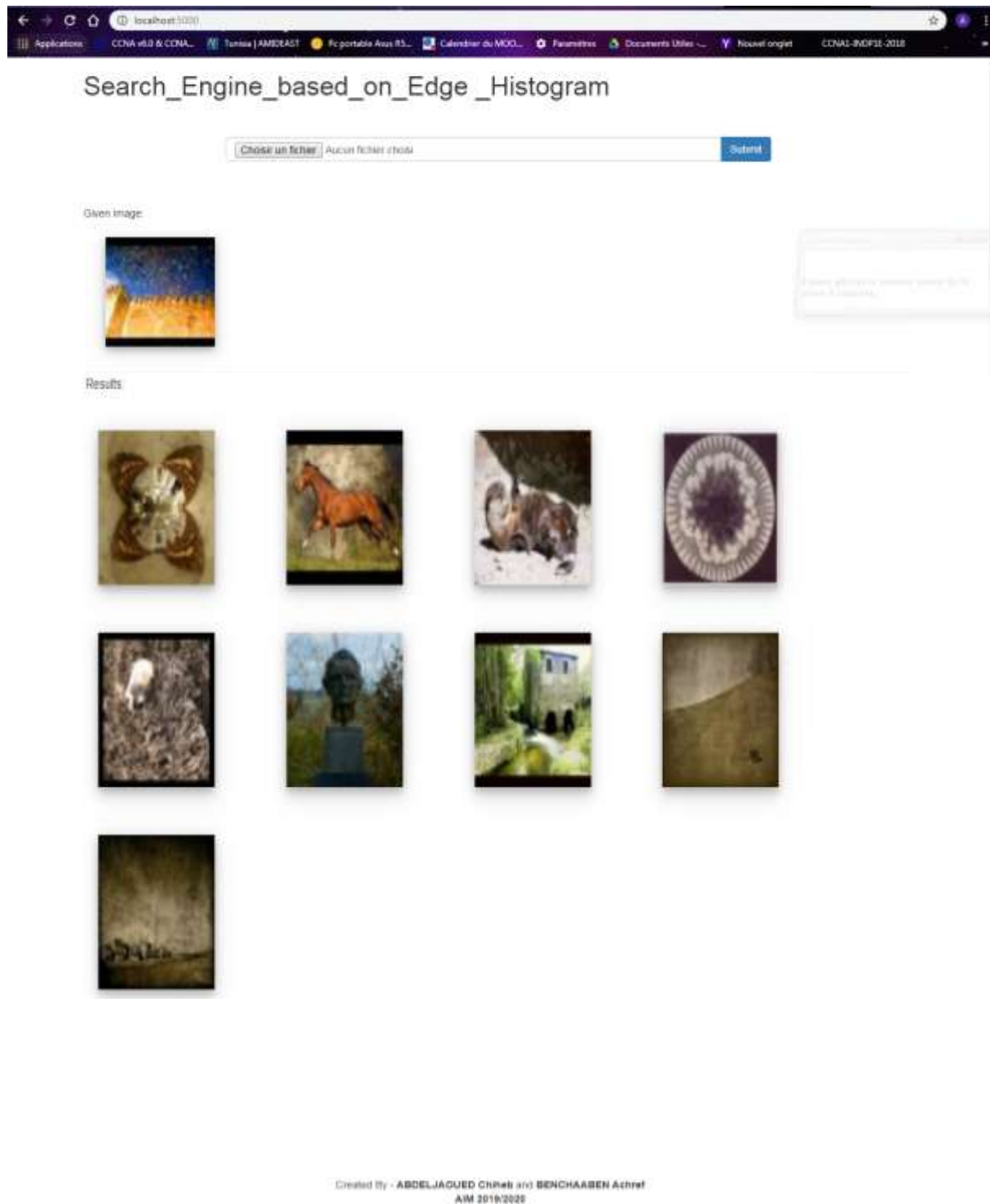
4. Résultats

Nous avons utilisé une base de données qui contient 80000 images (vu la limite de l'environnement matériel).

A screenshot of an Anaconda Prompt terminal window titled "Anaconda Prompt - python app.py". The terminal displays a series of HTTP requests and responses. The first line shows a 37m GET request to the root path, returning a 200 status. Subsequent lines show 36m GET requests to various image paths, each returning a 304 status. The paths include /static/img/0/6004.jpg, /static/img/5/53676.jpg, /static/img/2/29030.jpg, /static/img/5/52742.jpg, /static/img/7/70145.jpg, /static/img/6/62546.jpg, /static/img/3/32755.jpg, /static/img/0/3591.jpg, and /static/img/7/73162.jpg. The timestamps for all requests are [02/Nov/2019 00:05:42] or [02/Nov/2019 00:05:57] or [02/Nov/2019 00:05:58].

```
Anaconda Prompt - python app.py
127.0.0.1 - - [02/Nov/2019 00:05:42] "37mGET / HTTP/1.1" 200 -
[ '/static/img/0/6004.jpg', ' ' ], [ '/static/img/5/53676.jpg', ' ' ], [ '/static/im
g/2/29030.jpg', ' ' ], [ '/static/img/5/52742.jpg', ' ' ], [ '/static/img/7/70145.jp
g', ' ' ], [ '/static/img/6/62546.jpg', ' ' ], [ '/static/img/3/32755.jpg', ' ' ], [
'/static/img/0/3591.jpg', ' ' ], [ '/static/img/7/73162.jpg', ' ' ]

127.0.0.1 - - [02/Nov/2019 00:05:57] "37mPOST / HTTP/1.1" 200 -
127.0.0.1 - - [02/Nov/2019 00:05:57] "36mGET /static/img/2/29030.jpg HTTP/1.1"
304 -
127.0.0.1 - - [02/Nov/2019 00:05:57] "36mGET /static/img/5/53676.jpg HTTP/1.1"
304 -
127.0.0.1 - - [02/Nov/2019 00:05:58] "36mGET /static/img/0/3591.jpg HTTP/1.1"
304 -
127.0.0.1 - - [02/Nov/2019 00:05:58] "36mGET /static/img/5/52742.jpg HTTP/1.1"
304 -
127.0.0.1 - - [02/Nov/2019 00:05:58] "36mGET /static/img/3/32755.jpg HTTP/1.1"
304 -
127.0.0.1 - - [02/Nov/2019 00:05:58] "36mGET /static/img/6/62546.jpg HTTP/1.1"
304 -
127.0.0.1 - - [02/Nov/2019 00:05:58] "36mGET /static/img/7/70145.jpg HTTP/1.1"
304 -
127.0.0.1 - - [02/Nov/2019 00:05:58] "36mGET /static/img/7/73162.jpg HTTP/1.1"
304 -
```



Conclusion et perspectives

Nous avons développé un système d'indexation et de recherche d'images par le contenu. Ce système fonctionne avec le descripteur Edge histogram et utilise une base de données de 80000 images.

Nous avons aussi développé une interface qui permet à l'utilisateur de faire facilement une requête et de prendre les résultats.

Mais, nous voulons améliorer notre système et utiliser la totalité de la base d'image fournie ainsi de tester d'autres descripteur (descripteur locale de l'image).