

**La Recherche d'Images Similaires**  
**Projet intégrateur**

**Partie #1: Orientée-objet (Java)**



uOttawa

CSI2520[A] Paradigmes de programmation

Présenté pour Robert Laganière

Yassine Moumine

300140139

**Date:** le 17 Février 2024

## **Table de Matiere**

Introduction.....	2
Problème et.....	2
Solution Algorithmique.....	2
Données d'Entrée.....	2
Résultat Attendu.....	2
UML.....	3

## **Table de Figures**

Figure 1: Représentation tabulaire des pixels.....	3
Figure 2: Résultats de sortie.....	3
Figure 3: Diagramme UML.....	3

## Introduction

Dans le monde moderne, la prolifération rapide des images nécessite des outils informatiques avancés pour analyser, rechercher, classer et découvrir des images d'intérêt. Ce projet vise à implémenter une méthode de recherche d'images similaires en utilisant la programmation orientée objet en Java.

## Données d'Entrée

La base de données d'images est fournie au format JPG, avec les histogrammes précalculés enregistrés dans des fichiers texte. En outre, 16 images de requête sont fournies au format JPG et PPM. Pour cette partie nous utiliserons le format PPM.

## Problème

Le défi consiste à comparer des images en termes de couleurs, en se basant sur les histogrammes. Les images seront représentées en tant que grilles  $[x][y]$  de pixels, chaque pixel ayant 3 canaux pour les valeurs RGB.

```
R = pixel[x][y][0];  
G = pixel[x][y][1];  
B = pixel[x][y][2];
```

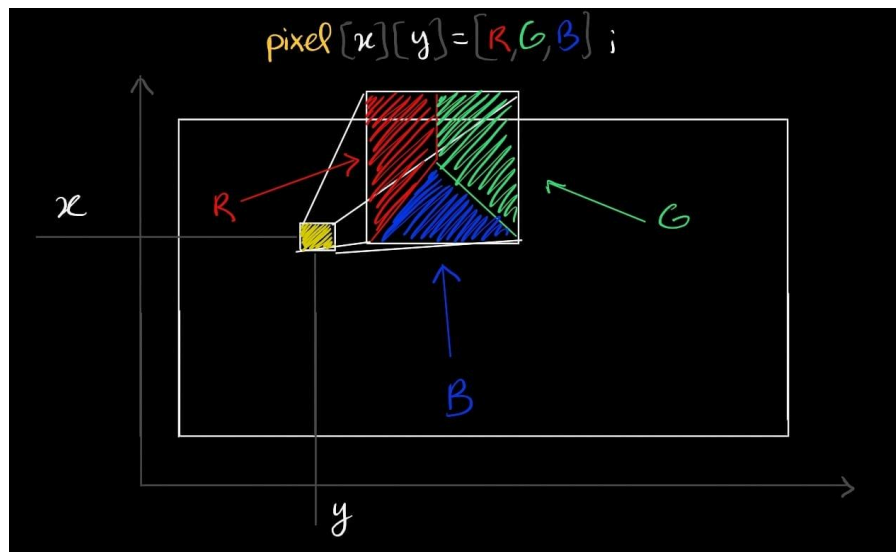


Figure 1: Représentation tabulaire des pixels

## Solution Algorithmique

L'algorithme propose une méthode de recherche d'images similaires en calculant et comparant les histogrammes couleur. Pour simplifier le décompte des couleurs d'une image, l'histogramme est réduit en limitant l'espace de couleurs. En passant de 8 bits par canal à 3 bits, ceci ramène l'espace à 512 couleurs, facilitant le calcul.

$$R' = R \gg (8-D)$$

$$G' = G \gg (8-D)$$

$$B' = B \gg (8-D)$$

Ensuite, pour normaliser l'histogramme et permettre la comparaison entre images de résolutions différentes, chaque entrée est divisée par le nombre total de pixels, assurant des valeurs comprises entre 0.0 et 1.0. Puis une simple opération d'intersection sera utilisée pour comparer deux histogrammes.

$$d(H_1, H_2) = \sum_I \min(H_1(I), H_2(I))$$

## Résultat Attendu

L'objectif est de trouver les 5 images les plus similaires à chaque image de requête, en utilisant l'intersection de leurs histogrammes comme vu précédemment.

```
Here's the top 5 most similar images to image q13.ppm:  
+ Image: 3225.jpg, with 100.000000% percent similarity  
+ Image: 2083.jpg, with 49.487269% percent similarity  
+ Image: 2082.jpg, with 49.453704% percent similarity  
+ Image: 2078.jpg, with 48.898148% percent similarity  
+ Image: 2607.jpg, with 47.237847% percent similarity
```

Figure 2: Résultats de sortie

# UML

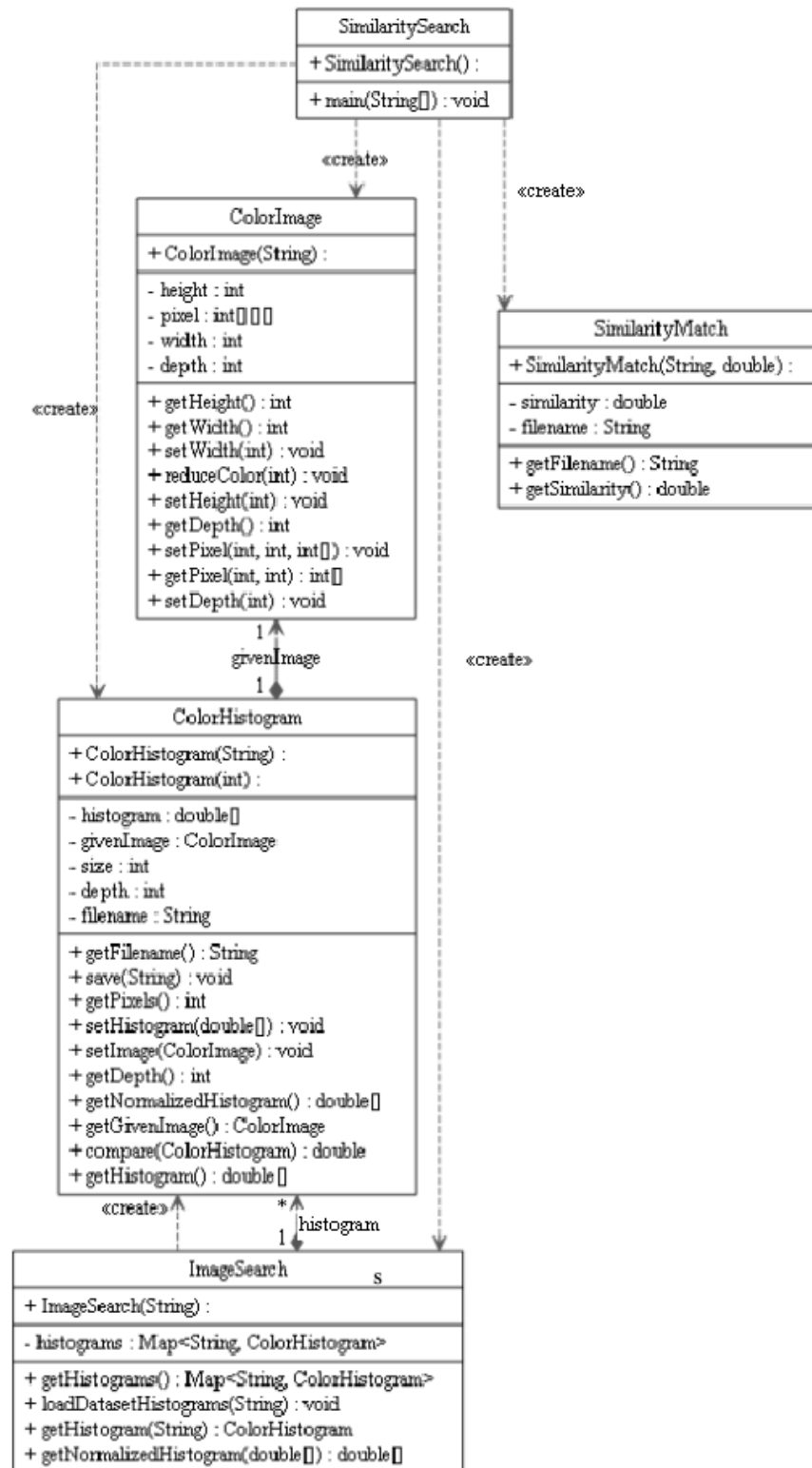


Figure 3: Diagramme UML