Traitement d'image sous Android
Projet

DIALLO Mamadou Saliou, FAYE Abdou Samath

Fevrier 2020

Table des matières

0.1 Introduction

Ce document contient une description du Projet de traitement d'image sous Android à rendre dans le cadre de l'UE Projet Technologique en 3ème année de licence Informatique à l'université de Bordeaux Campus Science Talence.
Le projet est effectué par un groupe de 4 étudiants.

0.2 Description du Projet

Le but de ce projet est de développer une application de traitement d'image sur smart phone avec système Android. Les images peuvent aussi bien être obtenues depuis la galerie du téléphone que directement depuis la caméra.

Ce logiciel est composé des fonctionnalités qui permettent de gérer, afficher, traiter et sauvegarder des images.

0.3 Description du code

Le projet est découpé en plusieurs classes (5) :
La classe principale MainActivity, Gray, Colorize, Contrast, et Convolution.

0.3.1 Rôle de chaque classe

1) **La classe principale MainActivity** :
La classe MainActivity est un composant crucial d'une application Android, elle permet de charger (ou démarrer) l'application.

Elle contient les méthodes :

createOnClickButton() qui permet la gestion de clique sur les boutons.

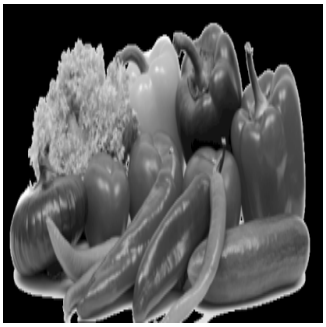
takePicture() qui permet de capturer une image depuis la ou les caméra(s) du téléphone.

changeSizeBitmap() qui permet de ré-dimensionner une image selon une certaine proportion.

onActivityResult() qui est une méthode overridé (redéfinie) gère le chargement de l'image depuis la galerie du téléphone et de la prise de photo par l'utilisateur et affiche de l'image.

onCreateOptionsMenu() et *onOptionsItemSelected*() pour la création du menu de l'application et l'appel des différentes méthodes de traitement d'image implémentées.

2) **La classe Gray :**
toGray(Bitmap bmp) , *toGrays*(), *toGrayRS*
Ces methodes permet de mettre une image couleur en gris.

AVANT :  APRÈS : 

3) **La classe Colorize** :

colorized()

Méthode applique une teinte choisie aléatoirement au Bitmap.

AVANT :  APRÈS : 

colorize()

Méthode applique une teinte choisie aléatoirement au Bitmap.

AVANT :  APRÈS : 

— *cannedColor*()
Méthode qui Conserve la couleur rouge

AVANT :  APRÈS : 

4) **La classe Contrast** :

a : ***Diminution du contraste d'une image couleur :***

- void *downContrasteColor*(Bitmap) :

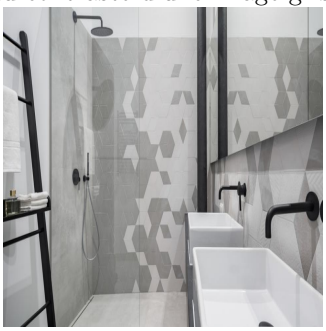
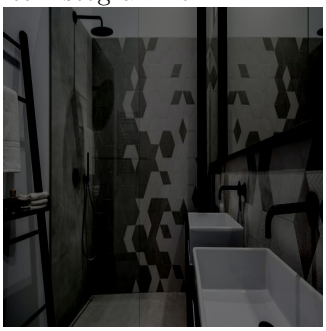
la methode permet la diminution du contraste d'une image couleur par égalisation d'histogramme

AVANT :  APRÈS : 

b : **Diminution du contraste d'une image gris :**

— void *decreasesContrastLUT*(Bitmap im) :

Diminution du contraste d'une image grise par égalité histogramme

AVANT :  APRÈS : 

c : **Augmentation du contraste d'une image gris :**

-

— void ***increasesContrast***(Bitmap)
augmentation du contraste d'une image grise par égalité d'histogramme

d : **Augmentation du contraste d'une image couleur :**

— void **increasesContrastLUT**(Bitmap im)
augmentation du contraste d'une image couleur par extension dynamique

— public static void ***upContrasteColor***(Bitmap im)

augmentation du contraste d'une image couleur par extension dynamique

AVANT :  APRÈS : 

(5) La classe Convolution :

a : **Le filtre Gaussien** :

— void *convolutionGaussN*(Bitmap ,int) :
Le filtre Gaussien donne un meilleur lissage et une meilleure réduction du bruit que le filtre moyenne.

AVANT :  APRÈS : 

b : **convolutionSobel(Bitmap bmp)** :

AVANT :  APRÈS : 

c : **Convolution Moyenneur :**

— void *convolutionMoy*(Bitmap bmp ,int n)

AVANT :  APRÈS : 

0.4 Présentation des fonctionnalités disponibles

1. Charger une image : L'application permet d'obtenir une image de plusieurs manières :

(a) depuis la galerie l'application permet de sélectionner une image parmi celles présentes dans la galerie du téléphone.

(b) depuis la caméra l'application permet de capturer une image depuis la ou les caméra(s) du téléphone.

2. L'application permet d'afficher sur l'écran l'image chargée

3. L'application permet de zoomer et dézoomer une image affichée sur l'écran, en utilisant l'interaction avec deux doigts.

4. Lorsqu'une image est affichée et déborde de l'écran, l'application permet de Scroller la zone affichée à l'aide d'une interaction avec un doigt.

5. L'application permet d'appliquer quelques filtrages usuels sur les images chargées.

(a) Régler la luminosité d'une image.

(b) Régler le contraste d'une image.

(c) L'application permet d'égaliser l'histogramme de l'image affichée.

(d) L'application permet de mettre en oeuvre les traitements couleur :la modification de la teinte ainsi que la sélection d'une teinte (rouge) à conserver lors du passage en niveaux de gris.

(e) L'application permet d'appliquer différents filtres basés sur une convolution : les filtres moyenneur, Gaussien, Sobel et Laplacien .

6. Elle permet de réinitialiser l'image après avoir appliqués un filtre à l'image.

7. Elle permet de sauvegarder une image modifiée dans la galerie du téléphone.

0.5 Les bugs et problèmes connus

L'application effectue une rotation de la photo prise par la caméra par le mode plein écran. La fonction *rotateBitmap*() qui permet d'éviter cela fait planter le logiciel.

0.6 Choix de fonctionnalités supplémentaires

Pas encore fait

0.7 Conclusion

Ce projet nous permet de decouvrir un domaine nouveau.

0.8 Bibliographie

https ://openclassrooms.com/fr/courses/4517166-developpez-votre-premiere-application-android

https ://openclassrooms.com/fr/courses/2023346-creez-des-applications-pour-android

https ://developer.android.com/training/camera/photobasics.htmlTaskPath