

Rapport Projet Technologique *Traitement d'image sous Android* *ANTRI Sofian*

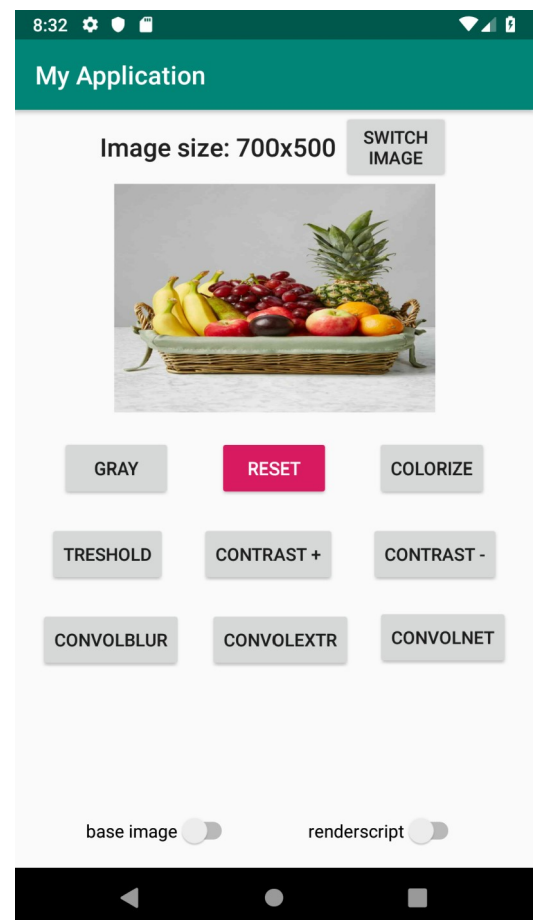
Introduction

Ce projet consiste à développer une application mobile permettant d'appliquer des effets sur une image. Cette application sera développée sous AndroidStudio.

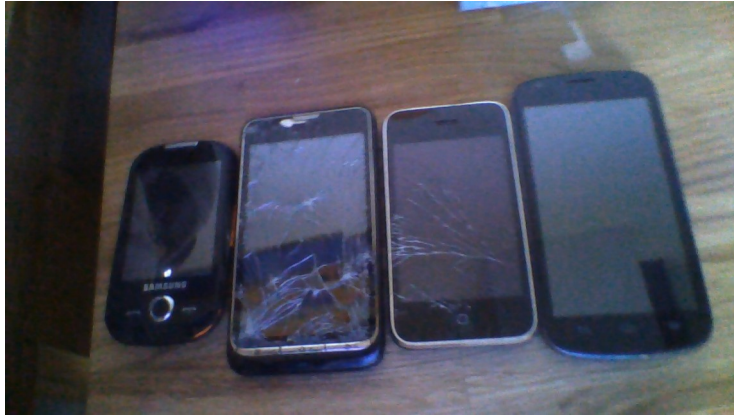
L'application ne possède qu'une seule fenêtre, celle-ci :

On choisira l'API 28 afin d'avoir accès au maximum de fonctionnalités.

Le téléphone choisi est le Nexus 5X car ses proportions sont plus plaisantes que celles des autres.



Images de base :



Algorithmes implémentés

L'application possède 10 boutons, et 2 interrupteurs.
Chaque bouton sert à appliquer un traitement sur l'image.

Le bouton SWITCH IMAGE sert, comme son nom l'indique, à passer sur une autre image, parmi les 3 images.

Si l'interrupteur « base image » est activé, les traitements se feront sur l'image de base, stockée dans la bitmap « old » réinitialisée à chaque fois. Autrement, le traitement sera effectué sur l'image actuelle.

Si l'interrupteur « renderscript » est activé, le traitement utilisera la version renderscript de son algorithme (implémentés seulement pour GRAY, RESET, et COLORIZE, fonctionnel uniquement pour RESET).

Le bouton GRAY appelle la fonction toGrayTab, qui effectue un simple calcul sur chaque pixel afin de le griser.

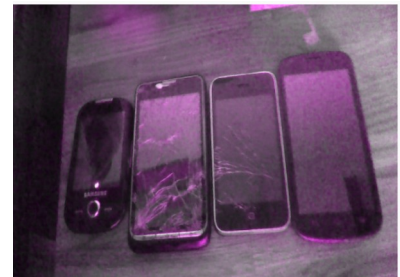
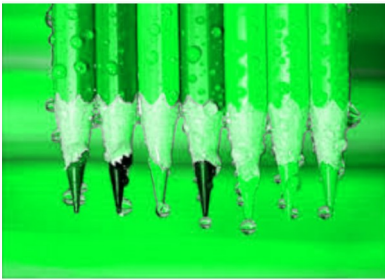


La version renderscript ne fonctionne plus, j'ignore pourquoi, j'ai pourtant repris le code du cours et elle ne fonctionne toujours pas, voici le résultat produit.



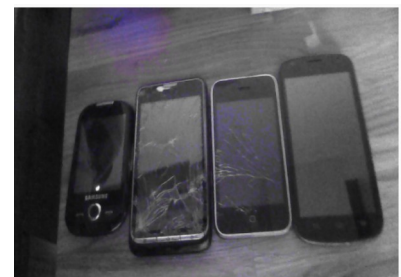
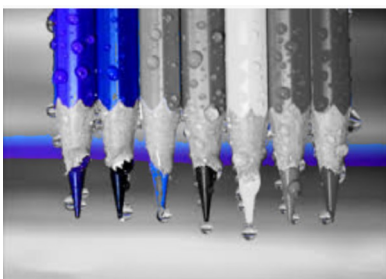
Le bouton RESET copie simplement l'image de base depuis old vers la bitmap affichée à l'écran, en version normale comme en renderscript.

Le bouton COLORIZE appelle la fonction colorize, qui, pour chaque pixel, choisit une teinte au hasard effectue le calcul de la saturation de l'image et de sa valeur, afin d'utiliser ces 3 valeurs pour les convertir en une nouvelle couleur en utilisant HSVToColor. J'ai choisi d'effectuer ces calculs au lieu de simplement convertir la couleur en HSV pour changer la teinte, car cela permettait un grand gain de temps.



Il y avait une version renderscript, qui effectuait la même chose, si ce c'est que la conversion de HSV en RGB s'effectuait manuellement, mais lors de l'implémentation j'ai rencontré quelques problèmes : elle effectuait la même chose que le renderscript du bouton GRAY, et ce même si le script était vide.

Le bouton TRESHOLD effectue un seuil, il choisit au hasard une marge de teinte, et met en gris tous les pixels n'appartenant pas à cette marge.

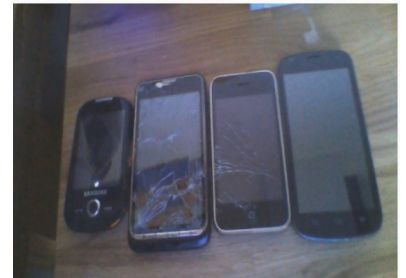
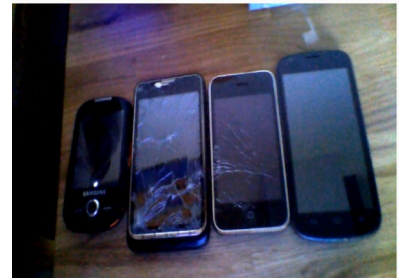
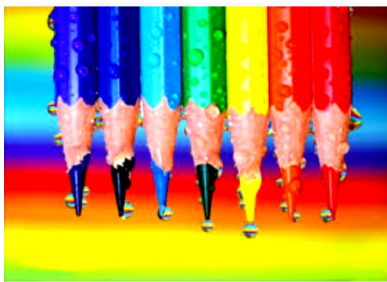


On remarque que la 3eme image étant relativement sombre, il est difficile d'en faire ressortir des couleurs avec cet algorithme.

Les boutons CONTRAST + et CONTRAST – jouent avec les champs R,G, et B de chaque pixel, afin de les éloigner ou les rapprocher de 128 ($=255/2$).

Je n'ai pas utilisée l'extension dynamique, car elle n'était que peu efficace, voire inefficace sur l'image « crayon », étant donné que son minimum et son maximum sont déjà de 0 et 255.

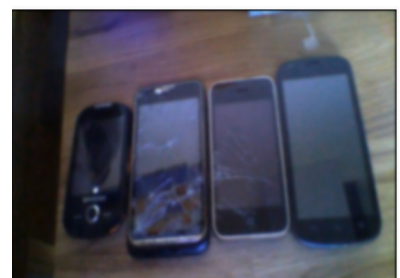
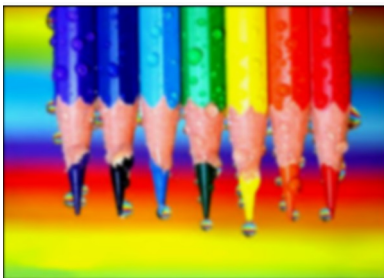
Résultats respectifs après 3 applications de chaque fonction :



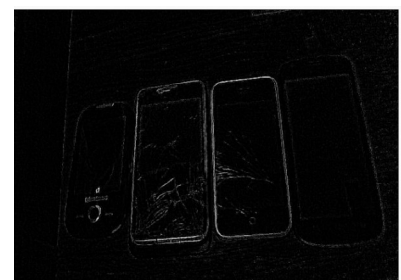
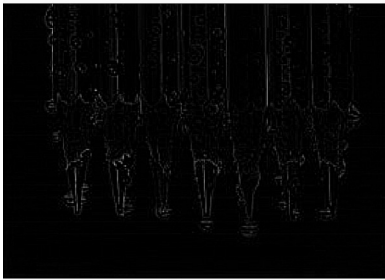
Les 3 derniers boutons utilisent la convolution, en appliquant chacun des masques prédéfinis à l'avance, de taille 3x3.

Le premier est un masque moyenneur, qui effectue un flou sur l'image, cependant, le masque étant petit, il faut effectuer plusieurs fois l'opération afin d'obtenir un véritable résultat.

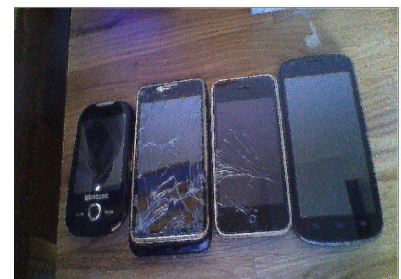
Résultats après 10 applications :



Le second est un masque de détection de bords, mettant toute l'image en noir, sauf ces derniers.



Le dernier est un masque de netteté, intensifiant les bords de l'image.



On remarque que les détails tels que les gouttes d'eau sur les crayons ou les fissures sur les téléphones ressortent beaucoup plus.

Conclusion

Ce projet fut particulièrement difficile à cause de l'environnement AndroidStudio, jusqu'ici inconnu.

De plus, ce logiciel ne fonctionne pas sur certaines machines du CREMI, ce qui a été relativement handicapant.

J'ai rencontré beaucoup de problèmes avec Renderscript, que je ne saurais pas expliquer et pour lesquels je n'ai pas trouvé de solutions dans le cours, ni sur le net.

Les feuilles de TD et les cours étaient moins fournies que dans les autres matières, nous étions moins accompagnés, cela était une bonne chose dans certains moments, mais dans d'autres, comme par exemple l'implémentation des scripts, cela rendait les choses assez difficile.