

Mosaïque d'Images avec Critères Avancés – Application Mobile

GONG NI(IASD) WANG XIHAO(IMAGINE)

April 2025

1 Introduction

Listing 1: La structure principale

```
1 com.example.motophosaique
2     MainActivity.java
3     HomeFragment.java
4     SelectFragment.java
5     ResultFragment.java
6     Utils.java
7     AlgoPagerAdapter.java
8     GreyAlgoFragment.java
9     ColorAlgoFragment.java
10    ObjectAlgoFragment.java
11    res/layout
12        fragment_home.xml
13        fragment_select.xml
14        fragment_result.xml
15        fragment_algos_grey.xml
16        fragment_algos_color.xml
17        fragment_algos_object.xml
```

1.1 MainActivity

- Point d'entrée de l'appli, qui héberge le NavHost
 - Lance le HomeFragment et gère la navigation entre les écrans

1.2 HomeFragment

- Demande et vérifie les permissions CAMERA / stockage
 - Utilise un ActivityResultLauncher pour récupérer la photo prise ou l'image choisie dans la galerie
 - Envoie l'URI de l'image à SelectFragment via SafeArgs/Bundle

1.3 SelectFragment

- Affiche un aperçu de l'image sélectionnée en décodant le PGM via Utils.decodePGM
- Propose un SeekBar pour ajuster la taille des blocs de mosaïque - Met en place un TabLayout + ViewPager2 pour passer entre les modes Grey, Color et Object
- Expose la méthode setAlgo(String) pour recevoir le choix d'algorithme des sous-fragments - Au clic sur Generate, regroupe blockSize, colorMode, algo et withRep dans un Bundle et navigue vers ResultFragment

1.4 AlgoPagerAdapter

- Adaptateur de ViewPager2 qui instancie GreyAlgoFragment, ColorAlgoFragment ou ObjectAlgoFragment selon la position d'onglet - Retourne le bon fragment pour chaque page et indique qu'il y a trois pages au total

1.5 GreyAlgoFragment / ColorAlgoFragment / ObjectAlgoFragment

- Chacun définit son propre layout avec trois boutons correspondant aux algorithmes disponibles - Lorsqu'on clique sur un bouton, appelle ((SelectFragment) getParentFragment()).setAlgo(...) avec la chaîne d'identification de l'algorithme

1.6 ResultFragment

- Récupère blockSize, colorMode, algo et withRep depuis les arguments du Bundle - En lance de fond l'appel JNI via MainActivity.generateMosaic(inputPath, outputPath, blockSize, algo, withRep) - Décode le fichier PGM de sortie en Bitmap et l'affiche dans un ImageView - Affiche le temps de traitement écoulé

1.7 Utils

- Méthodes pour copier un fichier ou un répertoire depuis assets vers le cache de l'appli - decodePGM(File) : lit un fichier P5 PGM et renvoie un Bitmap en niveaux de gris - (éventuellement) autres helpers pour encoder/décoder PPM ou sauver un Bitmap en PGM/PPM

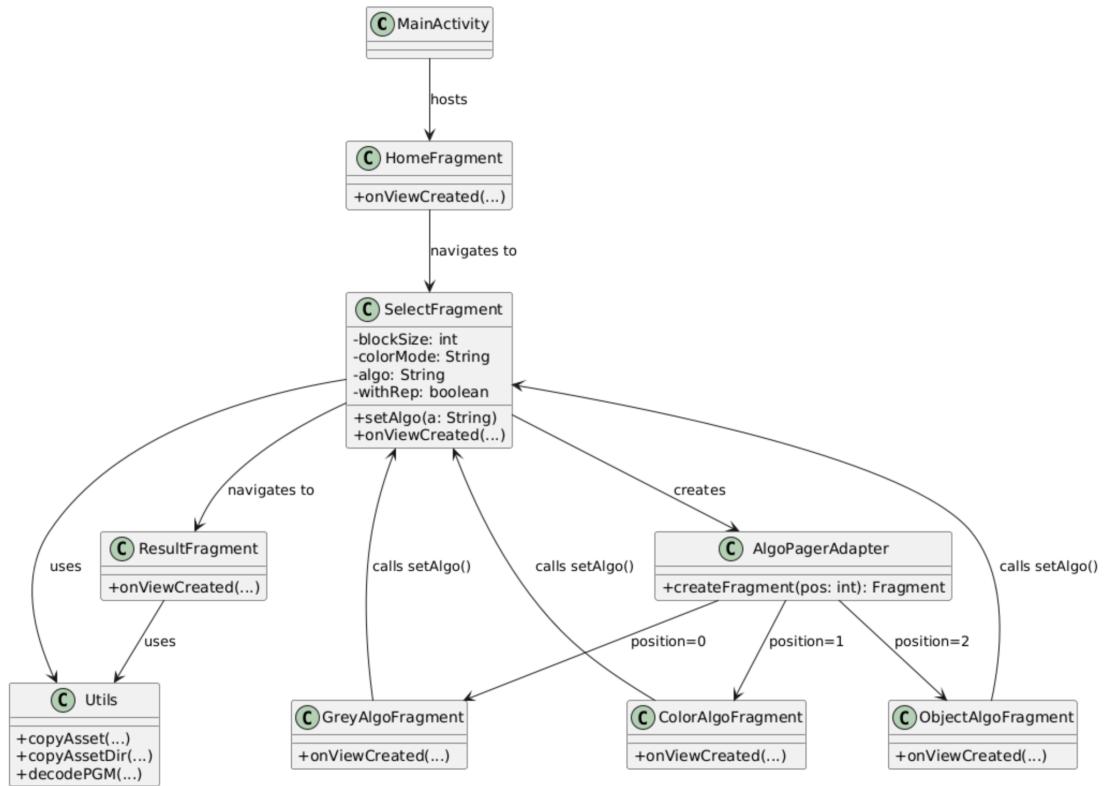


Figure 1: L'architecture

2 Captures d'écran

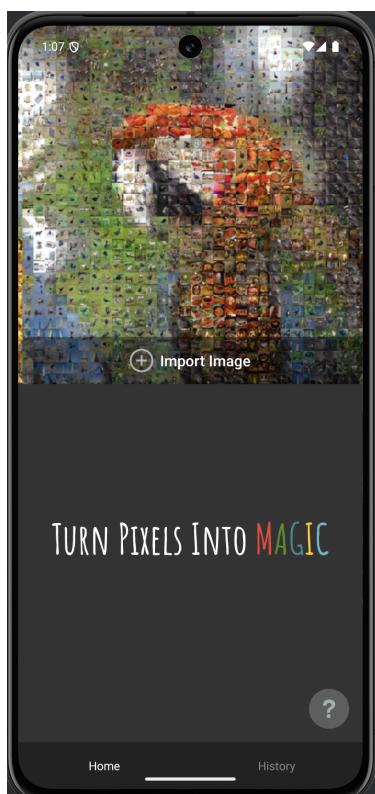


Figure 2: Home page

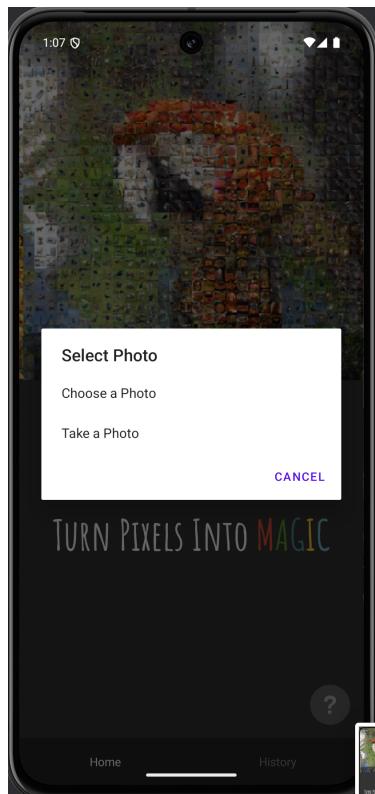


Figure 3: select photo by choose or take



Figure 4: example of taking a photo



Figure 5: select the size of block and the function

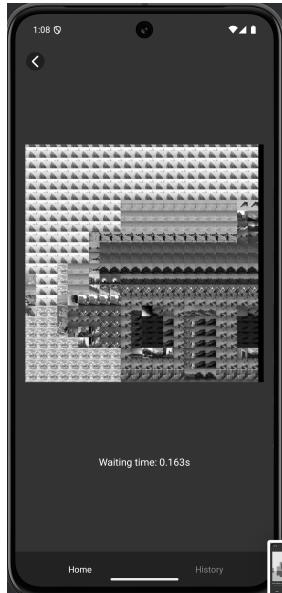


Figure 6: Image 1:
générée par méthode
moyenne

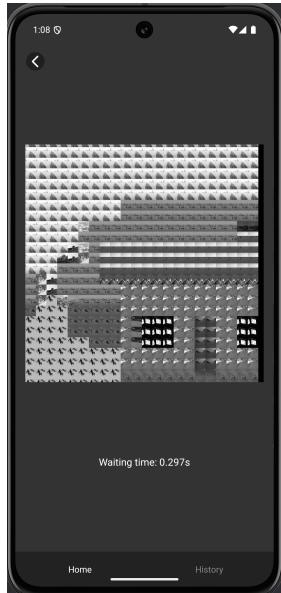


Figure 7: Image 2:
générée par méthode
histogramme

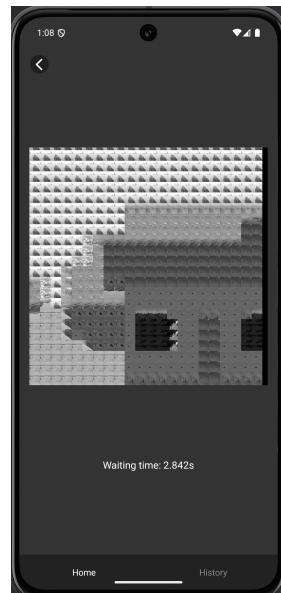


Figure 8: Image 3:
générée par méthode
distribution



Figure 9: slide SeekBar of size of block

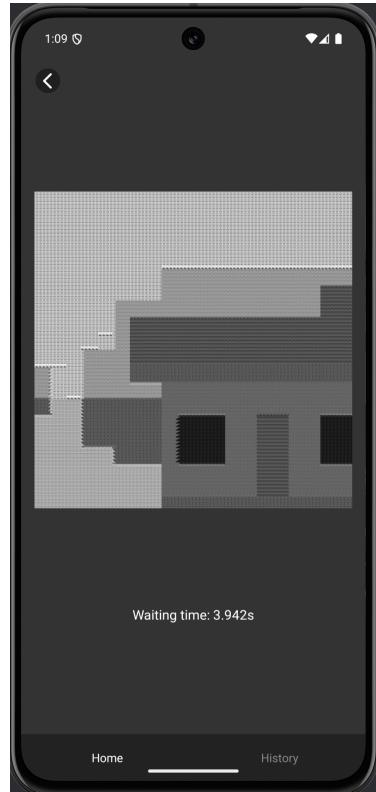


Figure 10: photo generated by block size:16



Figure 11: choose the function in COLOR

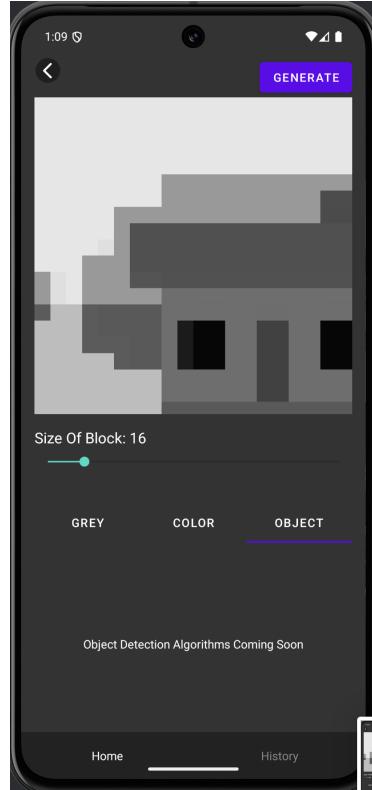


Figure 12: choose the function in OBJECT

3 Fonctionnalités réalisées

1. Sur la page d'accueil (Homepage) :

- Lorsqu'on clique sur le bouton d'accueil, deux options apparaissent : "Choose a photo" et "Take a photo".
- "Choose a photo" : L'application demande d'abord si l'accès à la galerie locale est autorisé. Si l'autorisation est accordée, l'utilisateur peut sélectionner une image dans la galerie.
- "Take a photo" : Après avoir obtenu l'autorisation, l'application permet de prendre une photo, et l'utilisateur peut également revenir à la page d'accueil.

2. Sur la page de sélection (SelectPage) :

- L'utilisateur peut voir un aperçu de l'image.

- Il peut ajuster la taille du bloc en utilisant un curseur (slider).
- En cliquant sur "GREY", "COLOR" ou "OBJECT", l'application applique la fonction correspondante.
- En cliquant sur la fonction, l'application génère une image de mosaïque avec la taille de bloc spécifiée.

3. Sur la page de génération :

- L'application affiche le temps d'attente pour la génération de l'image.

4 La liste TODO :

1. Sur la page d'historique (History) :

- Afficher la liste des mosaïques déjà générées.
- Pour chaque vignette, indiquer le PSNR obtenu, le temps de calcul et l'algorithme utilisé.
- Permettre de recliquer sur une vignette pour voir le détail (zoom, infos).

2. Sur la page de sélection (SelectPage) :

- Quand on choisit une méthode (Average, Histo, Distribution): changer la couleur du bouton actif pour qu'on sache ce qui est sélectionné.
- Compléter les onglets "Color" et "Object" avec leurs propres layouts et boutons d'algorithmes.
- Ajuster le SeekBar pour la taille de bloc et mettre à jour l'aperçu en temps réel si possible.

3. Sur la page de résultat (Resultat) :

- Afficher la mosaïque finale avec possibilité de zoom / dézoom (pinch-to-zoom).
- Bouton pour basculer entre l'image d'origine et la version mosaïque.
- Montrer les détails: PSNR, algorithme sélectionné, temps de calcul.
- Proposer de partager l'image (Intent ACTION_SEND) ou de l'enregistrer dans la galerie.