

UNIVERSITÉ DE MONTPELLIER
M2 IMAGINE



WMS Environment Analyzer

HAI918I : PROJET IMAGE

Étudiants :

Yahnis SAINT-VAL

Adrien HOULE

Encadrant :

William PUECH

Nicolas DIBOT

Année : 2023 – 2024

1 Informations

Dépôt git : https://github.com/HouleAdrien/WMS_Environment_Analyzer

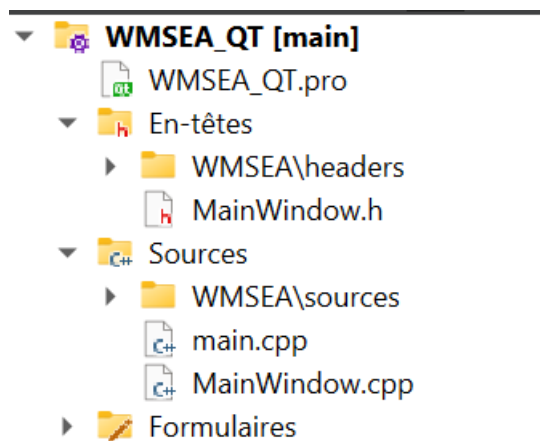
Pour le contexte cette semaine en entreprise Adrien a travaillé pour mettre en place le projet sur demande de son chef de projet c'est pour cela qu'il y a déjà beaucoup d'avancées.

Au cours de ce projet, chaque fonctionnalité sera développée dans une branche Git, puis fusionnée une fois vérifiée (la branche principale sera protégée). Veuillez respecter le format de message de commit suivant : préfixe - description du commit Exemple : Update - Mise à jour du requêtage asynchrone

- Préfixes possibles :
- Feature (nouvelle fonctionnalité)
- Update (mise à jour d'une fonctionnalité)
- Test (ajout de tests sur une fonctionnalité)
- Fix (correction d'un bogue)
- Remove (suppression d'une fonctionnalité)
- CI (uniquement pour les commits travaillant sur l'intégration continue)

2 Mise en place des prérequis

Pour démarrer le projet, nous devons d'abord mettre en place l'architecture du projet. Nous avons choisi d'organiser le projet de la manière suivante :



Nous avons créé un dossier WMSEA_QT qui contiendra les fichiers Qt, ainsi qu'un sous-dossier WMSEA qui contiendra la bibliothèque de traitement proprement dite. Ce dernier dossier se subdivise en deux sous-dossiers : "headers" et "sources". Cette structure nous permet de travailler à la fois dans Qt et de manière indépendante de Qt en même temps.

Ensuite, nous devons récupérer des données. Pour ce faire, nous utiliserons les données publiques des satellites Sentinel-1 et Sentinel-2 de l'ESA, téléchargeables depuis le lien suivant : <https://dataspace.copernicus.eu/>

De plus, pour comparer nos résultats, nous utiliserons la carte de couverture de l'ESA de 2021, disponible à l'adresse suivante : <https://worldcover2021.esa.int/>

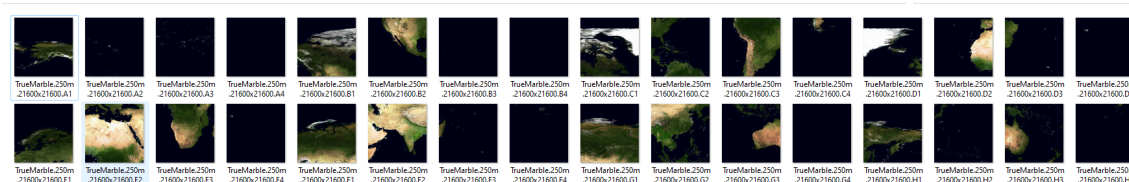
Cependant, nous n'allons pas créer exactement les mêmes classes. Il sera donc intéressant de comparer des classes communes, telles que l'occupation des forêts.

Pour commencer nous avons récupéré un dataset d'images publique avec une résolution satellite de 250 m / pixel s'appellant truemarble .

Nous avons donc récupéré la Terre en tuiles d'images de 21,600 x 21,600 en format TIFF (Tagged Image File Format).

Le format TIFF est un format d'image couramment utilisé en raison de sa haute qualité et de sa prise en charge de multiples couches. Il est idéal pour conserver des images sans perte de qualité.

Les tuiles ressemblent à ceci :



Cela nous donne une archive de 45 Go avec la Terre complète, qu'il faudra traiter pour les rendre utilisables dans notre contexte. L'utilisation du format TIFF garantit que nous ne perdons pas de qualité lors de la conversion et du traitement des images.

Cette archive contient donc 32 images TIFF de cette taille, et nous allons utiliser GDAL (Geospatial Data Abstraction Library) pour les convertir en format PPM (Portable Pixmap) et les diviser en images de la taille voulue. GDAL est un ensemble d'outils puissants pour travailler avec des données géospatiales, y compris des images TIFF. Il nous permet de gérer efficacement la conversion d'images, la géoréférenciation et d'autres opérations essentielles. L'utilisation de GDAL simplifie le processus de traitement des données géospatiales, ce qui est particulièrement bénéfique pour notre projet.

Pour cela nous avons mis en place le script bat suivant.

```

@echo off
setlocal enabledelayedexpansion

rem Spécifiez le chemin vers le dossier contenant les images TIFF
set "inputFolder=.\TM250\\"

rem Spécifiez la largeur et la hauteur de l'image d'origine
set "imageSize=21600"

rem Spécifiez la taille des tuiles voulues
set "tileSize=216"

rem Calculez le nombre de tuiles en largeur et en hauteur
set /a "numTilesX=!imageSize! / !tileSize!"
set /a "numTilesY=!imageSize! / !tileSize!"

rem Parcourez tous les fichiers TIFF dans le dossier
for %%f in ("%inputFolder%\*.tif") do (
    set "inputFile=%%~nf"
    set "outputFolder=%inputFolder%\!inputFile!"

    rem Créez un dossier pour chaque image
    mkdir "!outputFolder!"

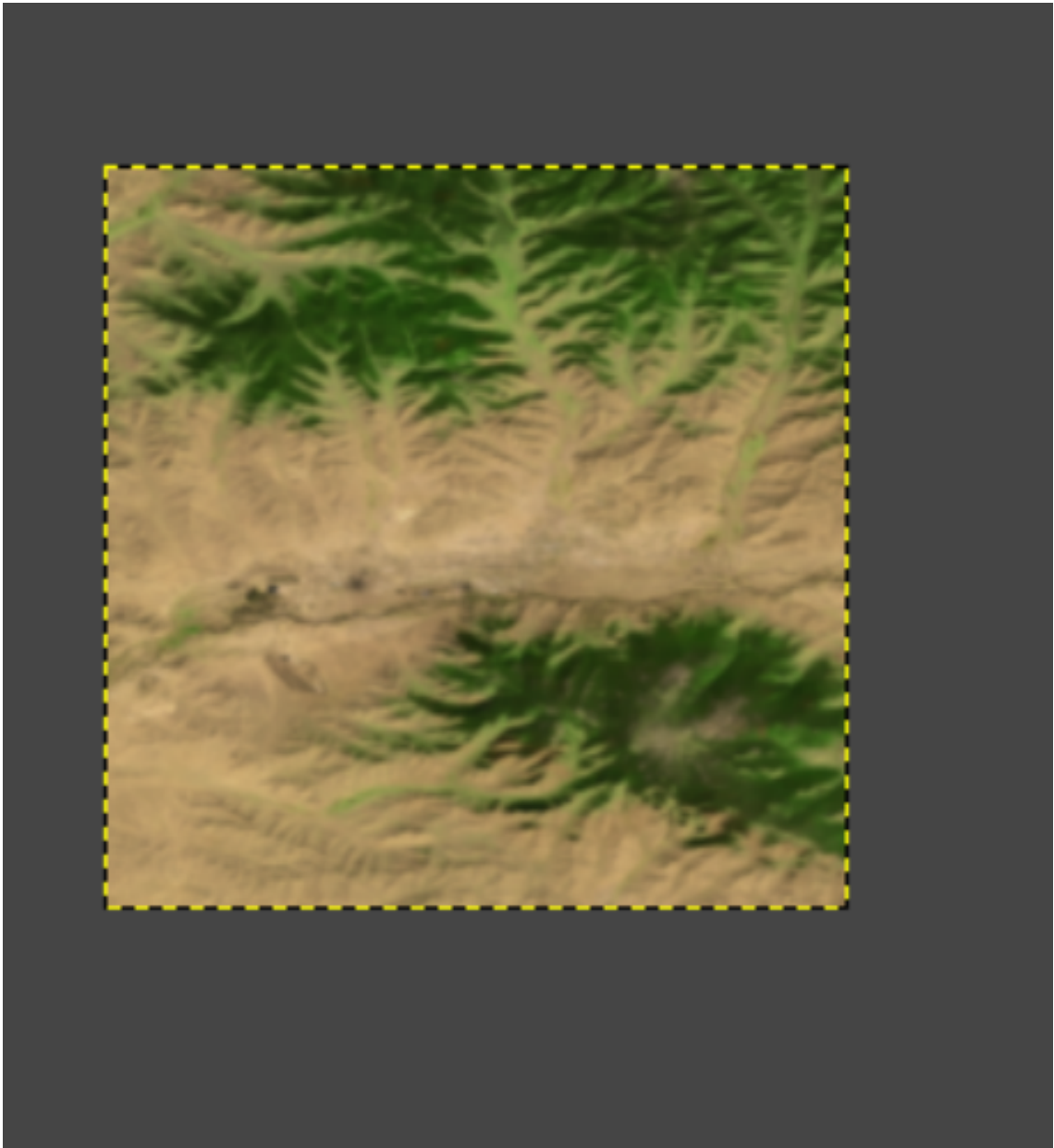
    rem Divisez l'image en tuiles et stockez-les dans le dossier
    for /l %%x in (0,%tileSize%,!imageSize!) do (
        for /l %%y in (0,%tileSize%,!imageSize!) do (
            set /a x_end=%%x+%tileSize%
            set /a y_end=%%y+%tileSize%
            gdal_translate -of PNM -srcwin %%x %%y %tileSize% %tileSize% "%%f" "!outputFolder!\!inputFile!_%%x_%%y.ppm"
        )
    )
)

```

Nous utiliserons le module `gdal_translate` dans le script `.bat` suivant, ce qui nous permet, en spécifiant un dossier de tuiles satellites `.tif`, de définir la résolution voulue et celle d'origine pour les diviser en petites tuiles de la résolution voulue et de les sauvegarder en images PPM.

Cela nous permet d'obtenir un dataset d'environ 320 000 (10k par image d'origine) tuiles PPM de 216x216 représentant le monde, et en plus de cela, GDAL génère aussi des fichiers `.xml` donnant les informations géographiques de la tuile 216x216 qui étaient présentes dans le TIFF d'origine.

Une tuile ressemblera à cela :



Le même processus sera fait sur des images orthophoto de l'ign sur un ou plusieurs département Français car la résolution est beaucoup plus élevée 10m/pixel, et donnera très probablement des résultats plus précis. Sauf que cette fois les tuiles seront en jpeg2000 avec des tailles différentes.

3 Semaine prochaine

Nous allons commencer à implémenter différents algorithmes de classifications afin de classer et détection différents zones sur les images de notre dataset avec les classes que l'on souhaite (qui restent encore à déterminer).

Nous allons commencer par implémenter un algorithme de type k-mean (en utilisant les couleurs), afin de déterminer "automatiquement" un certain nombre de classes, et ensuite voir ce que cela donne, et si les classes trouvées sont pertinentes.