

Projet : Le calcul de densité d'habitation à partir d'images satellitaires

Mode d'emploi :

Les modes d'exécution des différents scripts sont les suivant :

- Dans un noyau PYTHON : appel à la méthode de base
- En ligne de commande : via mode d'exécution

1. Téléchargement d'images satellitaires

a. Bing Satellite Maps

Nom du répertoire : « download_satellite_images\Bing-downloader »

Scripts	Details	Arguments	Exécution	Méthde de base
main.py	Permet de télécharger les images	bot_long bot_lat top_long top_lat	python main.py top_lat top_long bot_lat bot_long Exemple: Pour Le plateau, python main.py 5.311335 -4.034104 5.341868 -4.004153	Aucun
stitchTiles.py	Permet de réunifier les morceaux d'images	Aucun	Python stitchTiles.py	Aucun

Remorque : Les images ne doivent pas excéder la taille de 20,512 X 20,512.

Nécessite : opencv-python (pip install opencv-python)

Numpy (package de base)

b. Google Satellite Maps

Nom du répertoire : « download_satellite_images\GoogleMaps-downloader »

Scripts	Details	Arguments	Exécution
download_tiles_for_py3.py	Permet de télécharger les images	lon_start lat_stop lon_stop lat_start zoom-level	python download_tiles_for_py3.py top_lat top_long, bot_lat bot_long zoom Exemple pour le plateau : python download_tiles_for_py3.py -4.0279855151 5.3319522642 -4.0249599833 5.3350461774 19

Remarque : Ce script télécharge et construit l'image de la zone d'intérêt.

c. OSM satellite Maps

Répertoire : « download_satellite_images\OSM-tiles-downloader »

Scripts	Details	Arguments	Exécution
tiles_to_tiff.py	Permet de télécharger les images	lon_min lat_min lon_max lat_max zoom-level	python tiles_to_tiff.py lon_min lat_min lon_max lat_max zoom-level Exemple pour le plateau : python tiles_to_tiff.py 5.3319522642 -4.0279855151 5.3350461774 -4.0249599833 16

2. Traitement des images

Renseigne les traitements sur les images au format raster ou non.

a. Reprojection d'images

Répertoire : « Image-processing\image_reproject »

Scripts	Details	Arguments	Exécution
reproject_raster.py	Permet de reprojeter de façon automatique dans le crs de destination sans compression d'une image tiff	<i>tif_to_tiff</i> <i>path_to_new_tiff</i> <i>CRS_dst</i> (EPSG:32630)	python reproject_raster.py tif_to_tiff path_to_new_tiff CRS_dst

Remarque: Ce script est entièrement dépendant de la résolution de la taille de l'image. Pour des images de grande résolution, il faut envisager, l'outil **gdalwarp**, une extension de gdal :

```
gdalwarp -co compress=JPEG -co PHOTOMETRIC=YCBCR -co TILED=YES -co "BLOCKXSIZE=256" -co "BLOCKYSIZE=256" -s_srs crs_source -t_srs crs_dst 'src_tif' 'dst_tif'
```

srctif: Le chemin absolu de l'ancienne image

dst_tif : le chemin absolu de la nouvelle image

crs_source : Le crs source de l'image à reprojeter , voir les métadonnées de l'image

crs_dst : Le crs de destination

b. Extraction d'images

Répertoire : « Image-processing\image-clip »

Scripts	Details	Arguments	Exécution
image_clip.py	Permet d'extraire par un fichier GeoJson l'images référer par la géométrie	path_to_tif: chemin du fichier tiff geojson_url: chemin du fichier geojson new_tif_url : chemin du nouvel image	python image_clip.py path_to_tif geojson_url new_tif_url Exemple: Python image_clip.py tmp.tif gdf.GeoJson clip_tmp.tif

image_clip_gdf.py	Permet d'extraire par plusieurs géométries les images correspondant	path_to_tif: chemin du fichier tiff geojson_url: chemin du fichier geojson tiles_dir : chemin ou sauver les morceaux	python image_clip.py path_to_tif geojson_url tiles_dir Exemple: Python image_clip_gdf.py gdf_multiple.GeoJson tmp.tif tiles
-------------------	---------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------

c. Découpage

Répertoire : « project-details\Image-processing\image-clip »

Scripts	Details	Arguments	Exécution
split_raster.py	Permet de découper un fichier raster selon la taille spécifiée et de stocker les morceaux	path_to_tif path_to_save_tiles tile_size_by_256	python split_raster.py path_to_tif path_to_save_tiles tile_size_by_256 Exemple: python split_raster.py clip_tmp.tif tiles/tiles 256

d. Réunification d'images raster

Scripts	Details	Arguments	Exécution
merge_raster.py	Permet de réunir les morceaux d'images pour former une seule image	dir_of_tiles image_path	python merge_raster.py <dir_of_tiles> <image_path> Exemple : python merge_raster.py tiles clip_tmp.tif

Remarque : Les images à réunir doivent être obligatoirement au format « tiff ».

e. Transformation d'images à polygones

Répertoire : « Image-processing\image_to_polygons »

Scripts	Details	Arguments	Exécution
image_to_polygons.py	Permet de transformer une image raster à des polygones du même crs que celui de l'image	tif_url: Chemin ou url du fichier raster geojson_url : chemin du fichier en sortie	python image_to_polygons.py tif_url geojson_url python image_to_polygons.py '/content/tiff/inference_cocody_reproj.tif' 'f:/content/exports/buildigns_COCODY.geojson'

f. Extraction des formes des habitations

i. Par la méthode de canny

Répertoire : «Image-processing\canny»

Scripts	Details	Arguments	Exécution
canny_methd.py	Permet d'extraire les formes correspondant aux habitations	path_to_image path_to_result path_to_result2	Python canny_methd.py path_to_images path_to_result path_to_result1 Exemple : python canny_methd.py tiles tiles_res tiles_res_img

Remarque : Uniquement réservé aux fichiers d'extension « tif ».

Ajuster les condition de sélection en fonction des images à traiter

ii. Par la méthode du Pymeanshift

Répertoire : "Image-processing\watershed_meanshift"

Scripts	Details	Arguments	Exécution
meanshift.py	Permet d'extraire les formes correspondant aux habitations	path_to_image path_to_result path_to_result2	Python meanshift.py path_to_images path_to_result path_to_result1 Exemple : python meanshift.py tiles tiles_res tiles_res_img

3. Grattage des POI

Répertoire : « project-details\POI_crawler\GoogleMapsStreet-crawler »

Scripts	Details	Arguments	Exécution
GoogleMapCrawler.py	Permet de gratter Gogle Street Maps pour avoir les POIs désirés	path_to_keywords: chemin du fichier txt des mots clés filter_keyword critère de recherche	python GoogleMapCrawler.py fichier_txt_poi critère
merge.py	Permet de réunir les Poi collecté pour former un seul fichier.	Aucun	Python merge.py

Remarque : Il est nécessaire de créer un répertoire « data » pour contenir les POIs selon les mots clés. Le script « merge.py » produit en sortie le fichier « merged_frame.csv » qui contient tous les pois.

4. Les modèles

a. Détection des zones habitées

Répertoire : « classification\buildingNoBuilding »

Scripts	Details	Arguments	Exécution
buildingNoBuilding.py	Permet de classifier des images en taille 256x256	path_to_images: chemin des images ext: extension des images	python buildingnobuilding.py path_to_images ext

			Exemple: python buildingnobuilding.py /images.tif
--	--	--	---------------------------------------------------

Remarque : Les images ne peuvent obligatoirement être de taille 256x256.

b. Segmentation des habitations

Répertoire : models\segmentation\fastai »

Scripts	Details	Arguments	Exécution
inference_to_images.py	Permet de faire la prédiction d'images. En sortie, on obtient des images	path_to_images: chemin des images path_to_results: chemin du résultat	python inference_to_geojson.py path_to_images path_to_results Exemple: python inference_to_images.py /images /results
inference_to_geojson.py	Permet de faire la prédiction d'images. En sortie, on obtient des fichiers GeoJson	path_to_images: chemin des images path_to_results: chemin du résultat	python inference_to_geojson.py /images /results Exemple: python inference_to_images.py /images /results

Remarque : Les images doivent être au format 256x256.

Le scripts "inference_to_geojson.py" peut produire des erreurs sur jupyter

5. Estimation

a. Azimuth

Scripts	Details	Arguments	Exécution
azimuth_estimation.py	Permet d'estimer l'azimuth	Long: longitude Lat: latitude date_start: date de début date_end: date de fin spatial_resolution: resolution spatiale max_elevation: élévation maximale path_to_results: chemin du fichier csv des angles	python azimuth_estimation.py long lat date_start date_end spatial_resolution max_elevation path_to_results Exemple: python azimuth_estimation.py - 3.9837119579315186 5.301122665405273 "2008-01-01 08:00:00" "2008-01-01 18:00:00" 2.5 az.csv

b. La taille des habitations

Scripts	Details	Arguments	Exécution
height_estimation.py	Permet d'estimer la taille	img_dir : Répertoire des images ext : Extension des fichiers azimuth : angle azimuth	python height_estimation.py img_dir ext azimuth Exemple: %run -i height_estimation.py tiles ".tif" "178.3282145189649"

c. Household counts

Scripts	Details	Arguments	Exécution
household_estimation.py	Permet d'estimer les ménages	img_dir : Répertoire des images ext : Extension des fichiers azimuth : angle azimuth	python height_estimation.py img_dir ext azimuth Exemple: %run -i height_estimation.py tiles ".tif" "178.3282145189649"

Quelques Commandes utiles

Ces commandes seront utiles sur colab.

!gsutil cp R chemin_source chemin_destination : Pour copier un dossier

!gsutil cp chemin_du_fichier chemin_de_destination : Pour copier un fichier

!zip -r nom_du_fichier.zip chemin_du_fichier : Pour zipper un fichier

!unzip fichier.zip : Pour dézipper

!unrar x fichier.zip : Pour dézipper

%run -i python_script_.py : permet d'exécuter dans un notebook jupyter un script.

Pour télécharger directement un fichier partager depuis un drive

```
# Install PyDrive  
  
!pip install PyDrive  
  
#Import modules  
  
#Ex link : https://drive.google.com/file/d/1c7Ffo1Go1dtUpKcSWxdbdVyW4dfhEoUp/view?usp=sharing  
  
# Get the id from the link 1c7Ffo1Go1dtUpKcSWxdbdVyW4dfhEoUp  
  
!gdown --id id_from_url
```

Liens

[Lien 1](#) : Permet de trouver la boîte des coordonnées en long, lat min, max

[Lien 2](#) : Pour la classification et segmentation non supervisée

[Lien 3](#) : Pour les images de Cocody et Bingerville

[Lien 4](#) : Pour le modèle de segmentation

[Lien 5](#) : Pour le modèle de classification