

Détection de changements des milieux urbains : application pour la ville de New Delhi, Inde (Tuile 43RGM)

Données à télécharger ici : [<https://enlive.live.unistra.fr/index.php/s/Wtgfwib60NR3bbq>]

New Delhi fait partie des villes ayant l'étalement urbain le plus rapide, selon la World Population Review [<https://worldpopulationreview.com/world-cities/delhi-population>]. En 2021, New Delhi compte 6 millions d'habitants de plus qu'en 2016 !

Vous avez à votre disposition, deux images satellites 43RGM¹ Sentinel-2 de niveau 1C acquises le 04/06/2016 et le 29/06/2021. L'objectif de ce TP est de créer une carte d'évolution/de changements de la tâche artificialisée pour la ville de New Delhi. Pour cela, nous allons calculer le NDBI (Normalized Difference Built-up Index) [<https://www.linkedin.com/pulse/ndvi-ndbi-ndwi-calculation-using-landsat-7-8-tek-bahadur-kshetri/>] pour chacune de nos deux dates. Ensuite, nous pourrions calculer une carte d'agrégats pour cartographier les changements des milieux urbains. Enfin, nous calculerons la surface bâtie pour chaque date pour la ville de New Delhi.

Les images Sentinel-2 sont de niveau 1C², les fichiers se trouvent dans le dossier IMG_DATA accessible via ces chemins :

```
1 S2A_MSIL1C_20210529T052651_N0300_R105_T43RGM_20210529T072625.SAFE/GRANULE/  
  L1C_T43RGM_A030990_20210529T053826/IMG_DATA  
2  
3 S2A_MSIL1C_20160604T052652_N0202_R105_T43RGM_20160604T053931.SAFE/GRANULE/  
  L1C_T43RGM_A004964_20160604T053931/IMG_DATA
```

Les images sont au format *jp2* [<https://fr.wikipedia.org/wiki/JP2>] qui est un format d'image plus léger que *GeoTIFF*.

Conversion des images en GeoTIFF

Dans un premier temps, je vous invite à convertir toutes les bandes de chaque image au format GeoTIFF avec la commande *gdal_translate* [https://gdal.org/programs/gdal_translate.html].

Note : Vous pouvez soit les convertir une par une, soit faire une boucle en Python sur toutes les images du dossier.

¹[https://maps.eatlas.org.au/index.html?intro=false&z=7&ll=146.90137,-19.07287&l0=ea_ref%3AWorld_ESA_Sentinel-2-tiling-grid_Poly,ea_ea-be%3AWorld_Bright-Earth-e-Atlas-basemap,google_HYBRID,google_TERRAIN,google_SATELLITE,google_ROADMAP&v0=,,f,f,f,f,f]

²[<https://sentinel.esa.int/web/sentinel/sentinel-data-access/sentinel-products/sentinel-2-data-products/collection-0-level-1c>]

Calcul du NDBI

Le NDBI fonctionne de la même façon que le NDVI. Il se calcule en utilisant la bande du SWIR et la bande du NIR. Pour Sentinel-2, nous utiliserons donc les bandes 11 (SWIR1) et 8 (NIR) (cf. Figure 1 Cours 2). Vous remarquerez que les 2 bandes ne possèdent pas la même résolution spatiale. Vous devrez, avant calcul de l'indice, changer la taille de la bande 11. Pour cela, plusieurs méthodes s'offrent à vous. Je vous invite à utiliser la fonction ci-dessous qui prend en entrée le *dataset* de la bande 11 (cf. Cours 2).

```

1 #Installez scipy : pip install scipy
2 from scipy.ndimage import zoom
3
4 def resize_bands(dataset_raster):
5     """
6     Resize bands to match 10m spatial resolution.
7     @params:
8         dataset_raster - Required : dataset raster
9     """
10    gt = dataset_raster.GetGeoTransform()
11    res_x = int(gt[1])
12    res_y = int(abs(gt[-1]))
13
14    _res_band = None
15    if res_x != 10 or res_y != 10:
16        ratio = float(res_x / 10)
17        to_resample = dataset_raster.GetRasterBand(1).ReadAsArray()
18        _res_band = np.array(zoom(to_resample, ratio, order=0))
19    else:
20        _res_band = np.array(dataset_raster.GetRasterBand(1).ReadAsArray())
21
22    return _res_band

```

Note : Vous n'êtes pas obligé d'écrire la bande ré-échantillonnée sur votre disque, vous pouvez conserver la matrice côté code uniquement.

Calculez ensuite le NDBI pour chaque date :

$$NDBI = \frac{SWIR - NIR}{SWIR + NIR}$$

où *SWIR* correspond à la bande 11 et *NIR* à la bande 8.

Binarisez vos résultats en fonction d'un seuil. Généralement, le seuil de binarisation pour l'extraction des surfaces artificialisées à partir du *NDBI* est de 0.4. Vous pouvez tester différents seuils sur des *subset* (sous ensemble de 1000 pixels de côté par exemple) et afficher la composition colorée RGB ou IRRG :

```

1 #Installez matplotlib : pip install matplotlib
2
3 f, axarr = plt.subplots(1,2)

```

```
4 axarr[0].imshow((RGB * 255).astype(np.uint8))
5 axarr[1].imshow(res_ndvi_loop, cmap = "gray", vmin=-1, vmax=1)
```

où RGB correspond à un *stack* en trois dimensions de la bande du rouge, du vert et de bleu (dans cet ordre !!) et de l'image Sentinel-2.

Indice :

```
1 def rescale(arr):
2     #Normalisation des valeurs de reflectance entre 0 et 1 pour un
    affichage RGB sur matplotlib
3     arr_min = arr.min()
4     arr_max = arr.max()
5     return (arr - arr_min) / (arr_max - arr_min)
6
7 rgb = np.dstack((rescale(red), rescale(green), rescale(blue)))
8 #OR
9 irrg = np.dstack((rescale(pir), rescale(red), rescale(green)))
```

Si la visualisation avec *matplotlib* vous semble trop complexe, testez les seuils en écrivant sur votre disque le résultat et visualisez le sur **QGIS**.

Sauvegardez les résultats finaux sur votre disque (cf. Cours 2).

Création d'une carte d'agrégats

A partir de vos *NDBI* seuillés, faites une carte d'agrégats permettant de mettre en avant les pixels "tâches artificialisées" de 2016 avec la valeur 1 et les pixels de 2021 avec la valeur 2.

Indice : les pixels de 2016 sont, entre autre, les pixels communs à 2016 et 2021.

Sauvegardez les résultats sur votre disque.

Découpage des résultats avec l'emprise *shapefile* de la ville

Vous avez à votre disposition un *shapefile* *Delhi_Boundary.shp*. Découpez votre carte d'agrégats avec le *shapefile* à l'aide de *gdalwarp* [<https://gdal.org/programs/gdalwarp.html>]

Indice : regardez bien du côté de *-cutline* et *-crop_to_cutline* :-)

Bonus : Interprétez les résultats

Interprétez vos résultats. Qu'en pensez-vous ? Est-ce une bonne méthode pour la détection de changements ? Un exemple de méthode pour la détection de changements : [<https://>

[//www.sciencedirect.com/science/article/pii/S0143622810001451](https://www.sciencedirect.com/science/article/pii/S0143622810001451) |