

TD Modélisation - Informatique:

Analyse d'image (2h)

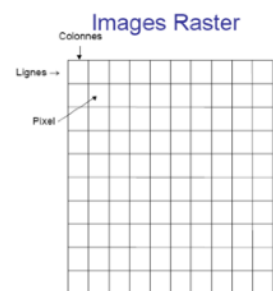
Objectifs pédagogiques de cette séance:

- Charger une image satellitaire avec la bibliothèque **GDAL**
- Savoir manipuler des images: extraction des bandes, affichage d'histogramme, correction de contrastes, binarisation, et seuillage.

Préambule sur les images

Un fichier image (un **raster**) comporte en général 2 parties :

- **La grille de valeurs:** une matrice 2D (si bande unique) ou 3D (si plusieurs bandes) de valeurs numériques. Chaque pixel a une taille fixe, et chaque pixel contient une valeur numérique: luminosité, température, altitudes, etc.
- **les métadonnées (metadata/header):** situées soit dans un fichier auxiliaire (.meta, .hdr, etc.) ou dans l' « entête » du fichier binaire, elle contiennent des informations sur le fichier : taille de l'image, taille du pixel, nombre de bits par pixel, système de coordonnées (datum+projection), emprise spatiale (origine), nombre de bande, valeurs spéciales, etc.



Les rasters peuvent être enregistrés dans divers formats d'image tels que BMP, JPEG, PNG... Mais aussi dans des formats spécialisés pour les systèmes d'information géographique (SIG) comme GeoTIFF, Grid... Des informations géospatiales sont alors contenues dans les métadonnées. Voir l'ensemble des formats sur: <https://gdal.org/en/latest/drivers/raster/index.html>

La taille du pixel (la résolution spatiale) dépend de la largeur/hauteur de l'image et du nombre de colonnes/lignes.

La profondeur de l'image (la résolution radiométrique), l'image correspond au nombre de bits stockés par pixel:

- Entier sur 1 octet / 8 bits: (0 - 255)
- Entier sur 2 octets / 16 bits = short (0 - 65535)
- Nombre flottant sur 4 octets /float32: 8 chiffres significatifs

La taille du fichier est donc égal au nb de lignes * nb de colonnes * taille du type de données * nb de bande

Préambule sur la bibliothèque GDAL

GDAL est une bibliothèque open-source très puissante utilisée pour lire, écrire, manipuler et transformer des données géospatiales. Elle supporte une large gamme de formats de fichiers raster et vectoriels. Elle est par exemple utilisée dans QGIS. Dans ce projet, vous allez utiliser **GDAL** pour manipuler des images au format **GEOTiff**, un format d'image raster.

Documentation : https://pcjericks.github.io/py-gdalogr-cookbook/raster_layers.html#close-a-raster-dataset

Pour utiliser GDAL, il faut l'importer :

```
from osgeo import gdal
```

Chargement d'une image:

- Ouverture du fichier : `ds = gdal.Open ( )`
- Accès aux propriétés de l'image :
 - Nombre de colonnes et de lignes : `ds.RasterXSize, ds.RasterYSize`
 - Projection : `ds.GetProjection ( )`
 - Géoréférencement : `ds.GetGeoTransform ( )` renvoie un tuple de 6 éléments : (originX, pixelWidth, rotX, originY, rotY, pixelHeight)
 - le nombre de bandes (par exemple, rouge, vert et bleu) : `ds.RasterCount`
- Accès à une bande de l'image :
 - Accès à la bande 1: `band1 = ds.GetRasterBand(1)`
 - Type de données de la bande : `gdal.GetDataTypeName(band1.DataType)`
 - Lire les valeurs de la bande comme un tableau NumPy: `array = band1.ReadAsArray ( )`

Une fois les bandes récupérées sous forme de tableaux **NumPy**, on peut les manipuler comme n'importe quel autre tableau numérique.

Préambule sur les images Sentinel-2

Les images « Sentinel » constituent le volet spatial du [programme Copernicus](https://www.copernicus.eu/en) de l'Union européenne: <https://www.copernicus.eu/en>. Les satellites Sentinel-2 ont pour objectifs l'observation des sols (utilisation des sols, végétation, zones côtières, fleuves, etc.) ainsi que le traitement des situations d'urgence (catastrophes naturelles...). Le premier satellite a été placé en orbite en juin 2015. Ils sont équipés d'un imageur multispectral (MSI) qui fournit des vues dans 13 bandes spectrales en lumière visible et proche infrarouge. Le tableau suivant résume les caractéristiques de chaque bande spectrale des images Sentinel-2.

Analyse d'une image Sentinel-2

Vous disposez sur **Arche** d'un découpage des 4 bandes Rouge/Vert/Bleu/IR d'une image Sentinel-2 aux alentours d'Athènes (Grèce) téléchargée sur le portail Copernicus : (<https://browser.dataspace.copernicus.eu/>): 2025-07-25-00_Sentinel-2_L2A_B4B3B2B8.tiff

Étape 1 – Exploration des métadonnées et affichage d'une bande

Bandes spectrales de l'instrument MSI à bord de Sentinel-2 ¹¹					
Bandes Sentinel-2	Sentinel-2A		Sentinel-2B		Résolution spatiale (m)
	Longueur d'onde centrale (nm)	Largeur de bande (nm)	Longueur d'onde centrale (nm)	Largeur de bande (nm)	
Bande 1 – Aérosol côtier	442.7	21	442.2	21	60
Bande 2 – Bleu	492.4	66	492.1	66	10
Bande 3 – Vert	559.8	36	559.0	36	10
Bande 4 – Rouge	664.6	31	664.9	31	10
Bande 5 – Végétation "red edge"	704.1	15	703.8	16	20
Bande 6 – Végétation "red edge"	740.5	15	739.1	15	20
Bande 7 – Végétation "red edge"	782.8	20	779.7	20	20
Bande 8 – PIR	832.8	106	832.9	106	10
Bande 8A – PIR "étroit"	864.7	21	864.0	22	20
Bande 9 – Vapeur d'eau	945.1	20	943.2	21	60
Bande 10 – SWIR – Cirrus	1373.5	31	1376.9	30	60
Bande 11 – SWIR	1613.7	91	1610.4	94	20
Bande 12 – SWIR	2202.4	175	2185.7	185	20

Figure 1: Bandes spectrales de Sentinel-2

- Ouvrir le fichier TIFF avec GDAL.

👉 Quelle est la taille de l'image, sa résolution spatiale, sa résolution radiométrique, ainsi que son système de coordonnées ?

- Charger la bande 1 (correspondant à la bande Sentinel-2 **B4 – Rouge**) dans un tableau NumPy appelé `array`
- Afficher cette bande en niveau de gris avec: `plt.imshow(array, cmap="gray", vmin=0, vmax=255)`. Tester différentes valeurs min et max pour améliorer les contrastes.
- Calculer et afficher son histogramme avec: `plt.hist()`. Attention, `plt.hist()` ne travaille qu'avec un tableau 1D, il faut donc aplatir le tableau avec `ravel()` ou `flatten()`.

👉 À quels types de surface correspondent les zones sombres (proches de 0) ou claires (proches de 255) ?

Étape 2 – Corrections de contraste

- Il existe de nombreuses fonctions d'étirement radiométrique: <https://www.nv5geospatialsoftware.com/docs/BackgroundStretchTypes.html>. Ici, nous allons appliquer une **normalisation linéaire** (min-max) pour étendre la dynamique de la bande B4 (Rouge).
- Pour cela nous allons:
 - Normaliser les valeurs de la bande par max-min.
 - Les valeurs min et max peuvent être les valeurs minimales ou maximales de la bande. Elles peuvent aussi être ajustées manuellement ou bien correspondre 2ème ou 98ème percentile, que vous pouvez calculer grâce à la commande: `np.percentile(array, (2, 98))`.
 - Clipper les valeur du tableau entre 0 et 1 grâce à la commande `np.clip(..., 0, 1)`.

👉 Ecrire une fonction d'étirement linéaire des contrastes entre percentile prenant en entrée un tableau NumPy et retournant en sortie le tableau étiré. Comparer visuellement et interpréter les différences.

Étape 3 – Binarisation et seuillage

- Choisir un **seuil radiométrique** (manuel à partir de l'histogramme ou un seuil simple = moyenne - $0.5 \times \text{ecart-type}$).
- Créer une image binaire (0 = en dessous du seuil, 1 = au-dessus) de telle sorte que les valeurs nulles correspondent à de l'Eau et les valeurs 1 correspondent à de la Terre.

Étape 4 – Image miroir

Écrire une fonction `set_mirror` qui retourne une image miroir avec la commande: `np.flipud()`

Étape 5 – Composition colorée

- Créer une **composition en couleurs naturelles** (RGB : B4, B3, B2). Pour cela vous devez stacker les 3 bandes ensemble (créer une matrice 3D de taille (nb_lignes, nb_colonnes, 3)) avec la commande:

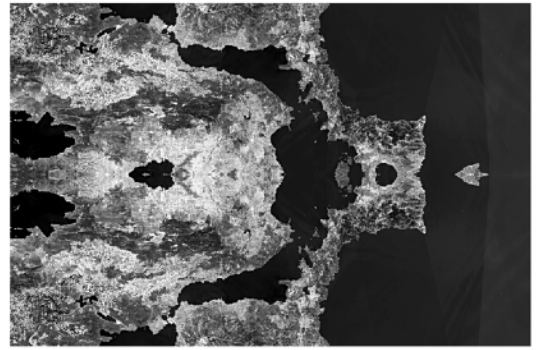
```
np.dstack((red_stretch, green_stretch,
blue_stretch)). Afficher le résultat avec
plt.imshow().
```

- Créer une **composition infrarouge couleur** (RGB : B8, B4, B3).

👉 Comparer les deux compositions : que met en évidence la combinaison infrarouge ?

Résultat attendu:

Image miroir



Composition colorée - Vraie couleur (R, G, B)



Composition colorée - Infrarouge (NIR, R, G)

