

## Classification supervisée 'objet' (OBIA)

La classification orientée objet (OBIA - *Object Based Image Analysis*) est une méthode de traitement d'images de télédétection qui s'appuie sur des concepts de l'analyse d'images et de la reconnaissance de formes. Elle diffère de la classification pixel à pixel, en utilisant des objets qui sont des ensembles de pixels qui sont groupés en fonction de leurs propriétés géométriques, spectrales et contextuelles. En utilisant cette méthode, il est possible de prendre en compte des informations supplémentaires, telles que la texture, la forme, la taille et la disposition spatiale des objets, pour améliorer la précision de la classification. L'OBIA est une technique largement utilisée dans la télédétection pour une variété d'applications, notamment la cartographie des sols, la gestion des ressources naturelles, la surveillance environnementale et la planification urbaine.

La classification orientée objet permet de mieux représenter la réalité du terrain, car elle permet d'identifier et de classer des objets entiers plutôt que de simplement considérer chaque pixel individuellement. Cette méthode prend également en compte les relations spatiales entre les objets, leur contexte et leur environnement, ce qui permet une classification plus précise et plus détaillée.

L'OBIA est souvent utilisée en combinaison avec des images satellitaires de haute résolution et des données géospatiales, telles que des données topographiques ou des données climatiques, pour améliorer la précision de la classification. Cette méthode nécessite toutefois une expertise en télédétection et en traitement d'images, car elle implique des étapes de segmentation, d'extraction de caractéristiques et de classification.

Enfin, il convient de noter que l'OBIA est une méthode qui évolue constamment, avec l'avancée de la technologie et des algorithmes de traitement d'images. Elle est donc un domaine de recherche actif en télédétection, avec de nombreuses perspectives d'amélioration et de développement pour les années à venir.

**Application 2** : Classification 'objet' par méthode supervisée d'une image Sentinel-2 pour la classification de l'occupation des sols. Application avec une méthode *Random Forest*.

OTB est déjà configuré sur les machines à la faculté. Si vous souhaitez l'installer sur une machine Windows, Linux ou MacOS, suivez ces guides :

[ <https://www.sigterritoires.fr/index.php/ajouter-orfeo-toolbox-dans-qgis-3-x/> ] et [ <https://www.orfeo-toolbox.org/CookBook-6.6.1/QGIS-interface.html> ]

## Étape 1 : Application d'une méthode de segmentation *Large Scale Meanshift*

Vous avez à votre disposition un *subset* d'une image Sentinel-2 :

*Sentinel2A\_20170619\_10m\_L93\_EMS.tif*. Pour la télécharger :

[ <https://enlive.live.unistra.fr/index.php/s/n9J2v0EmzBg3z6F> ]

Pour une approche 'objet', il est indispensable de segmenter l'image en régions homogènes. Avec Orfeo Toolbox (OTB), appliquer une méthode de segmentation *Large Scale Meanshift*. Aller dans la *Toolbox* et dérouler OTB pour trouver la méthode **LargeScaleMeanShift** (Figure 1).

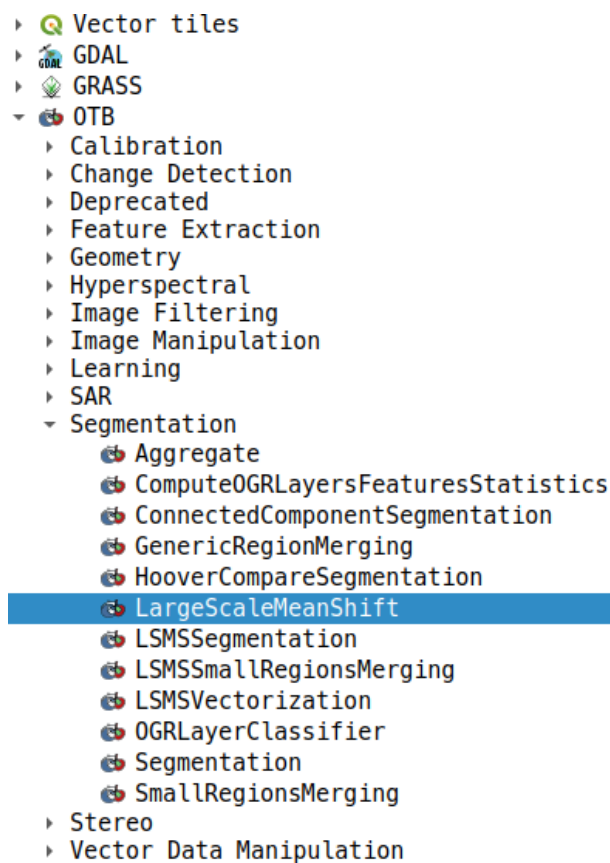


FIGURE 1 – *LargeScaleMeanShift* sur OTB

Choisi les paramètres suivants (pour l'image Sentinel-2) :

- **Spatial radius** : 8
- **Range radius** : 100
- **Minimum Segment Size** : 300
- **Size of tiles in pixel** : 500
- **Output mode** : vector

Sauvegarder le résultat dans le dossier de travail. Ouvrir le fichier de segmentation sur QGIS et regarder la table attributaire. Quelles sont les colonnes calculées ? Que représente *meanBX* et *varBX* ?

**Note :** les autres méthodes de segmentations (par exemple : *watershed*) n'ajoutent pas directement les métriques calculés sur l'images pour chaque polygone. En revanche, il est possible de sauvegarder la segmentation en *raster* (*Processing mode*) et de calculer des statistiques zonales avec la fonction *LSMSVectorization*.

## Étape 2 : Création d'une donnée de référence

Tout comme l'approche 'pixel', l'approche 'objet' nécessite une donnée de référence en entrée du modèle. Pour sélectionner les polygones d'entraînement, créer un nouveau fichier vecteur d'entraînement (*learn*) et d'évaluation (*control*). A partir de l'image Sentinel-2 sur l'Eurométropole de Strasbourg, digitaliser des points d'entraînement et d'évaluation (Figure 2). Dans un premier temps, travailler sur quatre classe d'occupation des sols : **Urbain**, **Surfaces agricoles**, **Forêt** et **Eau**.



FIGURE 2 – Création des points d'entraînement (et de validation)

## Étape 3 : Joindre les points d'entraînement et les attributs issues de la segmentation

Dans un premier temps, il s'agit de faire une sélection par location des polygones conte-

nant les attributs calculés par la méthode de segmentation :

*Vector > Reasearch Tools > Select by Location*

Sélectionner les *features* du fichier vectoriel de segmentation en comparant avec les *features* du fichier vectoriel contenant les points d'entraînement. Sauvegarder les *features* sélectionnées dans un fichier (par exemple : *learn\_sample\_from\_segments.shp*). Appliquer la même démarche pour les points de validation.

Faire une jointure spatiale (*Join Attributes by Location*) pour ajouter l'id de la classe contenue dans le fichier de points (*insitu\_learn\_samples.shp*).

Le nommer *learn\_samples\_from\_segments\_with\_class.shp*. Appliquer la même démarche pour les données de validation.

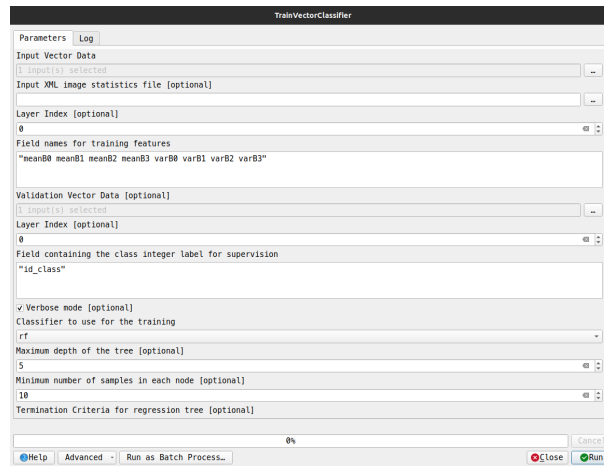
A l'issue de cette troisième étape, le fichier contient les polygones d'entraînement (ou de validation) avec les statistiques zonales (*mean* et *variance*) ainsi que l'id de la classe (Figure 3).

	label	nbPixels	meanB0	meanB1	meanB2	meanB3	varB0	varB1	varB2	varB3	id_class
1	7362	319	417,94982...	740,39813...	718,70532...	4097,0346...	10826,477...	28041,207...	61551,445...	54542,492...	2
2	10487	767	368,55670...	742,86437...	477,74838...	4887,9140...	9802,8095...	16236,574...	68574,640...	287888,37...	2
3	16171	547	369,32540...	658,00915...	577,46618...	3906,4423...	3684,6887...	7038,0659...	13753,347...	12366,769...	2
4	22261	520	924,73846...	1415,5039...	1699,1904...	3646,4633...	38117,425...	58911,074...	120821,14...	141547,78...	2
5	39037	321	120,61993...	288,91589...	94,433021...	4448,0249...	133,05468...	608,28125...	88,627342...	59401,601...	3
6	32215	578	126,26470...	301,98269...	97,486160...	4026,0588...	317,44195...	1439,5909...	578,30242...	98855,929...	3
7	42345	385	522,59478...	782,21557...	818,91168...	2929,1818...	30278,833...	32686,500...	72145,789...	390506,00...	1
8	68836	995	257,85125...	503,02212...	295,49850...	4052,2412...	2947,0905...	6556,9736...	8900,0000...	97916,648...	2

FIGURE 3 – Structure finale attendue des fichier d'entraînement et de validation

## Étape 4 : Apprentissage du modèle par fichier vecteur

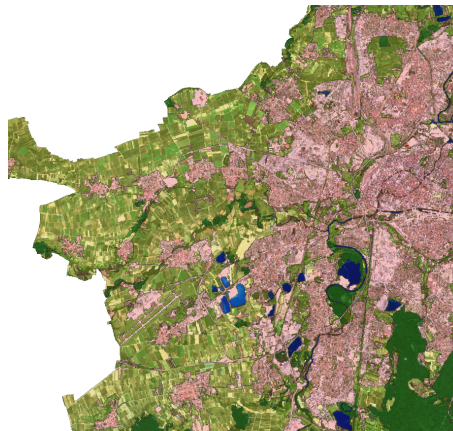
Lancer l'apprentissage du modèle avec l'outil *TrainVectorClassifier* pour le fichier d'entraînement *learn\_samples\_from\_segments\_with\_class.shp*. Ajouter les colonnes du fichier utilisées pour l'entraînement (meanB0 meanB1 meanB2 meanB3 varB0 varB1 varB2 varB3), le fichier de validation (*control\_samples\_from\_segments\_with\_class.shp*) et le champs contenant l'id des classes (id\_class) (Figure 4).

FIGURE 4 – Entraînement du modèles *Random Forest*

Sauvegarder le modèle dans un fichier (*model\_rf\_sentinel2.rf*) et la matrice de confusion dans un fichier texte (*confusion\_matrix\_rf\_sentinel2.txt*).

## Étape 5 : Application du modèle

Ouvrir le module *VectorClassifier*. Renseigner le fichier de segmentation en entrée, le fichier du modèle, le champs de classification (*predict\_class*) et le fichier de classification en sortie (*output\_rf\_Sentinel2.shp*). Visualiser le résultat et attribuer une typologie. Effectuer un *dissolve* sur la colonne *predict\_class* (Figure 5).

FIGURE 5 – Classification finale après *dissolve*

Reprojeter la données en UTM 32 Nord (EPSG :32632) puis rasteriser la données finale (Figure 6).

**Note :** Nous souhaitons une données avec une résolution spatiale de 10m !

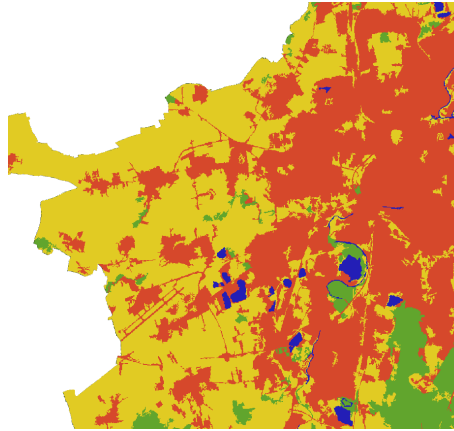


FIGURE 6 – Classification finale raster

## Étape 6 : Protocole de tests

Exemples pour démarrer le protocole de tests :

Classes	Attributs	Algorithmes
<u>Test 1</u> : 4 classes thématiques <ul style="list-style-type: none"> <li>• Urbain</li> <li>• Surfaces Agricoles</li> <li>• Forêts</li> <li>• Surfaces en eau</li> </ul>	4 bandes spectrales	Random Forest
<u>Test 1</u> : 4 classes thématiques <ul style="list-style-type: none"> <li>• Urbain</li> <li>• Surfaces Agricoles</li> <li>• Forêts</li> <li>• Surfaces en eau</li> </ul>	4 bandes spectrales + NDVI	Random Forest
<u>Test 3</u> : faire varier le nombre de classes thématiques <ul style="list-style-type: none"> <li>• Urbain Dense</li> <li>• Urbain Diffus</li> <li>• Zones industrielles</li> <li>• Surfaces Agricoles</li> <li>• Forêts</li> <li>• Surfaces en eau</li> <li>• ...</li> </ul>	4 bandes spectrales + NDVI	Random Forest
<u>Test 4</u> : ...	...	...

**Note** : Si vous souhaitez automatiser les traitements lorsque vous changez la typologie, vous pouvez utiliser des *Model Builder*.

## Étape 7 : Effectuer une classification 'pixel' avec l'algorithme *Random Forest*

Effectuer une classification 'pixel' sur SNAP ou sur QGIS à partir des paramètres du meilleur test issu de l'approche 'objet'. Deux approches sont possibles :

- Digitaliser les échantillons d'entraînement pour les mêmes classes que l'approche 'objet'
- Utiliser le même fichier d'entraînement que pour l'approche 'objet' en faisant un *dissolve* pour obtenir  $n$  classes d'occupation des sols

Sur QGIS (et OTB), utiliser *TrainImagesClassifier* [ [https://www.orfeo-toolbox.org/CookBook/Applications/app\\_TrainImagesClassifier.html](https://www.orfeo-toolbox.org/CookBook/Applications/app_TrainImagesClassifier.html) ] en indiquant le fichier image, le fichier vecteur contenant les échantillons et le nom de colonne contenant les identifiants des classes.

## Étape 8 : Évaluer les classifications

**Évaluation quantitative :** Sur QGIS (et OTB), utiliser *ComputeConfusionMatrix* [ [https://www.orfeo-toolbox.org/CookBook-6.2/Applications/app\\_ComputeConfusionMatrix.html](https://www.orfeo-toolbox.org/CookBook-6.2/Applications/app_ComputeConfusionMatrix.html) ] pour calculer une matrice de confusion. En entrée, utiliser la classification au format *raster* et le fichier vecteur contenant les polygones de validation. Sauvegarder **la (ou les) matrice(s) de confusion et les métriques affichées à l'écran (précision, rappel et F1-Score par classe)** sur votre disque. Attention à bien organiser vos données par test !

### Deux articles référence dans le domaine à lire :

*Article2\_Blaschke\_2010\_OBIA.pdf* Object based image analysis for remote sensing et *Article3\_Blaschke\_etal\_2014.pdf* Geographic Object-Based Image Analysis – Towards a new paradigm

<https://enlive.live.unistra.fr/index.php/s/l6rfCMjH3fhFiMN>