I. But du projet

Le but de ce projet est de développé un algorithme pouvant être intégré à un satellite qui permettrait de détecter du plastique. Mon but était de faire avancer le projet, en particulier dans la détection de plastiques difficilement détectable (à l'échelle du sous pixel par exemple). J'ai ainsi pu lire des articles scientifiques, et j'ai appliqué les différentes méthodes décrites. Comme vous allez le constater par la suite, je me suis surtout focalisé sur des algorithmes de classification non supervisé. En effet, il y avait peu de donnée à ma disposition pour tester mes algorithmes, et il est donc difficile d'entrainé des model (pas assez de donnée/résultat non généralisable).

II. Préparation des données

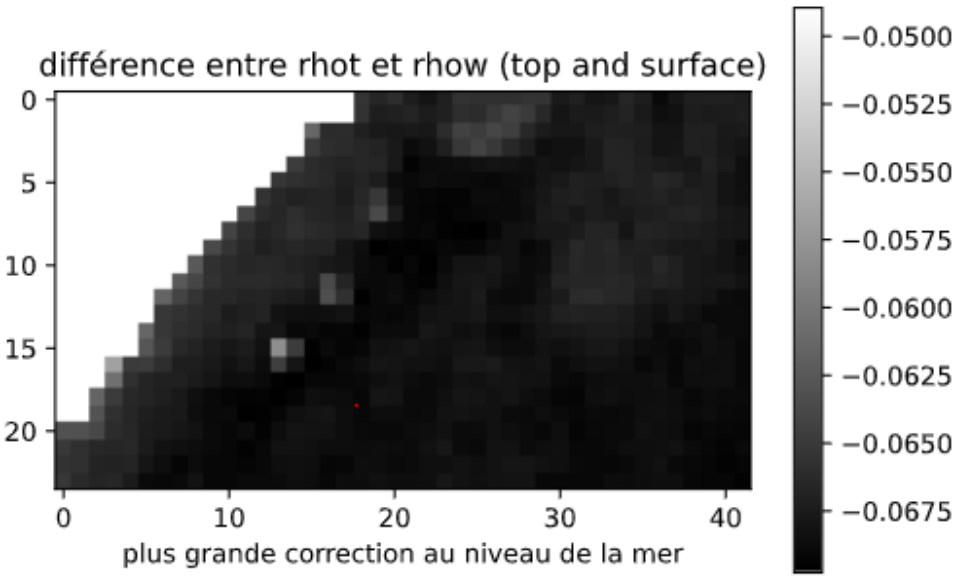
a. Présentation du dataset

J'ai choisi de travaillé sur un dataset appelé PLP 2019. Ce dataset correspond à des images dans lesquels des équipes de volontaires ont déployé des cibles de plastique lors du passage de satellite. Les cibles de plastiques ont été géolocalisées, ce qui nous permet de labéliser les différents pixels.

Dans ce Dataset, on a d'une part les informations du satellite Sentinel 2, d'autre part les informations liées au cibles plastiques (donnée avec une distance en m). Les fichiers contiennent beaucoup d'information, dont par exemple la distance en m de la prise de vue par rapport à un repère, ce qui va permettre de trouver les cibles dans les images satellites. Ensuite, le fichier netcdf présente des variables correspondantes à la réflectance avant traitement (rhot) et après traitement par un algorithme qui corrige les effets de l'atmosphère (rhos). De plus, en fixant un seuil de luminosité de 0.1, on est capable d'enlever de l'image les pixels correspondant à la terre.

B. influence de la correction atmosphérique

Dans le cadre de mon projet, j'ai choisi de travailler sur la dernière variable, rhow, est celle sur laquelle j'ai travaillé dans tout le sujet. Ce choix s'explique par le fait qu'après correction, les données correspondant au plastique ressorte plus qu'avant, car la correction est plus faible pour ces pixels.



La correction utilisée s'appuie sur la méthode dark_spectrum du logiciel ACOLITE (voir documentation : Cette algorithme est basée sur l'hypothèse que certaines bandes de fréquence vont avoir une fréquence nulle. L'algorithme est capable de sélectionner la bande de fréquence la plus adapté, et de corriger les différentes données. Cet algorithme à été conçu pour fonctionner sur des images contenant de l'eau turbide (donc prés des cotes), et est donc indiqué pour le Dataset PLP2019. Je n'ai pas étudié l'algorithme le plus adapté pour des eaux calme, mais on peut aussi utiliser cette algorithme.

Comme l'algorithme se base sur des données hyperspectral, on peut s'attendre à une baisse de performance pour notre application. Je n'ai pas évaluer cette baisse de performance, qui devra être étudié lorsque des algorithmes auront été développé.

Le notebook correspondant à cette première étude est : exploration netcdf.ipynb

C. manipulation des données

Lors du travail précédent réalisé dans le cadre du projet 3A à l'Imt atlantique (voir le github [2]), la structure de donnée était une liste de tableau numpy, ou chaque tableau numpy représentait une image hyperspectral. J'ai choisi de changer de structure de donnée car : 1. Cette structure de donnée ne permettait pas de stoker les informations sur le nom des variables utilisé 2. Certaines opération n'était pas immédiate (ainsi, la plupart des algorithmes utilisé imposait le redimensionnement d'une array pour donner aux algorithme des données unidimensionnel).

J'ai donc choisi d'utiliser une combinaison de pandas et de numpy, en utilisant pandas pour le stockage et la manipulation des données les plus simple (lorsque l'on peut travailler pixel par pixel), et utilisation de numpy lorsqu'il était nécessaire de prendre en compte la dimension spatial (affichage, comparaison des pixels avec les pixels voisins. J'ai choisi d'utiliser la librairie pandas en utilisant la structure de donnée Dataframe avec des index à 2 dimensions : chaque pixel avait pour indice (acquisition photo, n°pixel). Les attributs de ce Dataframe les bande spectrale était rassemblé dans une même bande (559 nm et 660 nm dans la bande B03 par exemple, ainsi que le label de la photo, et la distance aux cote (obtenue par distance euclidienne au pixel identifié comme faisant partie de la cote le plus proche). Cette approche à pour avantage qu'il était facile d'effectuer des statistiques sur les différentes photos, et que les opérations simples se faisait en quelques lignes. En revanche, l'affichage des données était plus complexe (nécessitait de repasser par la librairie numpy). Ce choix m'a permis d'améliorer mes compétences en traitement de donnée, mais une utilisation de la librairie Xarray aurais surement été plus judicieux et aurais mené à des algorithmes plus clairs.

D. notebook :

J'ai utilisé plusieurs notebooks. Le premier notebook, exploration Netcdf correspond à ce qui est décrit dans cette première partie. Le deuxième notebook et le troisième notebook, exploration des données, et reprise des travaux précédents, correspondent aux notebooks du même nom. Le notebook evaluation_result présentent plusieurs algorithmes développés. Les notebooks restants servent à explorer les différents algorithmes.

III. Reprise des travaux précédents :

J'ai créé un notebook ou j'ai repris différentes approches effectuées précédemment dans le cadre du projet. J'ai tout d'abords mis les données dans un dataset comme expliqué dans la partie précédente, puis j'ai appliqué les différentes approches en m'aidant si besoin du code effectué précédemment. Je n'ait pas repris toute les approches, mais uniquement les approche basée sur les couleurs des pixels (classification par K-Mean, spectral clustering par exemple

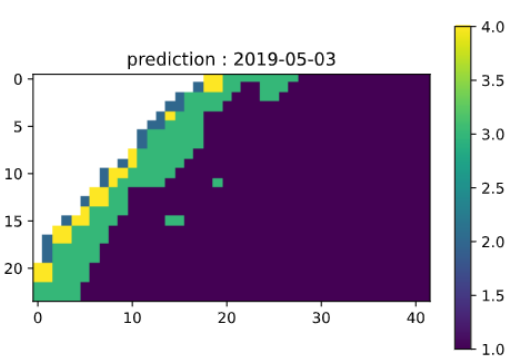
A. Algorithme K-MEAN

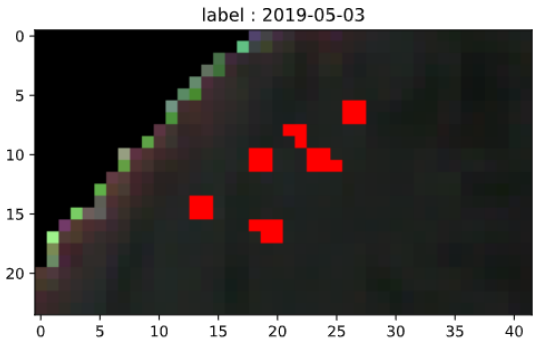
Pour l'utilisation de cet algorithme, j'ai choisi d'utiliser la bibliothèque sklearn qui implémentait l'algorithme. L'algorithme de k-mean est un algorithme qui est sensible aux conditions initiales, et qui peut tomber dans des minimums locaux. L'implémentation de sklearn présente l'avantage que le résultat de l'algorithme est le meilleur résultat obtenu pour divers initialisation avec un algorithme adapté. Précédemment, l'algorithme de K-Mean avait été appliqué sur une région plus grande contenant à la fois de la mer et de la terre, grâce à l'ensemble des bandes spectrale. Ce choix d'implémentation n'est pas forcément une bonne idée, cela pour deux raisons :

- Très grande variabilité des pixels correspondant à de la terre par rapport à ceux de la mer : on observait donc 1 cluster pour la mer, et plusieurs pour la terre.
- L'utilisation de toutes les bandes spectrales à pour conséquence qu'on s'expose à la malédiction de la dimension. En effet, plus le nombre de dimension d'entrée est élevée, plus le nombre de point devra être élevée pour estimer des régions. L'utilisation du k-mean est extrêmement sensible à ce facteur.

Pour obtenir des résultats, je me suis donc limité sur l'image aux pixels correspondant à l'eau, en utilisant uniquement les bandes déjà identifié comme étant les plus intéressante (B04,B06,B08,B011)

Résultats :



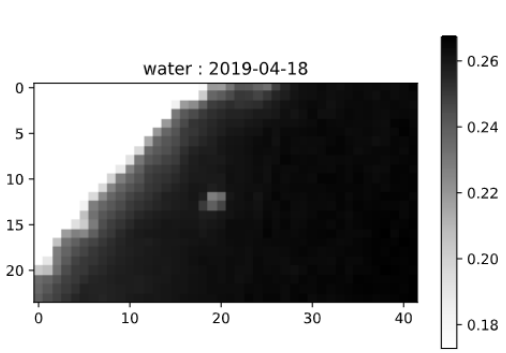


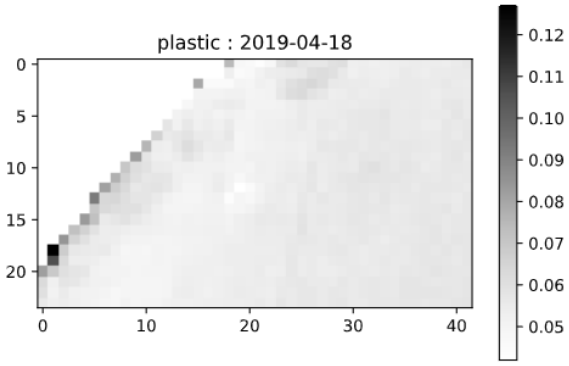
On remarque que l'on arrive à détecter grossièrement les plastiques en prenant le bon nombre de classe. Les cotes sont cependant détectées avec les plastiques, et les plastiques ne sont pas tous détecté. De plus, il faut identifier la classe correspondant au plastique, et le choix du nombre de classe est déterminant pour les résultats finaux.

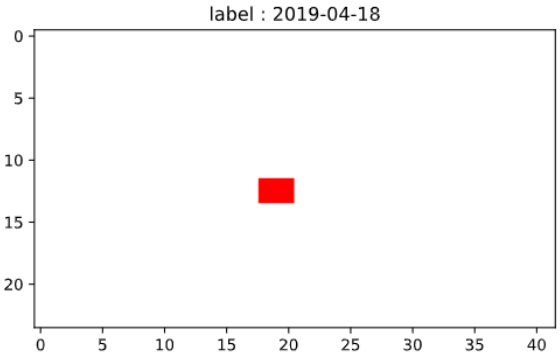
B. Algorithme basée sur l'exploitation d'une référence

Ces algorithmes sont basés sur la comparaison à partir d'élément de référence. Le premier algorithme est basé sur la distance euclidienne entre chaque pixel et des éléments de références.

Résultats :





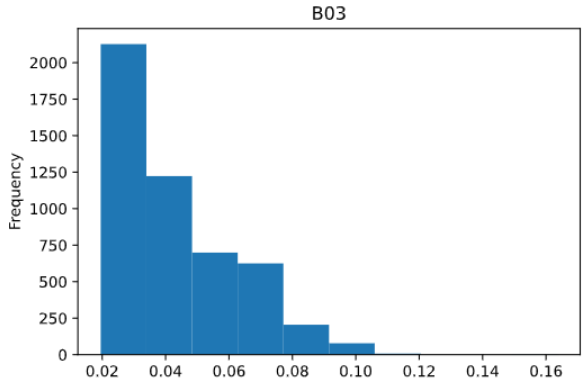


On observe que la distance entre les pixels et le plastique n'est pas forcément suffisant pour détecter le plastique. Cependant, la distance entre le plastique et l'eau présente des valeurs où l'on peut plus facilement détecter le plastique. Ce résultat viens surement du fait qu'il est plus facile d'avoir des informations sur la signature spectrale du plastique dans l'image que du plastique.

En plus de l'approche sur la distance, j'ai utilisé le partial spectral unmixing. Dans cette méthode, on essaye d'estimer la proportion de la valeur de référence dans chaque pixel (en considérant que le reste de l'image correspond à du bruit). Cette implémentation ne tenait pas en compte les différentes contraintes annexe, et avait pour but d'essayer d'implémenter un premier algorithme d'unmixing. Avec cette méthode, les résultats obtenus ne sont pas intéressants.

IV. Data exploration

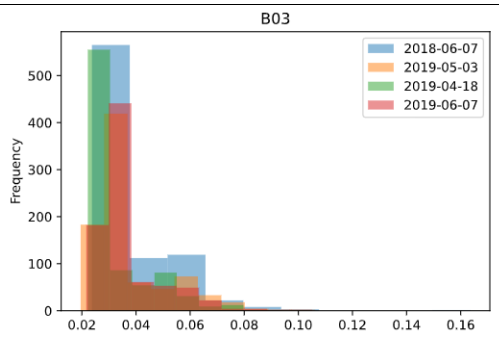
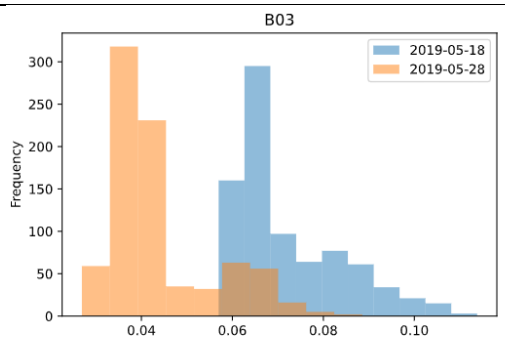
-1er résultat : grande variabilité des différents indices spectrales.



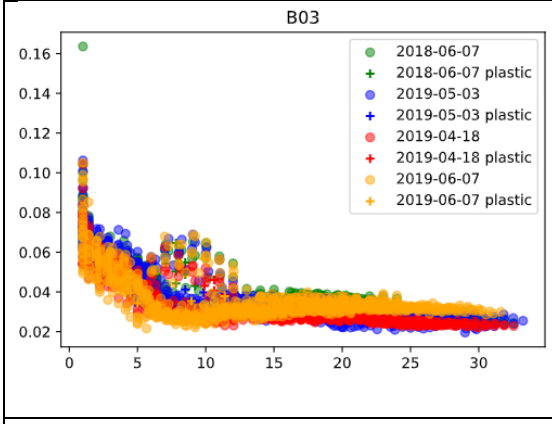
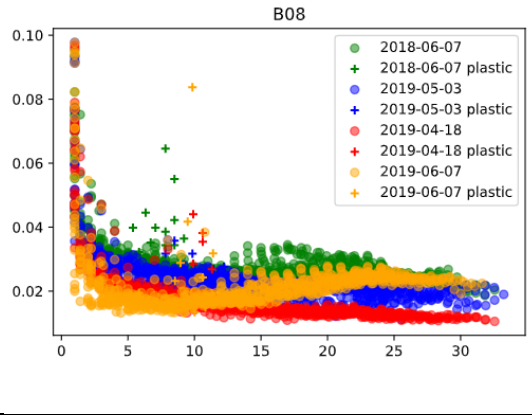
Histogramme des valeurs sur tout le dataset

Cette variabilité à deux causes : variabilité spatial (due à la petite profondeur proche des cotes) et changement entre les différentes images.

Variabilité entre images :

	
Histogramme des valeurs pour chaque image (temps calme)	Histogramme des valeurs pour des images (temps nuageux)

Variabilité (distance par rapport à la cote) :

	
Réflectance en bande B03 par rapport à la distance à la cote (en 10aine de m)	Réflectance en bande B08 par rapport à la distance à la cote (en 10aine de m)

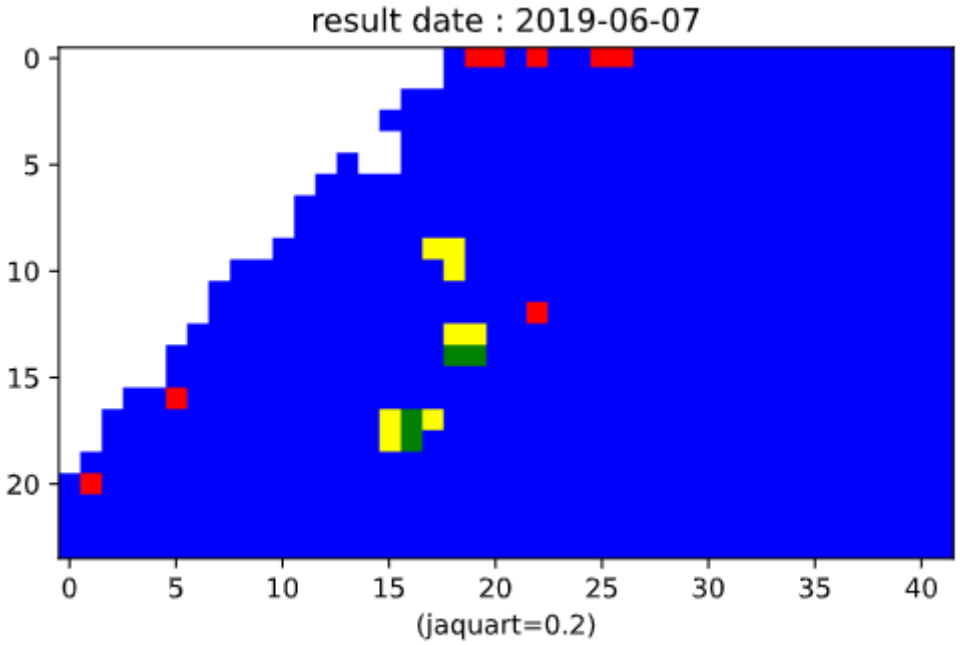
On remarque de plus que les pixels plastiques présentent des réflexivité supérieur dans les bandes correspondant à l'infrarouge.

Cette réflectance plus élevée dans la bande infrarouge inspire une idée d'algorithme basique :

Dans les pixels où il y a du plastique, la réflectance est plus élevée. Cela se traduit dans l'image par nombre plus élevée au niveau du laplacien dans l'image. Ainsi, le laplacien doit être plus (positif) autour des pixels ou il y a du plastique, et moins élevée (négatif) au niveau des pixels plastiques :

$$\frac{f(x+h,y)+f(x-h,y)+f(x,y+h)+f(x,y-h)-4f(x,y)}{h^2}$$ (avec h=1)

Pour obtenir une valeur de seuil variant en fonctions des caractéristiques des images, j'ai choisi comme valeur de seuille moins l'écart type. (Ainsi, seules les valeurs extrêmes sont sélectionnées)



Ce type d'approche ressemble beaucoup au test d'hypothèse (il est moins poussé).

Pour qu'un algorithme est des résultats intéressants, on doit avoir des résultats au moins similaires à cette algorithme ne se basant que sur une seule bande spectrale.

V. Spectral indices

Pour explorer les différents indices j'ai utilisé le logiciel rapidminer. Pour pouvoir reproduire mes résultats, il faut ouvrir avec rapidminer le processus PLP test indice (en copiant collant le processus dans le répertoire Repository de rapidminer), cliquer sur read CSV, puis import Configuration wizard sur la fenetre de droite, puis trouver l'excel PLP2019, cliquer sur next et choisi UTF-8 comme File encoding (colum separator : ,), next et cliquer sur le triangle a coté de n et choisir type -> polynomial, puis label -> binominial. Cliquer ensuite sur finish.

Ensuite, pour lancer le processus, appuyer sur f11. On peut ensuite voir le résultat sur la colonne de gauche (chaque indices de chaque images). Différents types de visualisation sont alors disponible dans la colonne Visualisation. Les visualisation intéressante sont par exemple scatter, Parralel coordinates, scatter matrix).

Dans un premier temps, j'ai implémenté l'indice décrit dans l'article
https://doi.org/10.3390/rs12162648

Dans cette article, cette indice est définit par la différence entre la bande B08 et son interpolation entre la bande B06 et B11. La formule donnée dans l'article présente de plus un coefficient qui n'est pas expliqué pour définir l'indice interpolé. J'ai donc testé plusieurs valeurs de coefficients. Le

coefficient qui discriminait le mieux le plastique étant bien celui décrit dans l'article, j'ai choisi de l'utilisé dans la suite de mon travail.

Pour comparer mes résultats avec l'algorithme basique, j'ai choisi une valeur de seuil pour toute les images. Avec cette approche, on obtenait des résultats comparables avec l'algorithme basique. En explorant les données sur rapidminer, on constate que le seuil optimal change pour chaque image. Ainsi, en se basant sur l'indice moyen du fdi dans chaque image, il est possible d'obtenir des meilleures détections, en particulier lorsque le temps est nuageux par exemple. Je n'ai cependant pas implémenté un algorithme de ce type.

Bande nécessaire : B08 (10m), B06 (20 m), B11 (20m)

Résultat : Bonne détection des plastiques avec le bon seuil. On à cependant des mauvaises détections dans certains cas.

Avantage :

-cette méthode n'est pas basée sur la signature spectrale du plastique, et est donc plus générale que d'autres méthodes.

Inconvénient :

-Les bandes spectrales ne sont pas celles envisagées à la base.
-Les bandes spectrales ont une résolution plus faible (20m)

Cet indice peut être couplé avec un autre indice. Ainsi, dans l'article, les auteurs utilisent l'indice NDVI, dans le but de différencier le plastique avec les algues par exemple. Cette indice NDVI nécessite de plus la bande B04. Dans mes expériences, cet indice ne semblait pas bien discriminer le plastique. Cela est peut-être du au fait que les cibles plastiques était disposé au-dessus d'algues. J'ai donc essayé de regarder si d'autres indices pouvait mieux fonctionner. L'article https://dx.doi.org/10.3390/rs12162648 résume ainsi plusieurs indices, en introduisant une métrique pour évaluer la performance de chaque article. Il y a quelques erreurs dans la définition des indices dans l'article, il faut donc regarder chaque article cité pour avoir les bonnes valeurs des bandes. En implémentant les différents indices intéressants, on constate qu'il est difficile de trouver quels indices sont intéressant : en fonction des situations, les différents indices donnent des résultats différents. Des indices très similaires peuvent de plus paraitre très similaire à première vue, mais permettre beaucoup moins de détection de plastique en fonction des situations. En raison du nombre de donnée plastique limité, il est difficile de savoir quel indice serait optimales. Certains indices donnent des informations redondantes avec l'indice FDI, et des indices donnant des résultats moins bons pourrait être plus complémentaire avec cet indice que d'autres indices meilleurs.

L'étude de ces différents indices (quels indices utilisés, quelle seuil choisir en fonctions des photos) est donc un sujet qu'il faudra explorer par la suite.

VI. Détection from endmenbers

En implémentant le spectral unmixing, j'avais deux objectifs : 1. Voir l'influence des deux contraintes non implémentées précédemment dans le projet 3A (abondance >0 et somme =1), et essayer de voir quels endmenbers était nécessaire pour implémenter les algorithmes.

Pour implémenter les différents algorithmes, j'ai choisi d'utiliser la librairie pysptools. Cette librairie implémente plusieurs algorithmes de détection satellite, ce qui m'a permis de faire différents test.

La première approche à été d'utiliser uniquement un endmenber, correspondant au plastique. J'ai essayé les différentes approches disponibles dans pysptools. Parmi elle, l'approche CEM et matchedfilter semblait donner des résultats, mais moins bon que l'algorithme de base. L'endmenber utilisé peut-être la cible plastique, ou un endmenber correspondant à l'eau.

La deuxième approche a été d'utilisé plusieurs endmenber, en utilisant le spectral unmixing. Pour résoudre le problème d'unmixing, pysptools utilise une méthode d'optimisation non linéaire (least square optimisation). La détermination des endmenbers est alors essentiel pour la bonne détection du plastique.

Comme je me suis limité à quelques bandes (afin d'imiter les caractéristiques du satellite qui serait lancé dans le cadre du projet de l'IMSAT), un plus petit nombre d'endmenbers peuvent être utilisé.

Première approche : NNLS (Non-negative Constrained Least Squares)

Cet algorithme rajoute la contrainte que l'abondance est supérieure à 0, sans la contrainte que la somme des abondances est inférieur à 1.

Détection bonne avec 2 endmenbers (water +plastic). Mauvaise détection en général

-avantage : même si les endmenbers correspondant à l'eau sont de réflectance plus faible que dans la réalité, on arrive a avoir des résultat grâce à la forme des différents pixels.

-inconvénient : on doit connaitre exactement la forme du plastique. Sinon, la détection ne marche pas (plus mauvaise détection que l'algorithme naïve) La détection est de plus en général mauvaise car on prend pas en compte l'intensité

Deuxième approche : FCLS (Fully constrained least squares)

On rajoute la contrainte somme des abondances égale à 1 (plus abondance supérieur à 0).

En utilisant les bons endmenbers, les résultats sont assez bons. Le choix des endmenbers est cependant difficile. En changeant les endmenbers, et en incluant/excluant des certains endmenber, le résultat change parfois, sans qu'il soit évident de quantifier les performance de l'algorithme.

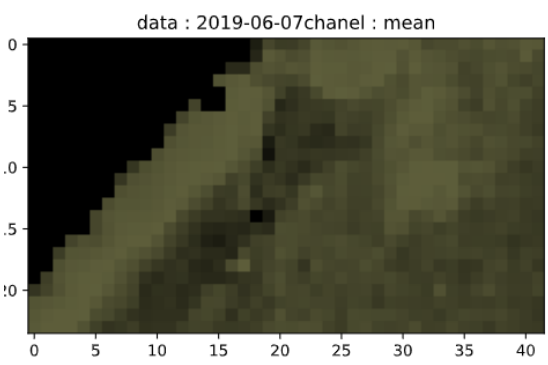
Dans un deuxième temps, en lisant [https://doi.org/10.1016/j.rse.2021.112414], j'ai compris qu'utiliser des bandes spectrales de résolution différente occasionnait une déformation dans le spectre théorique par rapport à un spectre de même résolution. J'ai ainsi dans le notebook spectral unmixing changé les bandes considérées. Grace à ces nouvelles bandes, la détection était améliorée.

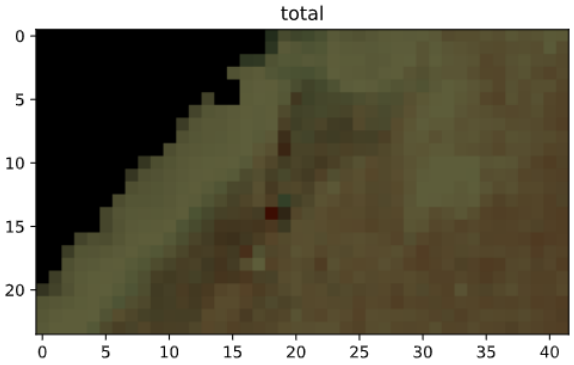
Pour pouvoir améliorer encore la détection par spectral unmixing, il faudrait développer un modèle pour prendre en compte ce point.

Pour pouvoir quantifier la performance des différents algorithmes, j'ai choisi d'utilisé un algorithme basée sur le laplacien sur la prédiction par le plastique, au lieu de de la bande B08.

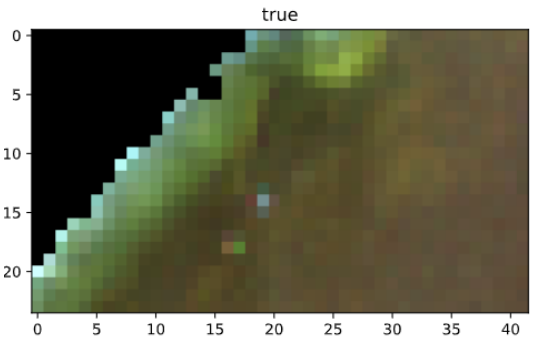
Une autre approche pourra être utilisé par un autre algorithme sur la différence entre les explications fournies par le spectral unmixing et la scène originelle :

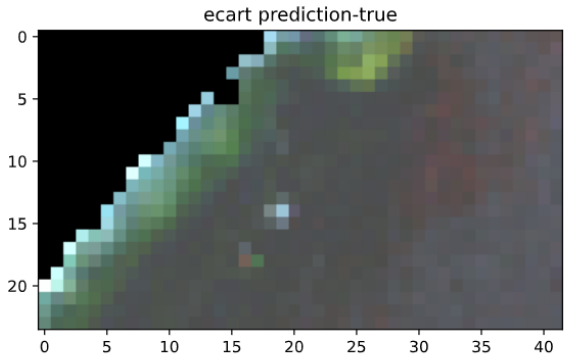
Résultat de l'unmixing (red B02, green B03, red B04) :





La majorité du changement est dans la classe mean, et rajouter les endmembers correspondant à l'eau permet d'affiner le résultat.





(Pour mieux visualiser les écarts, je les aie augmentés artificiellement).

On observe que dans l'ensemble, la modélisation permet d'expliquer une certaine variation.

Comme le spectral unmixing donnait de moins bon résultat que l'approche par indice, je pense qu'il faudrait mieux continuer de développer l'approche par indice. Cependant, pour les cas ou beaucoup de bande spectrale sont disponible (comme dans le cas des images Sentinel-2, cette approche peut être intéressante, en faisant attention de bien prendre en compte les différents points mentionnés ci-dessus.

VII. Autres approches (SAM, outliers)

A. Outliers (outlier detection.ipynb)

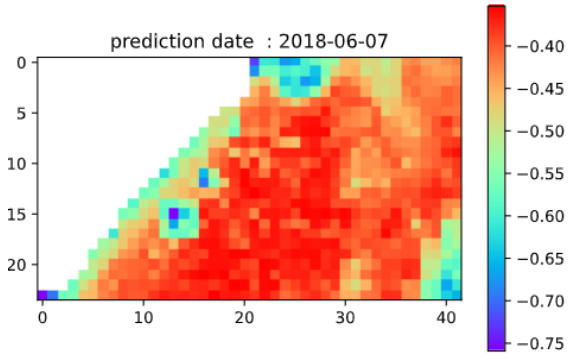
Cette méthode se base sur la couleur de chaque pixel. On souhaite alors décrire la distribution des pixels correspondant à l'eau, pour ensuite détecter les pixels ne faisant pas partie de cette classe (les plastiques). Dans ces méthodes, deux méthodes semblent intéressantes :

Isolation Forest :

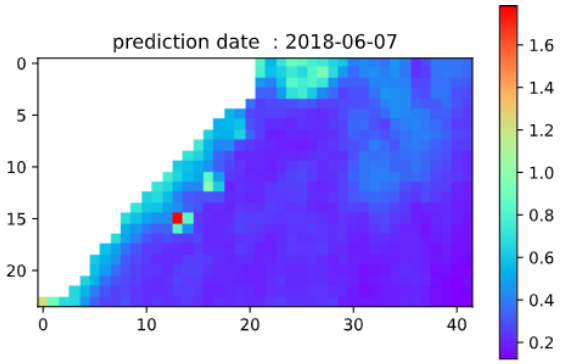
Totalement autonome, résultat intéressant mais on ne peut pas l'adapter pour avoir de meilleur résultat. La seule piste d'amélioration est alors d'augmenter le nombre de donnée.

SVM-oneclass : plus de paramètres à ajuster (type de kernel + paramètres du kernel). Même si la classification ne donne pour l'instant pas de bon résultat, les indices de séparation mettent en évidence que les résultats sont plus intuitifs.

L'avantage de ce type de méthode est qu'il n'y a pas de connaissance requise à prioris (contrairement à l'unmixing), et que l'on peut se baser sur les méthodes d'entrainement. On doit cependant avoir des données qui représente la signature spectrale de l'eau et qui ne ressemble pas aux pixels plastique pour espérer une détection.



Isolation forest



Svm-one class

Finalement, j'ai choisi d'abandonner ce type de méthode car 1. Je n'avais pas assez de data pour entrainer ce type d'algorithme, 2. Les détection était médiocre, 3. si dans l'image utilisé en entrainement, il y a des pixels ressemblant à du plastique, cela fausse l'entrainement de l'algorithme et le rend inefficace.

B. SAM (angle entre plusieurs signature spectrale)

Dans l'article [https://doi.org/10.1016/j.rse.2021.112414], les auteurs utilisaient le coefficient suivant pour discriminer les algues et le plastique :

$$\text{SAM (degrees)} = cos^{-1}\left[\left(\sum x_i y_i\right) \Big/ \left(\sqrt{\sum x_i^2}\sqrt{\sum y_i^2}\right)\right]$$

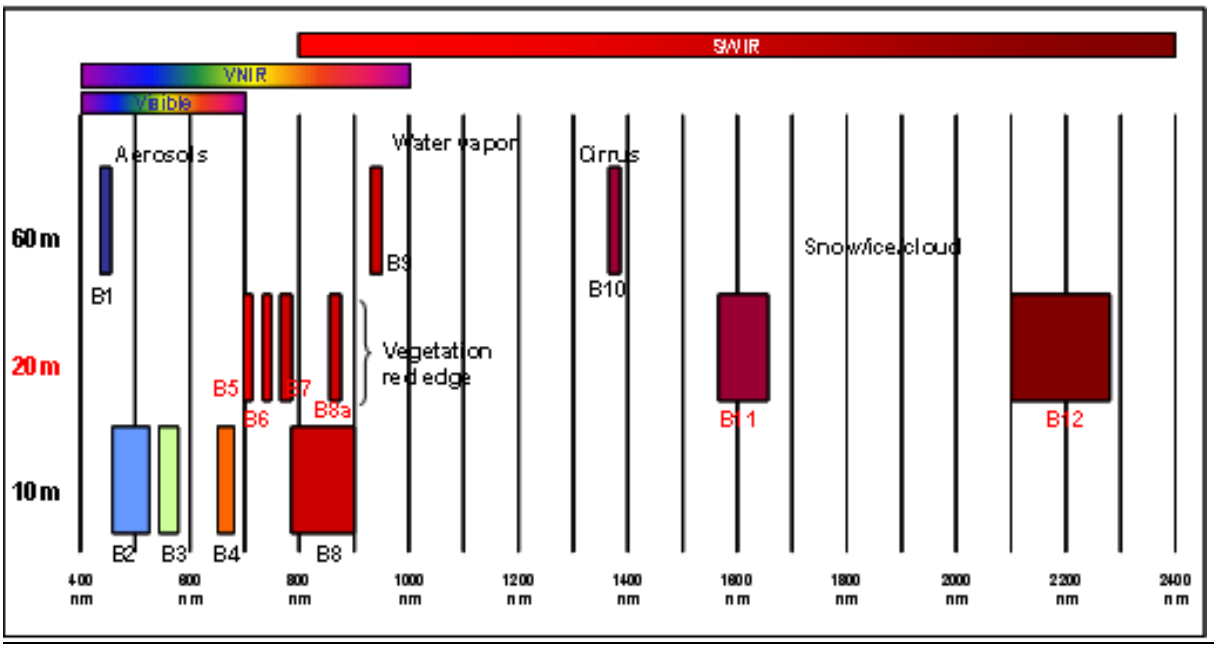
Où x représente la signature spectrale d'un pixel et y la signature spectrale d'un élément de référence. Ainsi, si un élément à un spectre similaire, on a un degré=0. Pour un spectre totalement différent, un degré=90. L'algorithme est alors appliqué non pas directement sur la signature spectral des plastiques, mais sur la différence de spectre entre un pixel et les pixels voisins.

En implémentant cette indice (pour B02, B03, B04, B08), j'ai remarqué que l'algorithme ne marchait pas bien : De nombreux pixels plastique était considéré comme très différent, avec les deux méthodes utilisées dans l'article (basée sur la réflectance, ou la différence de réflectance).

Dans l'article, l'image montrée était des images de pleine mer, contrairement au Dataset qui est près des côtes. Cette localisation explique peut-être les performances de l'algorithme : les variations sont trop élevées, et essayé de classifier les pixels en fonction de la forme du spectre de chacun des pixels est optimiste.

Le fait que cette méthode ne marche pas est cohérent : une des méthodes de la partie 5 correspond à cet indice.

VIII. Annexe



bandes spectrales observé (sentinel-2) et leurs résolution (spatiale et spectrale)

[1] https://odnature.naturalsciences.be/remsem/software-and-data/acolite

[2] https://github.com/BasileR/plastic_detection_from_space/tree/develop/data