Masque de Culture par les données d'observation de la Terre

Introduction

Dans ce rapport, nous allons décrire la méthode que nous avons utilisée pour la détermination du Masque-culture avec les données d'observation de la Terre et en ayant recours aux algorithmes de Machine Learning. dans cet exemple nous allons faire la démonstration sur une **tuile** de la méthode de classification à la fois du Deep Neural Network (**DNN**) et du Support Vector Machine Learning (**SVM**). Ce sont ces deux algorithmes qui sont retenus pour la production de masque de culture dans les zones intérêts. En effet, le DNN est produit localement pour le produire le masque culture et le SVM est produit dans le Cloud computing Plateforme, Google Earth Engine (**GEE**). Nous allons commencer par le SVM en utilisant la plateforme **GEE**. Nous allons définir la zone étude ici Trarza, les dates de début et de fin de la classification et la couverture nuageuse. Il est possible de changer ses paramètres suivant la zone. Ici une étude préliminaire de la couverture nuageuse sur une année pour la détermination de la période optimale de la classification. La période choisie ici à titre d'exemple, pour une prochaine utilisation, il est possible de le changer. Cependant le produit final sur toute la zone, nous avons choisi la période entre **2022-06-01** au **2022-09-30** Pour les deux de type de classification la base données est divisée en deux lots, un pour l'apprentissage du modèle et l'autre pour la validation. Tous les résultats, les codes, les figures et les données sont accessibles sur le lien suivant : https://github.com/mmbaye/crop-Mask-Arc

Rappel théorique sur les principales méthodes de machine Learning utilisées :

Support Vector Machine Learning

L'algorithme de support Vector machine (SVM) est un algorithme d'apprentissage supervisé qui peut être utilisé pour les tâches de classification et de régression. Dans le contexte de la classification, les SVM essaient de trouver la meilleure frontière de décision (ou hyperplan) qui peut séparer les différentes classes dans les données. Dans le cas de la régression, les SVM essaient de trouver la meilleure ligne de régression ou hyperplan qui peut prédire la variable de sortie continue en fonction des variables d'entrée.

Pour trouver la frontière de décision optimale, les SVM utilisent un processus appelé "maximal margin classification". Dans ce processus, les SVM cherchent à trouver l'hyperplan qui sépare les classes tout en maximisant la marge (ou l'espace) entre les différentes classes. L'objectif est de trouver un hyperplan qui sépare les classes de manière optimale en minimisant l'erreur de classification. Une fois que l'hyperplan optimal est trouvé, les SVM peuvent utiliser ce modèle pour prédire la classe d'une nouvelle observation en fonction de ses caractéristiques. Les SVM

peuvent également être utilisés pour résoudre des problèmes non linéaires en utilisant des techniques de transformation des données pour les projeter dans un espace à plusieurs dimensions où ils deviennent linéairement séparables.

La fonction de base radiale (RBF) est une fonction couramment utilisée dans le cadre du Support Vector Machine (SVM) pour résoudre des problèmes de classification ou de régression. La forme générale de la fonction RBF est donnée par :

$$f(x) = exp(-gamma * (x - x_i)^2)$$

Où x représente l'observation à classer, x_i représente une observation de l'ensemble de données d'entraînement, gamma est un paramètre de réglage du modèle SVM et $(x - x_i)^2$ est la distance Euclidienne. La fonction RBF est utilisée dans le cadre du SVM pour transformer les observations en un espace de plus grande dimension afin de séparer les différentes classes dans cet espace transformé. Cela permet au modèle de prendre en compte les observations non-séparables linéairement dans l'espace d'origine.

Notez qu'il existe un paramètre de paramétrage **C**, également appelé coût, qui détermine les erreurs de classification possibles. Il impose essentiellement une pénalité au modèle en cas d'erreur : plus la valeur de **C** est élevée, moins l'algorithme SVM est susceptible de faire une classification erronée d'une classe.

Deep Learning

La classification multi classes en Deep Learning est un domaine de l'apprentissage automatique qui vise à entraîner un modèle pour prédire l'appartenance d'un échantillon à une des n classes possibles. Contrairement à la classification binaire, qui ne permet de prédire qu'entre deux classes, la classification multi classes permet de prédire l'appartenance à une classe parmi un nombre plus élevé de classes.

Pour entraîner un modèle de classification multi classes en Deep Learning, on utilise généralement des réseaux de neurones profonds (**Deep Neural Networks, DNN**). Ces réseaux sont composés de plusieurs couches de neurones qui travaillent ensemble pour effectuer la tâche de classification. Chaque couche effectue une transformation sur les données en entrée en utilisant des poids et des biais, qui sont appris par l'algorithme d'entraînement. Il existe plusieurs approches pour entraîner un modèle de classification multi classes en Deep Learning. L'une des approches les plus courantes est celle du "one-vs-all" (un-contre-tous), dans laquelle on entraîne un modèle séparé pour chaque classe et on utilise ensuite ces modèles pour prédire l'appartenance à une classe en fonction des scores obtenus. Une autre approche consiste à utiliser un modèle unique en utilisant des sorties "chaudes" (ou "one-shot"), dans lequel chaque sortie correspond à une classe possible et le modèle prédit l'appartenance à une classe en sélectionnant la sortie avec la valeur maximale. La dernière couche de sorti donne les probabilités de chaque classe. En général, la classification multi classes en Deep Learning peut être très efficace pour résoudre des tâches de classification complexes comme l'occupation du sol issue des images satellitaires ou drone, en particulier lorsque les données sont volumineuses et ont un

grand nombre de classes. Cependant, il est important de disposer d'un jeu de données d'entraînement suffisamment important et de qualité pour obtenir un modèle précis. De plus, l'entraînement d'un modèle de Deep Learning peut être fastidieux et nécessiter des ressources informatiques importantes.

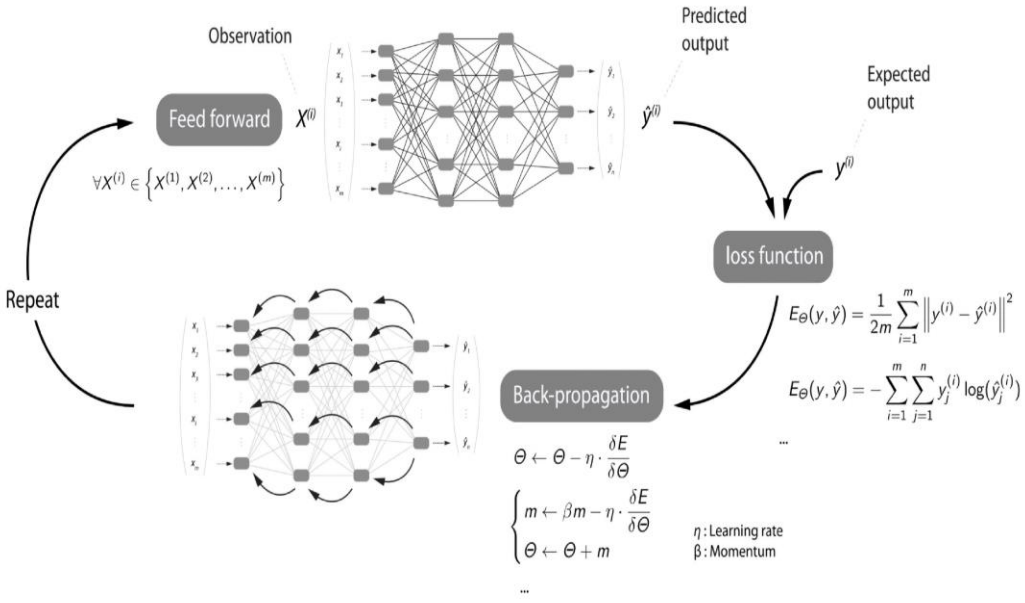

Fig. 1 : schéma d'apprentissage itératif des DNN pour la classification. Les couches sont des fonctions mathématiques de transformation des variables observées en l'entrée de façon non-linéaire. Ce réseau consiste apprendre à faire correspondre les données observées en leur classes respectives. Ce réseau optimise la fonction de perte (loss function) et de façon rétroactive en allant vers une minimisation des erreurs et une augmentation de la précision.

Méthodologie

Cloud Computing Plateforme

Pour la détermination du masque de culture dans la zone d'étude, la plateforme de « *cloud computing* » appelée *Google Earth Engine (*GEE) est utilisé. GEE est une plateforme informatique, notamment un Supercalculateurs pour le calcul de haute performance et un Datacenter pour le stockage des données. GEE consiste en un catalogue de données de plusieurs pétaoctets prêtes à être analysées. Grâce à ce catalogue de données continuellement mis à jour, les utilisateurs ont accès à un large éventail de données géospatiales provenant de satellites d'observation de la terre et de systèmes d'imagerie aérienne dans des longueurs d'onde optiques et non optiques, de variables environnementales, de prévisions météorologiques et climatiques, et d'ensembles de données rétrospectives, de couverture végétale, de topographie et de données socio-économiques. GEE fournit aussi des algorithmes innovants de traitement des données

Les données Satellitaires

Sentinel 2

Les données (open-source) optiques multispectrales Sentinel-2 de l'Agence Spatiale Européenne (ESA), sont utilisées dans ce travail. Elles sont acquises par deux satellites jumeaux (Sentinel-2A et Sentinel-2B), lancés séparément en orbite polaire synchrone à une altitude de 786 km et évoluant à 180° l'un de l'autre. Chaque satellite est équipé d'un capteur d'imagerie multispectrale (MSI) comprenant 13 bandes spectrales (de 443 à 2 190 nm) avec un champ de vue de 290 km et une résolution spatiale de 10 m pour quatre bandes dans les domaines du visible (VIS) et du proche infrarouge (PIR), de 20 m pour six bandes dans le domaine infrarouge proche et à courte longueur d'onde, NIR et SWIR), et de 60 m pour trois bandes de correction atmosphérique. Les caractéristiques des bandes principales sont présentées dans **le tableau 1**. L'objectif principal de ces données satellitaires est de surveiller la variabilité de la surface de la Terre, y compris les changements de végétation au fil des saisons, avec un temps de retour élevé (10 jours à l'équateur avec un satellite et 5 jours avec la constellation). **La Figure 2** montre la localisation de la zone d'étude qui se situe à cheval entre la Mauritanie (nord du Sénégal et le sud de la Mauritanie) et le Mali (Sud-Est du Sénégal et l'Ouest du Mali) et couvrant une superficie d'environ **1 034 184 km².**

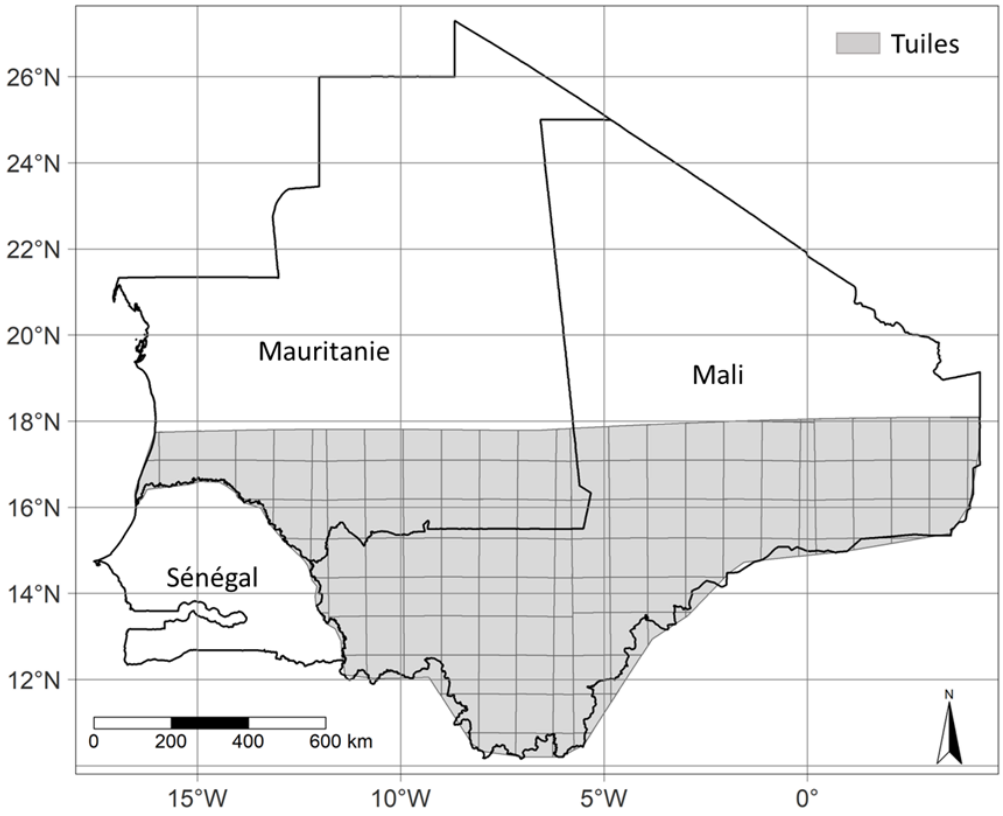

Fig. 2 : Carte de localisation de la zone d'étude (en gris) couvrant la Mauritanie et le Mali

Le tableau 1 donne quelques indications sur les bandes optique du satellite Sentinel-2.

Bandes	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11
Domaine	Bleu	Bleu	Vert	Rouge	Rededge- 1	Rededge-2	Rededge-3	PIR	PIR	MIR-1	MIR-2
Résolution (m)	60	10	10	10	20	20	20	10	20	20	20
longueur d'onde (nm)	443	490	560	665 n	705	740	783	842	945	1375	1610

Tableau 1 : Caractéristiques spectrales et spatiale des données Sentinel-2. PIR = Proche Infrarouge, MIR = Moyen infrarouge. Red edge

En considérant les données satellitaire MSI, une série d'indices ont été calculés. Dans un premier temps tous ces indices sont soumis aux algorithmes de machine Learning. Dans un second cas, une sélection de certains indices qui ne montrent pas de corrélation ont été enlevés et la troisième option consiste à classifier uniquement avec les bandes pertinentes ce qui assure une rapidité de calcul. L'objectif de cette utilisation multiple de bandes spectrales et d'indices est de définir le ou les indices et la ou les bandes qui sont finalement les plus utiles pour classifier avec le minimum de confusion avec les autres types d'utilisation du sol. Les équations utilisées pour générer ces indices sont présentées dans le tableau 2. Ces différents indices sont sensibles à la végétation (NDVI), à l'humidité du sol (NDWIa, NDWIb), à la brillance du matériel de surface (BIa, BIb), à la couleur du matériel de surface (IC, RI), ou à la présence de sol nu et de zones bâties (NDBI, BSI, NBAI, BRBA).

Nom de l'indice	**Formules**
Normalized Difference Vegetation Index (NDVI)	$\frac{B8 - B4}{B8 + B4}$
Normalized Difference Water Index (NDWIa)	$\frac{B8 - B11}{B8 + B11}$
Normalized Difference Water Index (NDWIb)	$\frac{B3 - B8}{B3 + B8}$
Brightness Index (BIa)	$\sqrt{B4^2 + B3^2}$
Brightness Index (BIb)	$\sqrt{B4^2 + B3^2 + B8^2}$
Color Index (CI)	$\frac{B4 - B3}{B4 + B3}$
Redness Index (RI)	$\frac{B4^2}{B3^2}$
Normalized Difference Built-up Index (NDBI)	$\frac{B11 - B8}{B11 + B8}$
Bare Soil Index (BSI)	$\frac{(B11 + B4) - (B8 + B2)}{(B11 + B4) + (B8 + B2)}$
Normalized Built-up Area Index (NBAI)	$\frac{(B12 - B8)/(B2)}{(B12 + B8)/(B2)}$
Band Ration for Built-up Area (BRBA)	$B3/B8$

Tableau 2 : Indices spectraux, leurs acronymes et les expressions mathématiques utilisées pour les calculer à partir de séries temporelles multispectrales Sentinel 2.

Les Données de Terrain

Les données de terrain ne concernent que la partie Malienne de la zone d'étude. Ces données de terrain sont collectées entre la période du 2022-09-23 et du 2022-10-23. En total, 1734 points GPS ont été acquises. Une étude de la qualité des ces données de terrain a été faite car le travail de terrain est très susceptible à des erreurs surtout en cas de perte du signal du GPS qui donnerait une mauvaise position. Ce travail d'assurance qualité a été fait en utilisant le package plotKML sous R et avec le logiciel Google Map pro qui a permis de faire du rééchantillonnage et de considérer des polygones en lieu et place aux points GPS. Concernant la Mauritanie, un échantillonnage de la classe agriculture a été générée par photo interprétation. Des polygones des ces classes agricultures ont été générés et essayant de les représenter de façon aléatoire dans la limite géographique de la Mauritanie en utilisant à la fois Google map Pro and le langage de programmation R, spécialement le package sf. La figure 4 montre la localisation et la distribution des points GPS obtenus lors de la mission de terrain au Mali. Ces points sont en grande partie localisés des routes principales. Les images à droite montrent les aspects de control des echantillons à exclure.

La figure 3 indique certains types d'occupation de sol entre autres la classe 'Build-up', les surface d'eau, le sol-nu et la classe agriculture. Apart les plans d'eau, les couvertures ne sont pas homogènes car comme indiqué ci-dessus pour les classes sol et habitat.

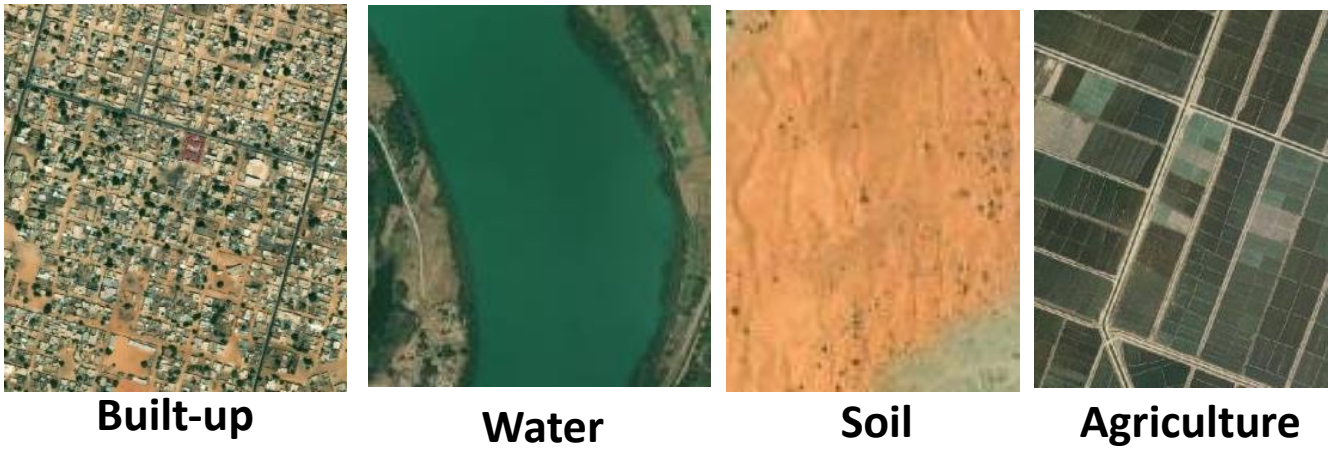

Fig. 3 : Illustration de quelques occupations de sol qui sont considérées dans cette étude

Pour la première recommandation, et dans le souci de reproduire ce masque de culture, il faudra veiller à une méthode plus précise d'échantillonnage sur terrain, ce qui peut éviter ce long travail de control des points GPS. Il est aussi conseiller de prendre des contours des champs et pas des points GPS.

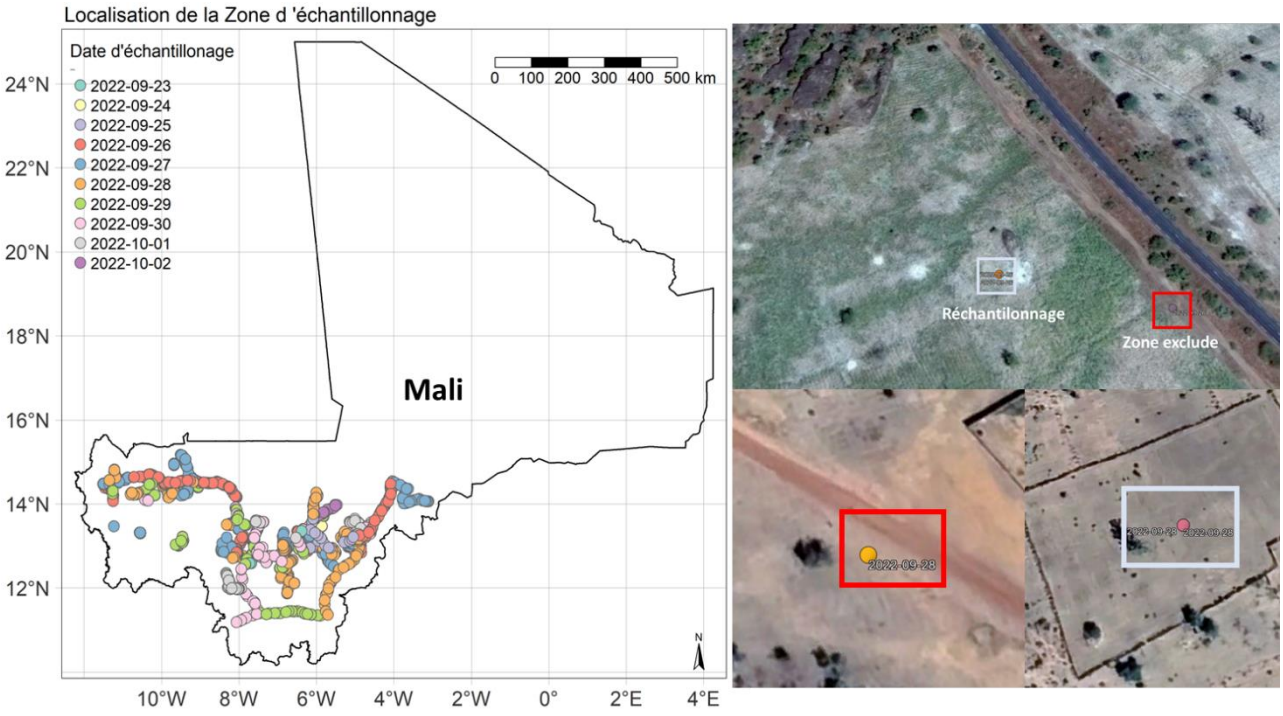

Fig. 4 : Distribution et localisation des données de terrain de la classe culture lors de la mission du Mali et les photos satellitaires pour le diagnostic de la pertinence de ces données de terrain.

Workflow

La Figure 5 ci-dessous décrit le pipeline utilisé pour la génération de masque de culture. La première étape consiste à la création de données d'apprentissage. Comme les données de terrain disponible ne sont que de la classe culture, il a fallu chercher les autres classes d'occupation du sol, notamment la vegetation aquatique, le sol-nu, les plan d'eau, les arbres etc. Pour cela, deux techniques ont été utilisées. La première consiste à utiliser Google map Pro et de localiser les autres classes d'occupation du sol. La seconde méthode consiste à extraire dans les bases de données certaines classes d'occupation du sol dans les zones d'intérêts. L'objectif de ces deux techniques était de permettre de mieux discriminer les autres occupations du sol par rapport a la classe culture. Car il était noté une confusion très importante de classe végétation herbacée et arbre aux zones de culture. Dans la figure 5, la première (1 et1') partie consiste à joindre les deux données d'apprentissage, ces données ont été évaluées sur deux ans entre 2021 et 2022. Ensuite la deuxième partie considère les données Sentinel-2 auxquelles un pré-traitement a été effectué comme le filtrage des images avec une couverture nuageuse de moins de 5%, la sélection des bandes de reflectance d'intérêt et le calcul des indices de vegetation comme décrit plus haut. La

troisième étape permet de subdiviser les données en deux lots : un pour le lot d'apprentissage et un autre lot pour la validation du modèle. Ces données d'apprentissages sont soumise a deux types de classifications a savoir le Support Vector Machine Learning et le Deep Learning. En cinq, après la validation en choisissant le modèle avec la meilleure précision basée sur la matrice de confusion , le pipeline donne les classes et on extrait le masque culture en format raster et shapefile. La conversion en format shapefile de certaines classes comme le sol nu est pas trop efficace.

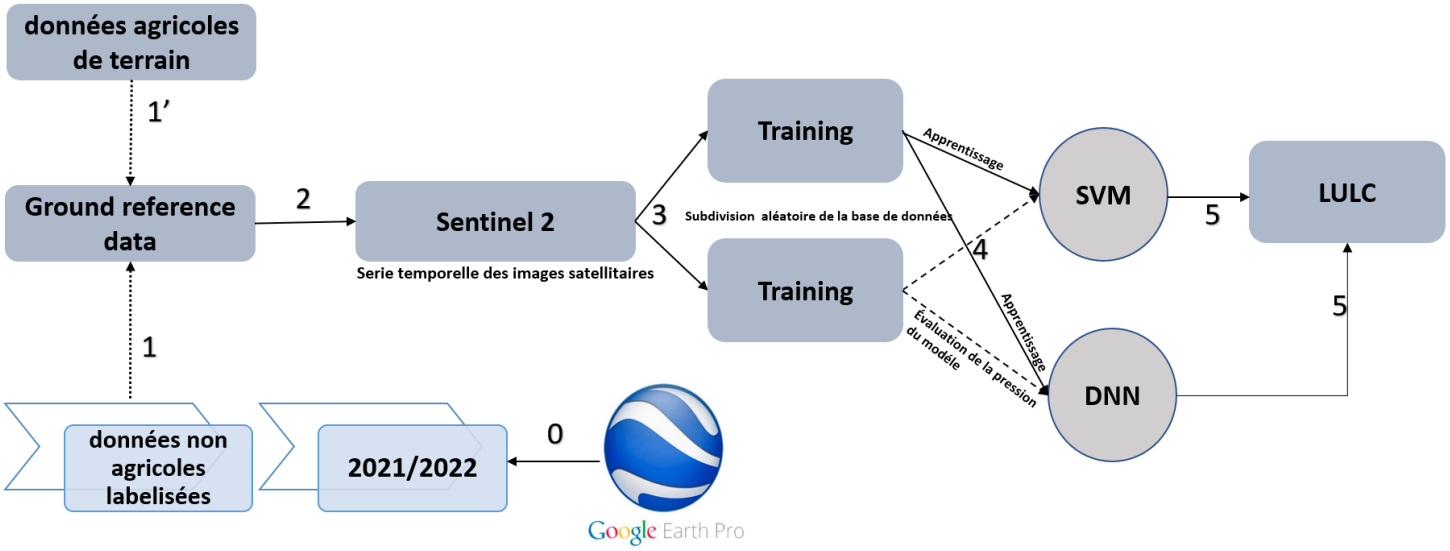

Fig. 5 : Schéma décrivant la méthodologie de la classification pour la production du masque de culture

Résultats

Support Vector Machine Learning SVM

Dans cette section, une présentation des résultats obtenus sont exposée. La première partie concerne la partie descriptive des données d'apprentissage. Ce travail a été fait sous GEE et R. Pour réduire le temps de calcul, il est recommandé d'explorer les distributions des valeurs des indices et des bandes afin de pouvoir discriminer les meilleurs bandes et indices pour la classification. Malheureusement GEE ne donne pas une façon efficiente de tout visualiser en même temps comme R ou Python. Tours les codes et certains résultats sont contenus dans une page GitHub pour la reproductibilité des résultats. La figure 6 montre les relations entre les classes d'occupation du sol en fonction des variables prédicteurs des modèles. On note une certaine cohérence de ces distribution, notamment la relation entre les bandes B11 le MIR et le B3 le vert qui montre une forte reflectance des classes habitats et végétation aquatiques et une faible reflectance de l'eau due a la forte absorbance de l'eau dans ce domaine spectrale. De la même façon une relation linéaire des classes dans le bleu et le vert.

On note que en somme, les distributions sont non-linéaires et les classes eau , sol nu et habits sont très discriminants.

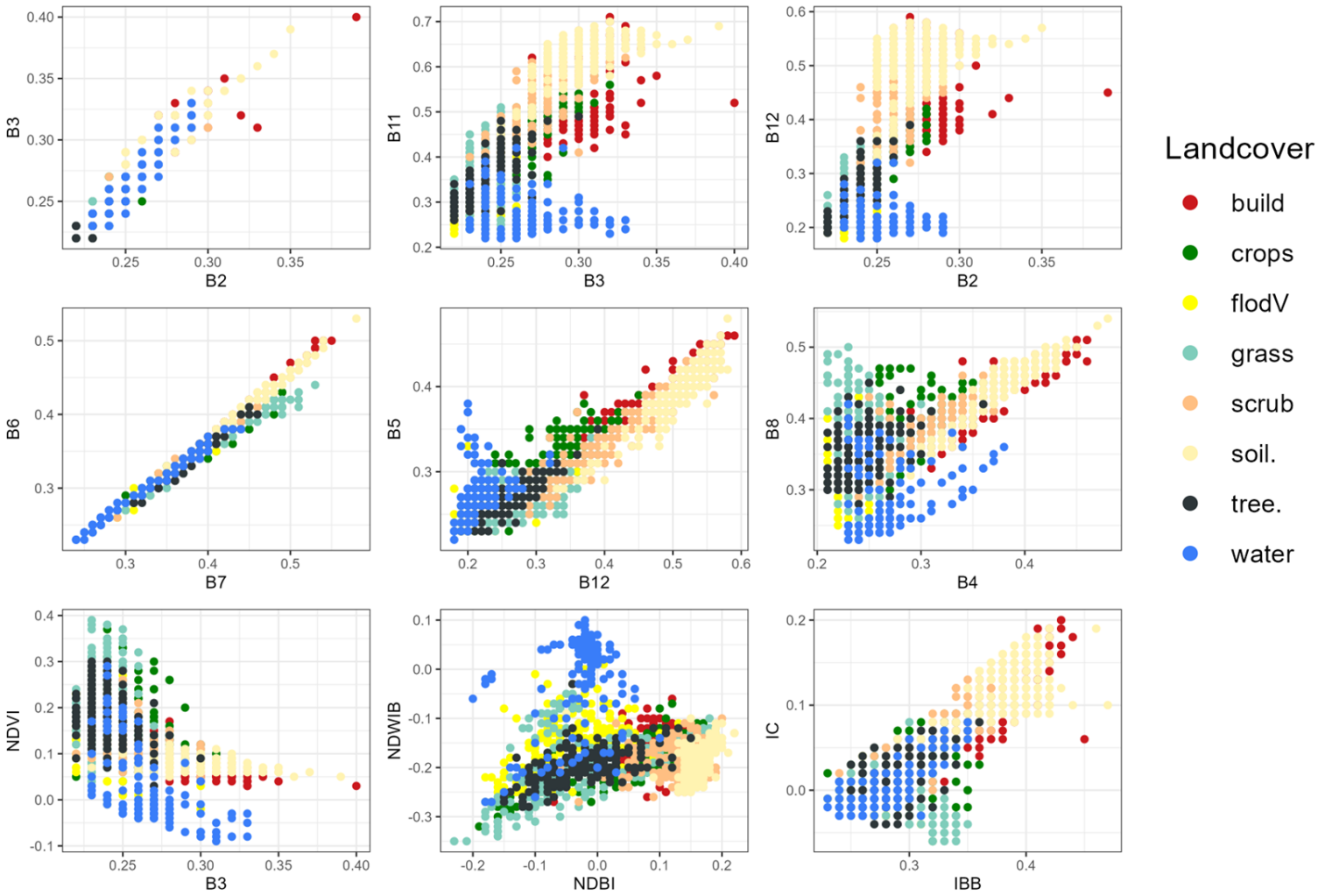

Fig. 6 : distribution des values d'entrainement du modèle de SVM sous google Earth engine. Ces données ont été exportées pour une meilleure visualisation.

La figure 7 montre un exemple de classification SVM de type non-linéaire avec des paramètre gamma de 0.5 de cost de 10 et un kernel Radial. Vue la grandeur de la région a classifier, nous avons choisi, d'exporter les images sous forme de raster par classes et de les vectoriser aussi par classes et par zone pour optimiser l'exportation. Ces données d'apprentissage sont partagées dans un fichier google drive que vous pouvez utiliser pour vérifier le code et les résultats. Rappeler que pour les données partagées nous avons exporté que les classes habitats, culture et eau. S'il y a un besoin, il est possible d'exporter les autres classes sous forme de raster. Néanmoins cela prend du temps vue la taille de la zone. Les codes est aussi partagé sous format texte, dans le répertoire google drive du projet.

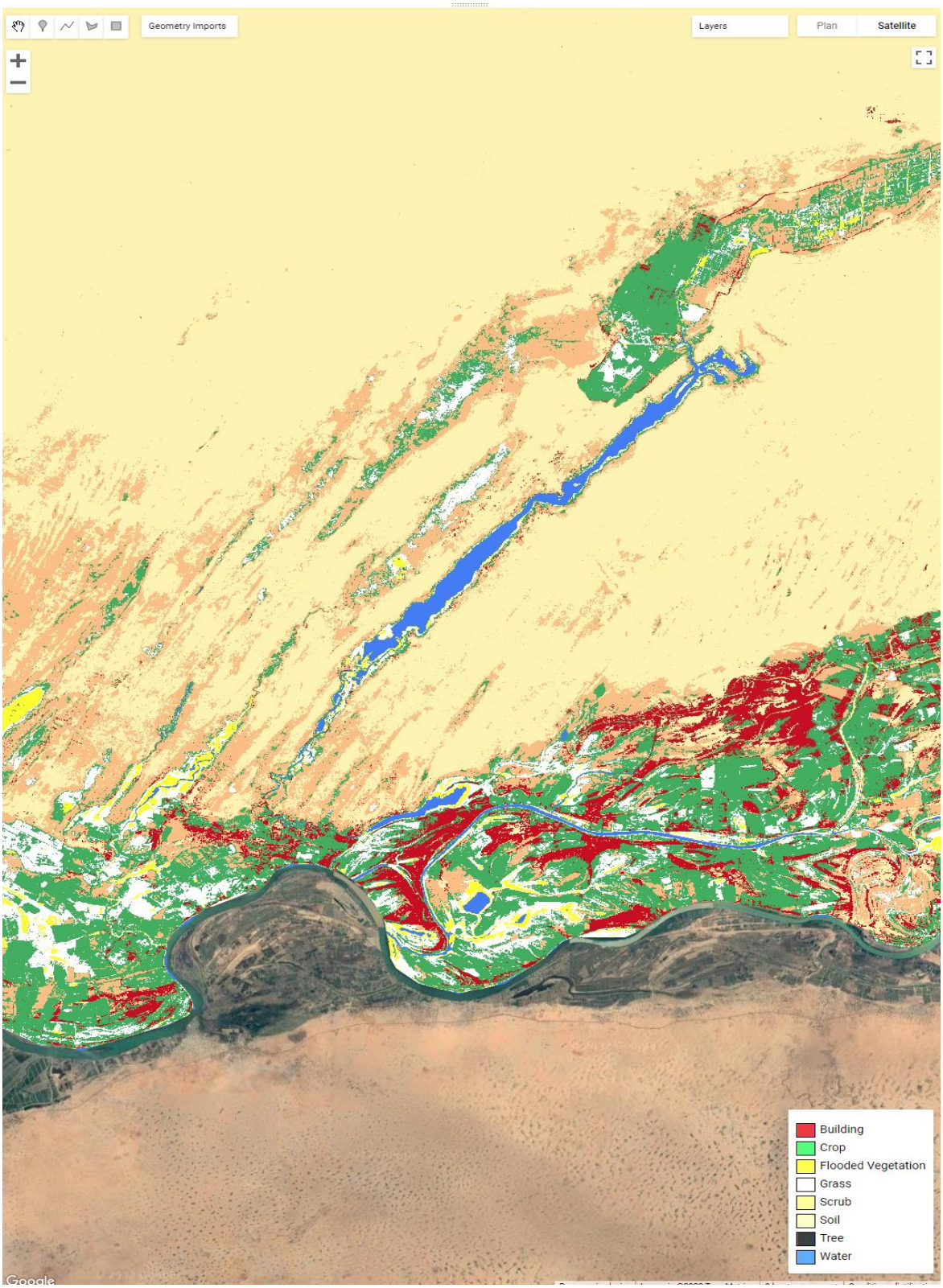

Fig.7 : sortie d'une classification SVM

Deep Learning

Il est assez connu que les réseaux neurone sont plus robustes pour la classification supervisée. En plus, il est relativement plus facile de faire des prédictions en utilisant les sortis du modèle. Ici, nous avons utilisé R, Cependant il est aussi possible de faire sous Python en utilisant google Colab. Car le training de ce jeu de données en local avec une carte GPU 32 NVIDIA GeForce GTX 1650 Ti a pris 7H pour la première fois et environ 9h pour le second apprentissage.

La première partie consiste à importer les données et les package avant de construire les couches. À noter que nous avons deux bases de données, une sous forme de shapefile de la localisation des types de cultures l'autre les données extraites des bandes et indices de l'image moyenne mensuelle de sentinel 2. Cette technique est aussi applicable avec les données radars de Sentinel 1. La deuxième partie est la préparation des données, et les mettre dans le format requis pour pouvoir faire le DNN. Il existe plusieurs façons de préparation de ces données, ici nous choisissons la façon la plus simple et consiste de faire. Le tableau 3 suivant décrit les classes utilisées en extrayant des parcelles des classes sous GEE et en utilisant le package Raster et sp de R pour auto-générer un lot de données d'apprentissage homogène. Dans ce tableau renvoie au nombre de point et % au pourcentage des classes dans la base donnée.

	Build-up	**Crop**	**FloodV**	**grass**	**Scrub**	**Soil**	**Tree**	**Water**
n	48813	65880	66880	83258	73682	93025	53108	61090
%	9	12	12	15	14	17	10	11

Tableau 3 : information sur la distribution des occupations du sol. FloodV indique la végétation aquatique

La troisième étape consiste de définir l'architecture du modèle du DNN. Il est aussi possible de le changer pour tester la précision du modèle. J'ai testé que 2 architectures, en appliquant de la régularisation avec un dropout de 40% et 30% pour optimiser le temps, et le overfitting. La quatrième partie consiste à entrainer le modèle et faire évaluation du modèle et puis exporter le fichier pour une future prédiction de classes sur une autre image d'une autre période. Il est aussi possible de changer ces paramètres, surtout le batch_size, le epochs et le Learning rate. Tous les paramètres de ces modèles sont disponibles en annexe dans le dossier code. Différentes mesures de performance ont été évalués parmi lesquelles la sensitivité, la spécificité, le Kappa, la précision etc. Différentes figures sont disponibles aussi dans le répertoire du projet.

La figure X illustre un résultat des mesures de performance du DNN ;

```
Confusion Matrix and Statistics

      y_test
y_pred      0      1
     0 766831    232
     1    166 109339

               Accuracy : 0.9995
                 95% CI : (0.9995, 0.9996)
    No Information Rate : 0.875
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9979

 Mcnemar's Test P-Value : 0.001121

            Sensitivity : 0.9998
            Specificity : 0.9979
         Pos Pred Value : 0.9997
         Neg Pred Value : 0.9985
             Prevalence : 0.8750
         Detection Rate : 0.8748
   Detection Prevalence : 0.8751
      Balanced Accuracy : 0.9988

       'Positive' Class : 0
```

Fig. 8 : Affichant des métriques évaluation de la performance du modèle DNN, comme la Sensitivité, la spécificité.

Ce calcul est fait en utilisant le package Caret sous R pour les mesures de performance, ou directement dans l'architecture du DNN. Plusieurs configuration des architectures du DNN ont été testé, en jouant sur les nombres de couches du réseau, le nombre de prédicteurs, et les algorithmes d'optimisation. Dans le répertoire du projet vous verrez les sorties des résultats sous forme de graphiques montrant les différentes métriques calculées. Les matrices de confusion entre les classes sont disponibles dans ces répertoires.

Conclusion

Ce rapport avait pour but de montrer comment extraire les Mask-culture soit sous GEE sous localement ; en suivant cette méthodologie il est possible de l'appliquer dans d'autres zones. Ainsi après plusieurs essaies, les prédicteurs a choisir pour bonne classification est de considérer ces bands : (B2, B3, B4, B8,B11, B12) qui sont les bandes (bleu, vert, rouge, PIR, MIR-1 & MIR-2) ce qui réduit le cout de calcul des indices spectraux. Ensuite considérer un DNN a 5 couches, suivi par des alternances de batch-normalisation et de dropout qui sont des méthodes de régularisations du réseau afin d'éliminer le sur-apprentissage du modèle. Concernant les optimisations de l'algorithme de back-propagation le meilleur choix dans ce cas est le rmsprop () était meilleur que SGD (Stochatic Gradient Descent) et le Adam (adaptive estimation of first-order and second-order moments). Pour le SVM fait dans la plateforme GEE, on recommande de calculer des paramètres du modèle (gamma et C) en utilisant le package Caret et ensuite implémenter ce résultat sous GEE. Pour l'exportation des résultats aussi, nous recommandons au moins pour la couche agriculture de passer par une vectorisation qui permet d'éliminer les artefacts. Pour une exportation aussi optimale, nous recommandons de faire séquentiellement en passant par les tuiles.

Recommandations

- La première recommandation fait référence à l'acquisition des données d'apprentissage, on recommande pour ces types de services de grande échelle, d'acheter des données a hautes résolution spatiale de l'ordre du mètre et de digitaliser les classes agriculture car les outils open-source pour faire ces types de procédures existent. Pour les autres occupations de sol, il existe énormément de base de données open qui permettent de les récupérer soit manuellement soit par ligne de code.
- Pour améliorer les zones de culture, je recommande l'utilisation de service cloud de type Google Earth Engine, de passer par un des algorithmes de machine Learning soit le Random Forest ou le Support Vector Machine Learning en prenant le soin de faire un zone agro-climatique car la classification va dépendre du signal spectral de la phénologie des cultures
- Il est préférable de passer par du Deep Learning pour mieux automatiser les prédictions car il est possible de sauvegarder la fonction de prédiction qui n'est le cas dans les autres types de classification. Mais il va nécessiter de s'équiper en infrastructure informatique ou de payer les services cloud comme google Earth engine ou AWS.