



UNIVERSITÉ DE MONTPELLIER

FACULTÉ DES SCIENCES

DÉPARTEMENT INFORMATIQUE

HMIN311M - EXTRACTION DE CONNAISSANCES AVANCÉE

Projet

# Détection de levée de culture à partir d'images satellites

*Étudiants :*

**Jeremy BRESSAND**

**Anastasiia PRYSIAZHNIUK**

*Responsables :*

**Jérôme AZE**

**Maguelonne TEISSEIRE**

Année 2017

# Table des matières

<b>Introduction</b>	<b>2</b>
<b>1 Prétraitements</b>	<b>5</b>
1.1 Comparaison NDVI . . . . .	5
1.2 Combinaison pixels-objets-parcelles . . . . .	6
1.3 Chevauchement objets-parcelles . . . . .	7
1.4 Représentativité objets-parcelles . . . . .	8
1.5 Temporalité . . . . .	9
<b>2 Construction et évaluation modèle CNN</b>	<b>11</b>
2.1 Construction jeux d'entraînement/test . . . . .	11
2.2 Entraînement des différents modèles . . . . .	12
2.2.1 CNN séquence de deux pixels sur jeu de données approche chevauchement . . . .	13
2.2.2 CNN séquence de trois pixels sur jeu de données approche chevauchement . . . .	13
2.2.3 CNN séquence de deux pixels sur jeu de données approche chevauchement + représentativité . . . . .	14
2.2.4 CNN séquence de trois pixels sur jeu de données approche chevauchement + représentativité . . . . .	15
2.3 Évaluation des différents modèles . . . . .	17
<b>3 Avantages et Inconvénients</b>	<b>20</b>
<b>Conclusion</b>	<b>21</b>

# Introduction

Ce rapport présente le travail effectué pour la réalisation d'un système permettant la détection de levée de récolte à la base d'une série d'images satellites capturée sur une zone au Nord-Est de la Tunisie.

Dans une première section nous allons présenter les différentes approches de prétraitement que l'on a utilisé pour nettoyer et transformer les données.

Puis dans une seconde section nous allons présenter comment nous avons construit et évalué les différents modèles sur différents jeux d'entraînement et de test.

Avant de conclure nous analyserons les atouts et les limites de notre approche de Deep Learning.

## Glossaire

**Objet est représentatif d'une parcelle** Dans ce rapport nous dirons qu'un objet est représentatif d'une parcelle si nous considérons que cet objet à le même label que la parcelle.

**Objet qui chevauche une parcelle** Dans ce rapport nous dirons qu'un objet chevauche une parcelle si la date d'acquisition d'image satellitaire de l'objet correspond à la date d'acquisition d'image satellitaire de la parcelle et qu'il existe des pixels appartenant à l'objet et à la parcelle.

**Objets** Dans ce rapport lorsque nous parlerons d'un objet nous désignerons un ensemble de couples de coordonnées géographiques associé à une date d'acquisition d'image satellitaire. Nous dirons qu'un objet est 'levé' si nous considérons que lors de la date d'acquisition de l'image satellitaire, les récoltes correspondant à la surface de terrain représentée par l'objet sont levées et inversement nous dirons qu'un objet est 'non levé' si nous considérons que lors de la date d'acquisition de l'image satellitaire, les récoltes correspondant à la surface de terrain représentée par l'objet ne sont pas levées.

**Parcelles** Dans ce rapport lorsque nous parlerons d'une parcelle nous désignerons un ensemble de couples de coordonnées géographiques associé à une date d'observation de levée de récolte et une date d'acquisition d'image satellitaire. Une parcelle est dite 'levée' si sa date d'acquisition d'image satellitaire est postérieure ou égale à sa date d'observation de levée des récoltes. Inversement une parcelle est dite 'non levée' si sa date d'acquisition d'image satellitaire est antérieure à sa date d'observation de levée.

**Pixel appartenant à l'objet** Dans ce rapport nous dirons qu'un pixel appartient à un objet si le couple de coordonnées géographiques du pixel appartient à l'ensemble de couples de coordonnées géographiques de l'objet et si la date d'acquisition d'image satellitaire du pixel est la même que la date d'acquisition d'image satellitaire de l'objet.

**Pixel appartenant à une parcelle** Dans ce rapport nous dirons qu'un pixel appartient à une parcelle si le couple de coordonnées géographiques du pixel appartient à l'ensemble de couples de coordonnées géographiques de la parcelle et si la date d'acquisition d'image satellitaire du pixel est la même que la date d'acquisition d'image satellitaire de la parcelle.

**Pixel qui précède un pixel** Dans ce rapport nous dirons qu'un pixel précède un deuxième pixel si les deux pixels ont le même couple de coordonnées géographiques et si la date d'acquisition d'image satellitaire du premier pixel est la date précédant la date d'acquisition d'image satellitaire du deuxième pixel dans l'ensemble des dates d'acquisition des images satellitaire.

**Pixels** Dans ce rapport lorsque nous parlerons d'un pixel nous désignerons un ensemble de 10 canaux correspondant à une date d'acquisition d'image satellitaire et à un couple de coordonnées géographiques. Nous dirons qu'un pixel est 'levé' si nous considérons que lors de la date d'acquisition de l'image satellitaire, les récoltes correspondant à la surface de terrain représentée

par le pixel sont levées et inversement nous dirons qu'un pixel est 'non levé' si nous considérons que lors de la date d'acquisition de l'image satellitaire, les récoltes correspondant à la surface de terrain représentée par le pixel ne sont pas levées.

# 1 Prétraitements

Dans cette section nous allons présenter les différentes étapes du prétraitement des données et à quelles problématiques sont-elles adressées.

Tout d’abord nous expliquerons pourquoi nous avons décidé de ne pas utiliser l’indice NDVI.

Nous allons après présenter l’approche globale choisie pour les prétraitements qui est de combiner les données correspondants aux Parcelles, aux Objets et aux Pixels.

Par la suite nous présenterons l’étape de résolution des chevauchements multiples entre un objet et plusieurs parcelles labelisées.

Puis nous présenterons l’étape de filtrage des objets représentatifs des parcelles labelisées.

Enfin nous présenterons l’étape de sélection de la temporalité à associer aux données.

## 1.1 Comparaison NDVI

Pour construire notre dataset labelisé nous avons trouvé deux choix possibles. Soit calculer l’indice de végétation pour définir des pixels qui sont ‘levés’, soit prendre en compte les données d’observations qui ont été fournies. Après plusieurs tests effectués en choisissant des différents seuils nous avons observé qu’aucun des seuils (0.1, 0.2, 0.4) ne correspond complètement aux observations sur les figures 1, 2 et 3. Ce que nous considérons comme incertitude en espérant que les observations faites par les agriculteurs sont correctes. On suppose que cette différence est causée par une sensibilité d’indice NDVI aux différentes conditions météorologiques. A la suite nous ne prendrons pas en considération cet indice et nous utiliserons les observations pour construire le jeu de données en combinant l’échelle du pixel, de la parcelle et de l’objet.

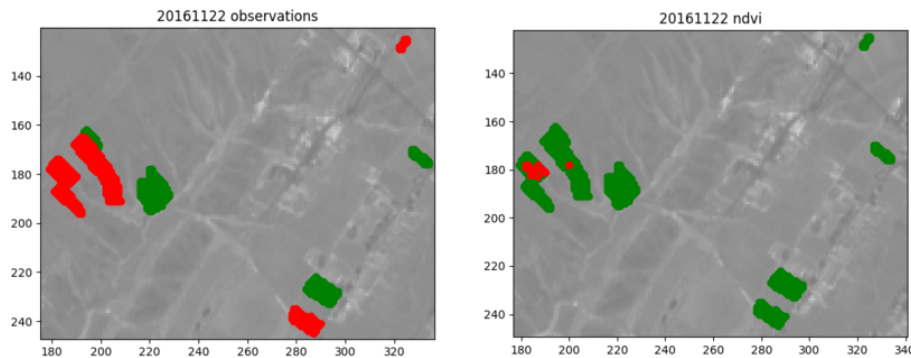


FIGURE 1 – Comparaison des parcelles levées (en vert) et non levées (en rouge) en se basant sur les observations des agriculteurs à gauche et des valeurs calculées en utilisant NDVI à droite, pixels verts ( $\text{NDVI} > 0.1$ ) et pixels rouges ( $\text{NDVI} < 0.1$ ) pour la date 2016/11/02.

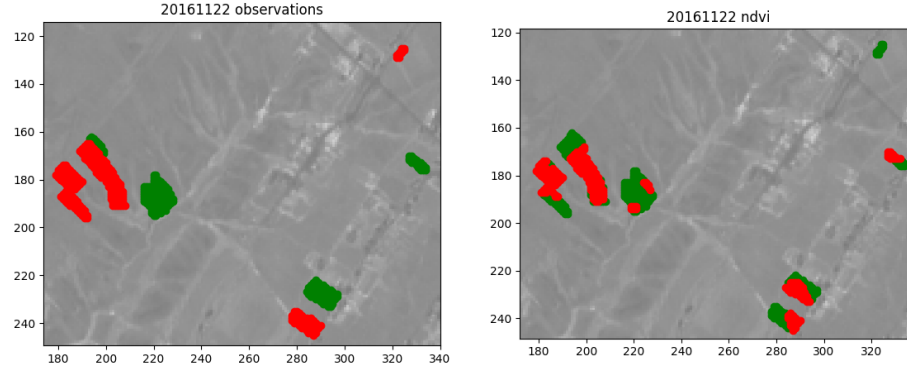


FIGURE 2 – Comparaison des parcelles levées (en vert) et non levées (en rouge) en se basant sur les observations des agriculteurs à gauche et des valeurs calculées en utilisant NDVI à droite, pixels verts ( $\text{NDVI} > 0.2$ ) et pixels rouges ( $\text{NDVI} < 0.2$ ) pour la date 2016/11/02.

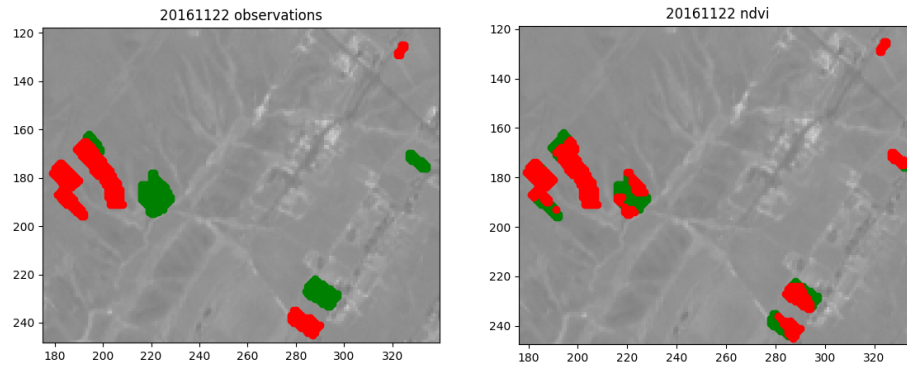


FIGURE 3 – Comparaison des parcelles levées (en vert) et non levées (en rouge) en se basant sur les observations des agriculteurs à gauche et des valeurs calculées en utilisant NDVI à droite, pixels verts ( $\text{NDVI} > 0.4$ ) et pixels rouges ( $\text{NDVI} < 0.4$ ) pour la date 2016/11/02.

## 1.2 Combinaison pixels-objets-parcelles

Dans ce projet nous avons le choix de travailler sur une approche basée sur les objets et/ou pixels. Nous avons choisi de travailler sur les pixels car nous utilisons des modèles deep-learning et nous avons besoin d'un grand nombre d'éléments labélisés pour entraîner et tester ces modèles.

L'approche la plus simple aurait été de labéliser chaque Pixel appartenant à une parcelle 'levée' comme étant 'levé' et inversement chaque pixel appartenant à une parcelle 'non levée' comme étant 'non levé'.

Cependant cette approche possède deux défauts :

- Les parcelles 'levées' peuvent contenir des pixels 'non levés' et inversement, ce qui n'est pas pris en compte dans cette approche.
- Nous pourrions labéliser les objets à partir des parcelles et les objets contiennent plus de pixels que les parcelles.

C'est pourquoi nous avons choisi une approche qui labelise les objets à partir des parcelles et qui récupère les pixels labelisés à partir des objets labelisés. Nous mettons en oeuvre plusieurs étapes pour labeliser les objets et les pixels que nous allons voir par la suite.

### 1.3 Chevauchement objets-parcelles

Une première approche pour labeliser les objets à partir des parcelles labelisées est de considérer qu'un Objet qui chevauche une parcelle possède le même label que cette parcelle.

Cependant nous avons constaté qu'un objet pouvait chevaucher plusieurs parcelles avec des labels différents.

C'est pourquoi nous avons créé une mesure que nous appellerons le rapport de chevauchement :

$$\delta_{ch}(obj) = \frac{ch_{lev}(obj)}{ch_{lev}(obj) + ch_{non\_lev}(obj)}$$

- $ch_{lev}(obj)$  : nombre de Pixel appartenant à l'objet obj et appartenant à une parcelle 'levée'.
- $ch_{non\_lev}(obj)$  : nombre de Pixel appartenant à l'objet obj et appartenant à une parcelle 'non levée'.

Nous considérons qu'un objet est 'levé' s'il possède au moins deux fois plus de pixels appartenant à des parcelles 'levées' que de pixels appartenant à des parcelles 'non levées' et inversement.

Nous labelisons les objets de la manière suivante :

```

Fonction labelise_objet_chevauchement( obj : Objet) : Label
    label : Label
    Si ( $\delta_{ch}(obj) \geq \frac{2}{3}$ ) Alors
        | label ← 'levé'
    Sinon
        Si ( $\delta_{ch}(obj) \leq \frac{1}{3}$ ) Alors
            | label ← 'non levé'
        Sinon
            | label ← 'aucun label'
        Fin Si
    Fin Si
    Retourner label
Fin

```

Algorithme 1: Labeliser les objets grâce au rapport de chevauchement

Si l'objet est 'levé' alors nous enlevons de cet objet tous les pixels qui appartiennent à une parcelle 'non levée'. Inversement si l'objet est 'non levé' alors nous enlevons de cet objet tous les pixels qui appartiennent à une parcelle 'levée'.



TABLE 1 – Nombre de pixels levés et non levés pour l’approche par parcelle et l’approche combinant les parcelles et les objets grâce au rapport de chevauchement.

Nombre de pixels	Approche parcelle	Approche parcelle/objet
levées	1029	3473
non levées	2191	7129

Pour terminer nous labelisons tous les pixels appartenant à des objets ‘levés’ comme ‘levés’ et tous les pixels appartenant à des objets ‘non levés’ comme ‘non levés’.

Cette approche permet d’obtenir plus de pixels labelisés que l’approche précédente comme nous pouvons le voir dans le tableau 2 cependant elle ne prend pas en compte que les parcelles ‘levées’ peuvent contenir des pixels ‘non levés’ et inversement.

Une solution à ce problème est de prendre en compte la représentativité des objets auprès des parcelles.

## 1.4 Représentativité objets-parcelles

La problématique soulevée précédemment est qu’une parcelle ‘levée’ peut contenir des pixels ‘non levés’ et inversement.

Comme nous considérons que les pixels appartenant à un même objet (dont on a enlevé des pixels lors de l’étape de chevauchement) ont le même label, on peut reformuler cette problématique de cette manière : un objet ‘non levé’ peut chevaucher une parcelle ‘levée’ et inversement.

La question qui se pose est comment déterminer si un objet est représentatif d’une parcelle.

Pour cela nous considérons qu’une parcelle est ‘levée’ s’il y a au moins deux fois plus de pixels ‘levés’ que de pixels ‘non levés’ appartenant à cette parcelle.

Pour être sûr qu’un Objet est représentatif d’une parcelle, il faut donc que l’objet possède au moins un tiers des pixels de la parcelle.

C’est pourquoi nous avons créé deux mesures que nous appellerons la confiance de levée d’un objet et la confiance de non levée d’un objet :

$$\rho_{lev}(objet) = \max_{parcelle \in P_{lev}} \frac{intersection(parcelle, objet)}{taille(parcelle)}$$

$$\rho_{non\_lev}(objet) = \max_{parcelle \in P_{non\_lev}} \frac{intersection(parcelle, objet)}{taille(parcelle)}$$

—  $intersection(parcelle, objet)$  : Nombre de pixels appartenant à parcelle et à objet.

—  $taille(parcelle)$  : Nombre de pixels dans la parcelle.

—  $P_{lev}$  : Ensemble des parcelles ‘levées’.

—  $P_{non\_lev}$  : Ensemble des parcelles ‘non levées’.

Finalement nous labelisons les objets de la manière suivante :

**Fonction** `labelise_objet_chevauchement_confiance( obj : Objet ) : Label`

```

    label : Label
    Si ( $\delta_{ch}(obj) \geq \frac{2}{3}$ ) Alors
        enlever_pixels_parcelles_non_levees(obj)
        Si ( $\rho_{lev}(objet) \geq \frac{1}{3}$ ) Alors
            label  $\leftarrow$  'levé'
        Sinon
            label  $\leftarrow$  'aucun label'
        Fin Si
    Sinon
        Si ( $\delta_{ch}(obj) \leq \frac{1}{3}$ ) Alors
            enlever_pixels_parcelles_levees(obj)
            Si ( $\rho_{non\_lev}(objet) \geq \frac{1}{3}$ ) Alors
                label  $\leftarrow$  'non levé'
            Sinon
                label  $\leftarrow$  'aucun label'
            Fin Si
        Sinon
            label  $\leftarrow$  'aucun label'
        Fin Si
    Fin Si
    Retourner label
Fin

```

Algorithme 2: Labeliser les objets grâce au rapport de chevauchement et aux confiances de levée et de non levée.

Cette approche a le mérite de fournir des données fiables cependant elle fait diminuer énormément le nombre de pixels comme nous pouvons voir dans le tableau 2.

## 1.5 Temporalité

Un défaut de l'approche précédente est de ne pas prendre en compte l'évolution des canaux aux cours du temps.

En effet, les données d'un Pixel qui précède un pixel peuvent être utiles à nos modèles pour prédire si ce dernier pixel est 'levé' ou non 'levé'.

C'est pourquoi nous avons choisi une approche ne se basant pas sur des pixels mais sur des séquences de pixels.

TABLE 2 – Nombre de pixels levés et non levés pour l’approche utilisant uniquement le rapport de chevauchement et l’approche utilisant le rapport de chevauchement et les confiances de levé et non levé.

Nombre de pixels	Approche chevauchement	Approche chevauchement + représentativité
levées	3473	1119
non levées	7129	2393

Nous avons construit différents jeux de données avec des tailles de séquences de pixels différentes :

- Jeu de séquences de 1 pixel : C’est le jeu de données que l’on obtient avec l’approche précédente.
- Jeu de séquences de 2 pixels : On prend le jeu de séquences de 1 pixel et pour chaque séquence, si le pixel dans la séquence possède un pixel qui le précède alors on ajoute ce dernier pixel à la séquence, sinon on supprime la séquence.
- Jeu de séquences de 3 pixels : On prend le jeu de séquences de 2 pixels et pour chaque séquence, si le dernier pixel dans la séquence possède un pixel qui le précède alors on ajoute ce pixel qui le précède à la séquence, sinon on supprime la séquence.
- Jeu de séquences de 4 pixels : On prend le jeu de séquences de 3 pixels et pour chaque séquence, si le dernier pixel dans la séquence possède un pixel qui le précède alors on ajoute ce pixel qui le précède à la séquence, sinon on supprime la séquence.

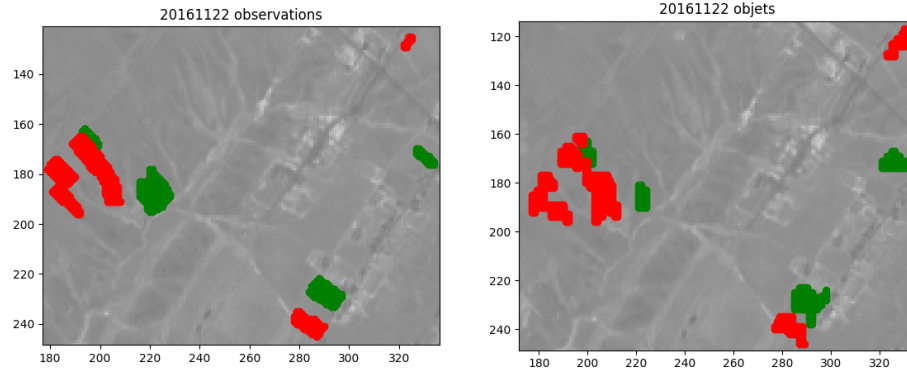


FIGURE 4 – Comparaison des parcelles levées en vert et non levées en rouge d’après les observations des parcelles à gauche avec les pixels levés en vert et non levés en rouge d’après le jeu de données de l’approche chevauchement et représentativité pour la date 2016/11/02.

Nous pouvons voir la quantité de données pour chaque jeu de séquences dans le tableau 3.

Pour le jeu de séquences de trois pixels de l’approche chevauchement nous pouvons visualiser la répartition des pixels levés et non levés par rapport aux observations des parcelles sur la figure 5 et pour l’approche chevauchement avec représentativité sur la figure 4.

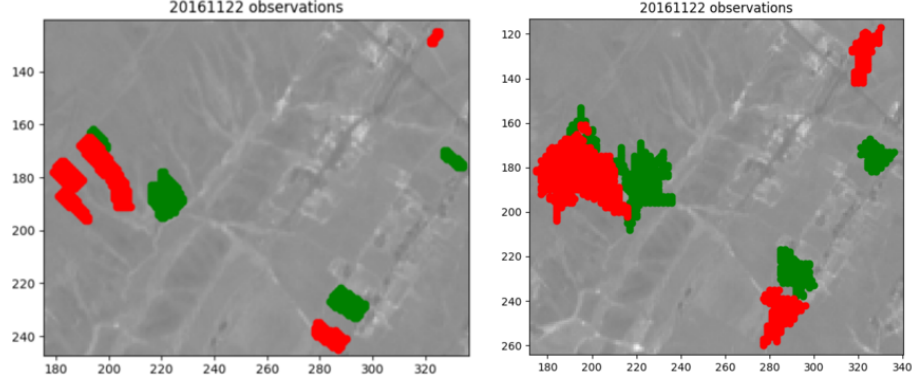


FIGURE 5 – Comparaison des parcelles levées en vert et non levées en rouge d’après les observations des parcelles à gauche avec les pixels levés en vert et non levés en rouge d’après le jeu de données de l’approche chevauchement pour la date 2016/11/02.

TABLE 3 – Nombre de séquences levées et non levées par jeu de données.

Nombre de séquences levées / non levées	Séquence 1 pixel	Séquence 2 pixels	Séquence 3 pixels	Séquence 4 pixels
Chevauchement	3473/7129	3473/5041	3473/3067	3036/1298
Chevauchement + représentativité	1119/2393	1119/1740	1119/1104	1029/478

## 2 Construction et évaluation modèle CNN

Dans la section précédente nous avons présenté notre approche pour la construction des jeux de données.

Dans cette section nous allons tout d’abord présenter l’étape de construction des jeux d’entraînement et des jeux de test.

Puis nous allons présenter nos différents modèles CNN et notre approche pour choisir les hyperparamètres associés.

Enfin nous terminerons par comparer les performances de nos CNN.

### 2.1 Construction jeux d’entraînement/test

Nous utilisons que les jeux de données de séquences de trois pixels avec l’utilisation de la notion du chevauchement uniquement et avec l’utilisation des notions de chevauchement et de représentativité.

Pour chaque jeu de données nous le séparons en un jeu d’entraînement et un jeu de test de tailles similaires comme nous pouvons voir sur le tableau 4.

Nous allons à présent pouvoir entraîner nos différents modèles CNN sur les différents jeux de données.

TABLE 4 – Nombre de séquences levées et non levées dans les jeux d’entraînement et de test des deux jeux de données.

Nombre de séquences levées/non levées	Approche chevauchement	Approche chevauchement + représentativité
Jeu d’entraînement	1747 / 1542	562 / 555
Jeu de test	1726 / 1525	557 / 549

## 2.2 Entraînement des différents modèles

Nous possédons deux jeux de données : le jeu de données obtenu par l’approche chevauchement et le jeu de données obtenu par l’approche chevauchement + représentativité.

Pour chaque jeu de données nous allons entraîner deux CNN :

— *Un CNN prenant en entrée des séquences de deux pixels :*

Le jeu de données étant un ensemble de séquences de 3 pixels, ce CNN prendra en compte les deux premiers pixels de chaque séquence.

— *Un CNN prenant en entrée des séquences de trois pixels :*

Ce CNN prendra en compte tous les pixels de chaque séquence du jeu de données.

L’intérêt de comparer les performances de ces deux modèles CNN sur chaque jeu de données est de déterminer qu’elle est la séquence temporelle utile à appliquer aux pixels pour déterminer leurs levés.

Pour construire nos modèles nous prenons en considération des séquences temporelles de deux ou trois pixels. Nous aimerions savoir si l’ajout d’un autre pixel pour chaque séquence temporelle serait bénéfique pour la prédiction.

L’étape d’entraînement pour chaque modèle nécessite de tester différentes valeurs pour les hyperparamètres et d’analyser les performances associées pour trouver la meilleure configuration pour chaque modèle.

Pour chaque modèle CNN nous prenons en compte les différents hyperparamètres suivants :

- Le nombre d’époques d’entraînement.
- Le nombre de couples de couches convolution + pooling.
- Le nombre de noeuds dans la couche ‘fully connected’ après les couches de convolutions et de pooling.

Pour chaque modèle CNN nous fixons les valeurs des différents hyperparamètres suivants :

- La taille du batch d’entraînement : 128.
- Le nombre de filtres dans la première convolution : 32.
- Le nombre de filtres dans les couches de convolution suivantes : 64.
- La taille des noyaux des couches de convolution : (1,2).
- Le ‘stride’ des couches de convolution : (1,1).

- La méthode d'activation des couches de convolution : relu.
- La taille de la fenêtre (en anglais 'pooling size') des couches de pooling : (1,2).
- Le 'stride' des couches de pooling : (1,1).
- La méthode d'activation dans la couche 'fully connected' après les couches de convolutions et de pooling : relu.
- La méthode d'activation dans la dernière couche 'fully connected' : softmax.

### 2.2.1 CNN séquence de deux pixels sur jeu de données approche chevauchement

Nous pouvons voir les différentes performances associées aux différentes configurations de notre CNN travaillant sur des séquences de deux pixels sur le jeu de données correspondant à l'approche chevauchement dans le tableau 5.

Nous pouvons voir sur les 5 premières lignes du tableau que les performances du modèle augmentent avec l'augmentation du nombre d'époques d'entraînement.

Cependant à la cinquième ligne nous pouvons observer une différence significative entre les performances du modèle sur le jeu d'entraînement et les performances sur le jeu de test ce qui peut s'expliquer par un problème d' 'overfitting'.

Nous diminuons donc de la ligne 5 à 7 le nombre de noeuds dans la couche 'fully connected' ce qui a pour conséquence de diminuer la complexité du modèle et d'homogénéiser les performances entre celles sur le jeu d'entraînement et celles sur le jeu de test.

Nous décidons alors d'augmenter la complexité du modèle en ajoutant une deuxième couche de convolution et pooling et nous voyons sur les lignes 8 à 10 que les performances du modèle augmentent avec l'augmentation du nombre d'époques d'entraînement.

### 2.2.2 CNN séquence de trois pixels sur jeu de données approche chevauchement

Nous pouvons voir les différentes performances associées aux différentes configurations de notre CNN travaillant sur des séquences de deux pixels sur le jeu de données correspondant à l'approche chevauchement dans le tableau 6.

Nous pouvons observer sur les deux premières lignes du tableau que l'augmentation du nombre d'époques d'entraînement a pour conséquence l'augmentation globales des performances du modèle.

Cependant à la ligne 2 nous observons une différence significative entre les performances du modèle sur le jeu d'entraînement et les performances du modèle sur le jeu de test. Cette différence peut s'expliquer par un problème d' 'overfitting'.

Nous décidons de faire baisser le nombre de noeuds dans la couche 'fully connected' sur les lignes 3 et 4 et d'augmenter le nombre d'époques d'entraînement du modèle afin d'obtenir de meilleurs performances qui soient plus homogènes entre les jeux d'entraînement et de test.

N'ayant pas réussi à homogénéiser les performances mais ayant obtenu de meilleurs performances nous avons augmenté le nombre de couche de convolution et de pooling sur les lignes 5 et 6 avec les

TABLE 5 – Moyennes sur trois exécutions du CNN séquence de deux pixels sur jeu de données approche chevauchement.

Nombre d'époques d'entraînement	Nombre de couches de convolution + pooling	Nombre de noeuds dans la couche 'fully connected'	Performances sur jeu d'entraînement (Precision / Rappel / F-mesure)	Performances sur jeu de test (Presicion / Rappel / F-mesure)
20	1	1000	0.827 / 0.809 / 0.816	0.829 / 0.809 / 0.817
50	1	1000	0.861 / 0.850 / 0.855	0.870 / 0.846 / 0.858
100	1	1000	0.886 / 0.891 / 0.887	0.879 / 0.882 / 0.879
150	1	1000	0.872 / 0.933 / 0.901	0.868 / 0.926 / 0.896
200	1	1000	0.916 / 0.926 / 0.921	0.900 / 0.913 / 0.906
200	1	750	0.929 / 0.920 / 0.924	0.917 / 0.905 / 0.910
200	1	500	0.924 / 0.887 / 0.905	0.918 / 0.881 / 0.899
300	2	500	0.952 / 0.915 / 0.933	0.924 / 0.884 / 0.909
400	2	500	0.977 / 0.926 / 0.951	0.940 / 0.890 / 0.914
500	2	500	0.961 / 0.969 / 0.965	0.913 / 0.928 / 0.920

mêmes nombres de noeuds dans la couche 'fully connected'. Nous avons augmenté les performances du modèle mais nous n'avons toujours un problème d' 'overffiting'.

Sur la ligne 7 nous avons augmenté le nombre d'époques d'entraînement ce qui a eut pour effet l'augmentation des performances. Nous avons essayé de régler le problème d' 'overfitting' en diminuant le nombre de noeuds dans la couche 'fully connected' à la ligne 8. Nous observeons une légère homogénéisation des performances au prix d'une baisse de performances.

Enfin à la ligne 9 nous avons augmenté le nombre d'époques d'entraînement ce qui n'a pas eu de conséquence significative sur les performances du modèle.

### 2.2.3 CNN séquence de deux pixels sur jeu de données approche chevauchement + représentativité

Nous pouvons voir les différentes performances associées aux différentes configurations de notre CNN travaillant sur des séquences de deux pixels sur le jeu de données correspondant à l'approche chevauchement + représentativité dans le tableau 7.

Sur les deux premières lignes du tableau nous observons que l'augmentation du nombre d'époques d'entraînement du modèle à pour conséquence l'augmentation de ses performances globales mais aussi l'apparition d'un problème d' 'overfitting' que nous essayons de régler à la ligne 3 sans succès.

Sur la ligne 4 nous observons que l'ajout d'une couche de convolution et de pooling a pour effet l'augmentation des performances globales du modèles mais aussi la persistance du problème d' 'overfitting' que nous essayons de régler à la ligne 5 sans succès.

TABLE 6 – Moyennes sur trois exécutions du CNN séquence de trois pixels sur jeu de données approche chevauchement.

Nombre d'époques d'entraînement	Nombre de couches de convolution + pooling	Nombre de noeuds dans la couche 'fully connected'	Performances sur jeu d'entraînement (Precision / Rappel / F-mesure)	Performances sur jeu de test (Presicion / Rappel / F-mesure)
100	1	1000	0.905 / 0.868 / 0.886	0.898 / 0.859 / 0.878
200	1	1000	0.952 / 0.890 / 0.919	0.938 / 0.875 / 0.905
300	1	750	0.958 / 0.928 / 0.943	0.935 / 0.910 / 0.922
300	1	500	0.939 / 0.908 / 0.922	0.927 / 0.893 / 0.909
300	2	750	0.964 / 0.987 / 0.975	0.935 / 0.960 / 0.947
300	2	500	0.940 / 0.978 / 0.959	0.915 / 0.961 / 0.937
400	2	500	0.990 / 0.990 / 0.990	0.953 / 0.959 / 0.956
400	2	250	0.981 / 0.975 / 0.978	0.948 / 0.946 / 0.947
500	2	250	0.979 / 0.974 / 0.976	0.946 / 0.946 / 0.946

Sur la ligne 6 nous augmentons le nombre d'époques d'entraînement ce qui améliore les performances du modèle et nous n'arrivons pas à régler le problème d'overfitting sur les lignes 7 et 8.

Enfin l'augmentation du nombre d'époques d'entraînement à la ligne 9 fait augmenter les performances tandis que l'augmentation du nombre d'époques à la ligne 10 n'a pas d'effet significatif.

#### 2.2.4 CNN séquence de trois pixels sur jeu de données approche chevauchement + représentativité

Nous pouvons voir les différentes performances associées aux différentes configurations de notre CNN travaillant sur des séquences de trois pixels sur le jeu de données correspondant à l'approche chevauchement + représentativité dans le tableau 8.

Sur les deux premières lignes nous observons que l'augmentation du nombre d'époques d'entraînement fait augmenter les performances mais fait aussi apparaître un problème d' 'overfitting' que l'on échoue à corriger à la ligne 3.

Sur les lignes 4 et 5 l'augmentation du nombre d'époques d'entraînement fait augmenter les performances du modèle.

Sur la ligne 6 l'ajout d'une couche de convolution et de pooling permet d'augmenter les performances et sur la ligne 7 nous échouons à résoudre le problème d' 'overfitting'.

Sur la ligne 8 l'augmentation du nombre d'époques d'entraînement permet d'augmenter les performances et sur la ligne 9 nous échouons à résoudre le problème d' 'overfitting'.



TABLE 7 – Moyennes sur trois exécutions du CNN séquence de deux pixels sur jeu de données approche chevauchement + représentativité.

Nombre d'époques d'entraînement	Nombre de couches de convolution + pooling	Nombre de noeuds dans la couche 'fully connected'	Performances sur jeu d'entraînement (Precision / Rappel / F-mesure)	Performances sur jeu de test (Precision / Rappel / F-mesure)
100	1	1000	0.869 / 0.916 / 0.892	0.851 / 0.916 / 0.882
200	1	1000	0.959 / 0.920 / 0.939	0.927 / 0.895 / 0.911
200	1	500	0.931 / 0.899 / 0.914	0.901 / 0.895 / 0.897
200	2	1000	0.945 / 0.949 / 0.947	0.908 / 0.931 / 0.919
200	2	500	0.950 / 0.908 / 0.927	0.919 / 0.899 / 0.907
300	2	1000	0.973 / 0.916 / 0.942	0.940 / 0.888 / 0.911
300	2	500	0.974 / 0.959 / 0.967	0.935 / 0.934 / 0.934
300	2	250	0.956 / 0.951 / 0.953	0.921 / 0.928 / 0.925
400	2	500	0.980 / 0.953 / 0.966	0.940 / 0.928 / 0.933
500	2	500	0.963 / 0.957 / 0.958	0.931 / 0.935 / 0.931

TABLE 8 – Moyennes sur trois exécutions du CNN séquence de trois pixels sur jeu de données approche chevauchement + représentativité.

Nombre d'époques d'entraînement	Nombre de couches de convolution + pooling	Nombre de noeuds dans la couche 'fully connected'	Performances sur jeu d'entraînement (Precision / Rappel / F-mesure)	Performances sur jeu de test (Precision / Rappel / F-mesure)
100	1	1000	0.883 / 0.910 / 0.895	0.865 / 0.905 / 0.883
200	1	1000	0.937 / 0.945 / 0.941	0.901 / 0.923 / 0.912
200	1	500	0.925 / 0.920 / 0.922	0.902 / 0.910 / 0.905
300	1	500	0.957 / 0.903 / 0.928	0.945 / 0.890 / 0.915
400	1	500	0.966 / 0.944 / 0.954	0.943 / 0.917 / 0.930
400	2	500	0.981 / 0.987 / 0.984	0.937 / 0.958 / 0.947
400	2	250	0.973 / 0.973 / 0.973	0.928 / 0.948 / 0.937
500	2	500	0.995 / 0.992 / 0.994	0.950 / 0.955 / 0.952
500	2	250	0.996 / 0.960 / 0.978	0.964 / 0.921 / 0.942

## 2.3 Évaluation des différents modèles

Pour chaque jeu de données, nous considérons deux modèles performants associés :

- Le modèle ayant obtenu le meilleur score de précision sur le jeu de test parmi les deux modèles ayant obtenu les meilleurs scores de f-mesure.
- Le modèle ayant obtenu le meilleur score de rappel sur le jeu de test parmi les deux modèles ayant obtenu les meilleurs scores de f-mesure.

En effet les mesures de précision et de rappel sont toutes les deux importantes. Quelqu'un qui accorde plus d'importance à la mesure de rappel est quelqu'un qui veut ne pas rater la date de levée des cultures quitte à ce qu'il se rende compte en observant sur place que la levée n'est pas encore apparue.

A l'inverse quelqu'un qui accorde plus d'importance à la mesure de précision qu'à la mesure de rappel est quelqu'un qui souhaite pas vérifier plusieurs fois les prédictions faites par le modèle quitte à rater la date de levée des culture.

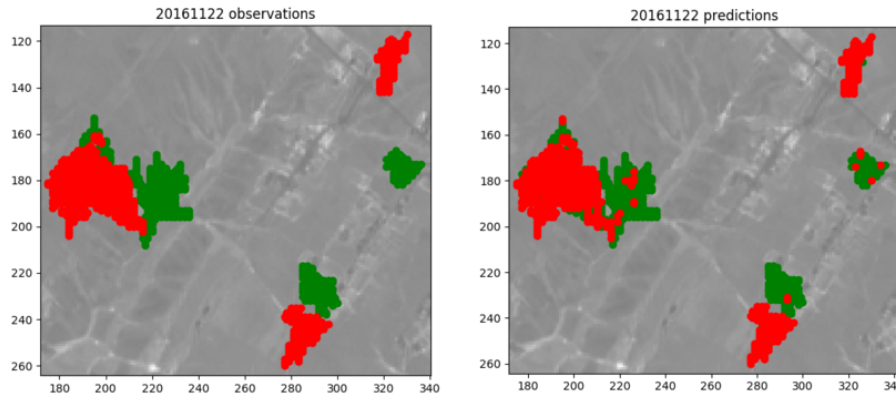


FIGURE 6 – Comparaison des pixels levées (en vert) et non levées (en rouge) en se basant sur le jeu de données de l'approche chevauchement à gauche et des pixels levés et non levés qui ont été prédit à partir du modèle CNN séquence de trois pixels privilégiant la précision et entraîné sur les données de l'approche chevauchement à droite, pour la date 2016/11/02.

Pour le jeu de données de l'approche chevauchement, les modèles performants associés sont :

- Le modèle privilégiant la précision : Modèle CNN séquence de trois pixels avec nombre d'époques d'entraînement = 400, nombre de couche de convolution + pooling = 2, nombre de noeuds dans la couche 'fully connected' = 500, performances sur jeu de test : 0.953 / 0.959 / 0.956 (figure 6).
- Le modèle privilégiant le rappel : Modèle CNN séquence de trois pixels avec nombre d'époques d'entraînement = 300, nombre de couche de convolution + pooling = 2, nombre de noeuds dans

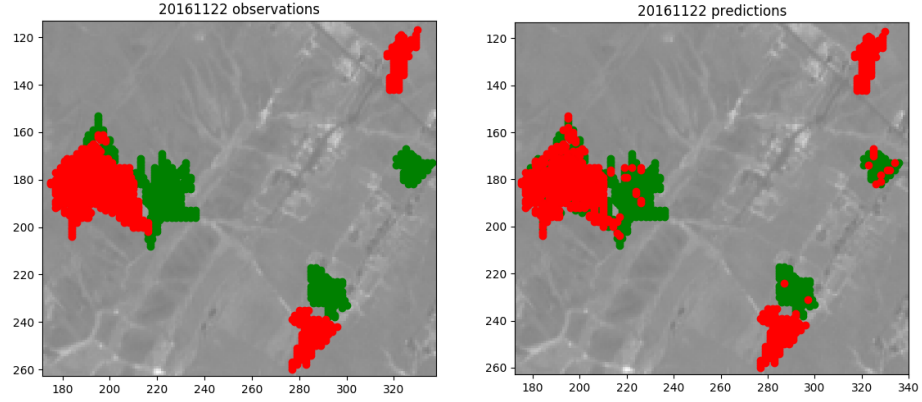


FIGURE 7 – Comparaison des pixels levées (en vert) et non levées (en rouge) en se basant sur le jeu de données de l’approche chevauchement à gauche et des pixels levés et non levés qui ont été prédit à partir du modèle CNN séquence de trois pixels privilégiant le rappel et entraîné sur les données l’approche chevauchement à droite, pour la date 2016/11/02.

la couche 'fully connected' = 750, performances sur jeu de test : 0.935 / 0.960 / 0.947 (figure 7).

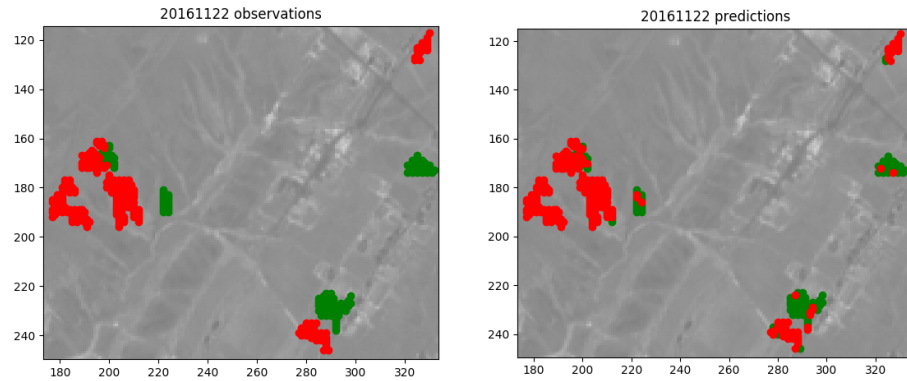


FIGURE 8 – Comparaison des pixels levées (en vert) et non levées (en rouge) en se basant sur le jeu de données de l’approche chevauchement avec représentativité à gauche et des pixels levés et non levés qui ont été prédit à partir du modèle CNN séquence de trois pixels privilégiant la précision et entraîné sur les données de l’approche chevauchement avec représentativité à droite, pour la date 2016/11/02.

Pour le jeu de données de l’approche chevauchement avec représentativité, les modèles performants associés sont :

- Le modèle privilégiant la précision : Modèle CNN séquence de trois pixels avec nombre d’époques d’entraînement = 500, nombre de couche de convolution + pooling = 2, nombre de noeuds dans la couche 'fully connected' = 500, performances sur jeu de test : 0.950 / 0.955 / 0.952 (figure 8).

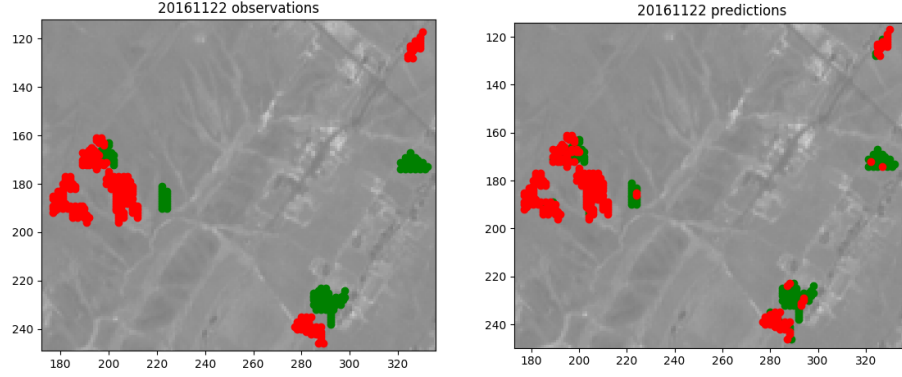


FIGURE 9 – Comparaison des pixels levées (en vert) et non levées (en rouge) en se basant sur le jeu de données de l’approche chevauchement avec représentativité à gauche et des pixels levés et non levés qui ont été prédit à partir du modèle CNN séquence de trois pixels privilégiant le rappel et entraîné sur les données de l’approche chevauchement avec représentativité à droite, pour la date 2016/11/02.

- Le modèle privilégiant le rappel : Modèle CNN séquence de trois pixels avec nombre d’époques d’entraînement = 400, nombre de couche de convolution + pooling = 2, nombre de noeuds dans la couche ‘fully connected’ = 500, performances sur jeu de test : 0.937 / 0.958 / 0.947 (figure 9).

### 3 Avantages et Inconvénients

En réalisant ce projet nous avons utilisé une approche de fouille de données, Deep Learning, plus précisément l'algorithme de CNN. Nous avons pu remarquer plusieurs avantages et inconvénients d'utilisation de cette approche pour la télédétection. Parmi les avantages nous avons :

- La découverte des 'features' utiles automatiquement.
- Une approche qui permet créer des types de modèles de nature différentes et complexes.
- Une compréhension plus facile par rapport aux SVM.
- Facilité d'interprétation de résultats grâce à la visualisation sur une carte.

Parmi les inconvénients on trouve :

- Le besoin d'un vaste jeu de données labélisées.
- Le temps de calcul assez élevé.
- Problème de fiabilité de données (par exemple un pixel appartenant à une parcelle levée est-il levé?).

## Conclusion

Ce projet nous a apporté une expérience enrichissante dans une approche de Deep Learning de plus en plus utilisé dans les milieux scientifiques et en entreprise pour résoudre des problèmes complexes dans différents domaines.

Nous avons affronté plusieurs défis, premièrement les problèmes de sélection des données labelisées et fiables. Deuxièmement la mise en oeuvre du modèle CNN avec le framework Python Keras. Puis, la sélection du modèle le plus performant en effectuant plusieurs tests avec des configurations différentes. Nous avons de plus comparé les prédictions faites par nos modèles en les visualisant sur des cartes.

De plus nous sommes très intéressés par le domaine du machine learning et cette expérience sera très utile pour l'un de nous qui souhaite travailler dans le machine learning.