



Land Cover Classification Project - Report Guide and Results Analysis

Hugo Dupuis

Département Sciences du Numérique - Troisième année
2023-2024

Table des matières

1	Introduction	3
2	Model Selection	3
3	Data Preparation	3
4	Training and Parameters	4
5	Model Evaluation	4
6	Results Analysis	5
7	Results Visualisation	5
8	Conclusion	6

1 Introduction

La classification d'images satellites au moyen du deep learning est une problématique majeure dans le domaine de l'analyse spatiale. Avec la prolifération des satellites délivrant une quantité massive de données visuelles, la capacité à extraire des informations significatives à partir de ces images revêt une importance croissante. Ce projet vise à exploiter les avancées du deep learning pour automatiser la classification des images satellites, offrant ainsi des solutions efficaces pour des applications diverses, telles que la surveillance environnementale, la gestion des ressources naturelles, et la prévision des catastrophes.

Les défis que nous pouvons rencontrer dans ce cadre sont de différentes natures : cela peut venir de la variabilité des conditions atmosphériques qui perturbent la visibilité des paysages, de l'hétérogénéité des paysages rencontrés (les classes sont assez éloignées entre elles) ou encore la résolution des satellites qui peut varier, parmi tant d'autres.

Nous disposons d'une base d'entraînement de 20 000 images labélisées, dont 80% ont servi à l'entraînement du modèle (16 000 images) et les 20% d'images restantes ont servi à la validation du modèle, pour en évaluer sa précision et sa robustesse notamment.

Pour ce projet, j'ai utilisé un ResNet50 pour classifier les différentes images en fonction de leurs paysages.

2 Model Selection

J'ai testé et expérimenté de nombreux modèles durant ce projet, afin d'avoir plusieurs approches au problème. Le premier modèle testé était un simple CNN avec deux couches de convolution, en utilisant du Maxpooling et une fonction d'activation ReLU, et enfin, une softmax en sortie.

Je me suis ensuite tourné sur un ResNet50, réseau plus profond qui permet d'extraire des caractéristiques plus discriminantes dans les images. Il est également plus robuste puisque les connexions du réseau permettent d'éviter les problèmes de vanishing gradients. Enfin, le ResNet50 permet de capturer des infos à différentes échelles, ce qui peut être intéressant dans le cadre d'images satellites prises à différentes échelles.

Afin de voir si un autre modèle pouvait donner des résultats encore meilleurs, j'ai testé d'entraîner un UNet préentraîné. Le UNet étant plus utilisé dans la segmentation d'images plutôt que pour de la classification, j'ai dû modifier un peu l'architecture pour avoir des labels en sortie et non des images, je n'obtenais pas des résultats meilleurs.

Je suis donc resté sur un ResNet50, que j'ai préentraîné afin d'augmenter ses performances.

3 Data Preparation

Dans la version originale du projet, les images ont été normalisées et converties en tenseurs, et la précision obtenue était de 93.03%. J'ai appliqué d'autres techniques de pré-processing qui ont permis d'arriver à une précision de 95.71%, grâce à l'augmentation de données.

Les transformations que j'ai appliqué sont : une rotation de l'image d'un angle aléatoire, une translation aléatoire, un zoom aléatoire, et l'ajout d'un bruit gaussien. Ces transformations ont permis d'augmenter le jeu d'apprentissage initial afin d'avoir une plus grande base pour apprendre le modèle.

J'ai remarqué qu'après avoir ajouté d'autres transformations en plus de celles évoquées précédemment, comme un flip horizontal ou vertical, l'ajout d'un contraste ou d'une transformation élastique, le modèle se perdait complètement dans la classification et affichait des résultats très mauvais avec une précision de l'ordre de 40%.

4 Training and Parameters

La fonction de loss utilisée était la CrossEntropyLoss de PyTorch, j'ai utilisé un optimiseur Adam, avec un paramètre de learning rate à 0.001.

Pour déterminer les hyperparamètres optimaux (la Batch Size et le Learning Rate de l'optimiseur), j'ai réalisé une validation en prenant 3 différentes tailles de batch (16, 32 et 64) et 3 différentes valeurs pour le learning rate (0.001, 0.01 et 0.1) pour voir quels paramètres maximisaient la précision globale.

Les résultats sont dans le tableau ci-dessous :

Learning Rate / Batch Size	16	32	64
0.001	0.9355	0.951	0.95
0.01	0.8938	0.903	0.9197
0.1	0.6975	0.793	0.8782

TABLE 1 – Précision suivant la valeur des hyperparamètres

Au vu des résultats précédents, j'ai donc opté pour un batchsize de 32, que j'ai entraîné sur 50 epochs.

5 Model Evaluation

Pour évaluer la précision du modèle, en plus de l'accuracy qui a atteint 95.71%, j'ai affiché la matrice de confusion, qui indique pour chaque classe, le nombre d'images bien labélisées.

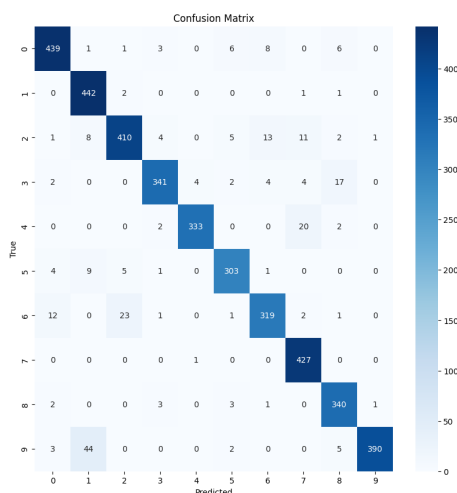


FIGURE 1 – Matrice de confusion

On peut remarquer qu'il y a des petits clusters avec des labels erronés (exemple : la classe 1 et la classe 9) qui caractérisent des classes assez proches au niveau du paysage.

D'autres méthodes de validation ont été réalisées : le recall, qui mesure la capacité du modèle à repérer les exemples positifs, ou le F1-score, qui combine le recall et la précision. On obtient des valeurs de 0.936 pour les deux mesures.

6 Results Analysis

Après analyse de la matrice de confusion, on peut dire que certaines classes, étant très proches les unes des autres, sont difficiles à classifier. Parmi les erreurs de classification, le label SeaLake est prédit comme étant une Forest, ou encore le label Industrial qui est parfois prédit en Residential.

En revanche, d'autres classes approchent des 100% de classifications correctes, c'est le cas pour le label Residential (427/428), le label Forest (442/446) ou encore le label River (340/350).

De plus, on a effectué de l'ajout de bruit dans l'augmentation de données, ce qui permet à notre modèle d'être robuste.

Les résultats sont satisfaisants, on réussit à atteindre une précision assez importante avec un modèle plutôt simple, mais robuste. On a tout de même un certain déséquilibre dans l'efficacité du modèle selon les classes : certaines classes ont presque 100% d'accuracy tandis que d'autres ont nettement plus d'erreurs. Le modèle pourrait être amélioré afin de mieux tenir compte des classes qui sont proches, peut-être en distinguant davantage de caractéristiques sur l'image.

7 Results Visualisation

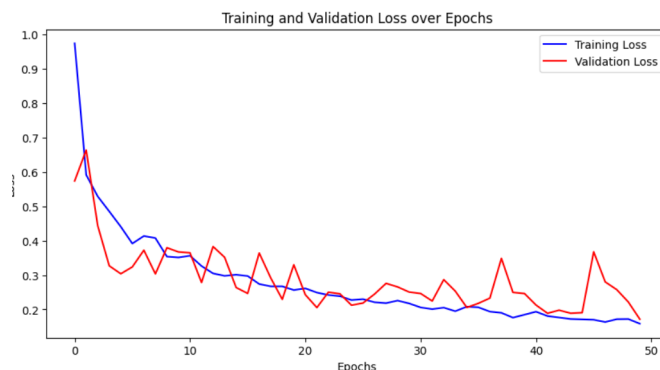


FIGURE 2 – Courbe de Validation loss et Training loss

On peut identifier des patterns qui peuvent expliquer les erreurs de classifications. Par exemple, les images ayant le label 6, soit PermanentCrop, sont caractérisés par des divisions bien marquées entre des parcelles de champ qui sont de forme rectangulaire, donc facile à caractériser pour un modèle comme celui-ci.

Néanmoins, on peut voir sur les images dont les prédictions sont fausses qu'elles ne présentent pas ces divisions, ou alors elles sont très peu marquées, ce qui perturbe le modèle.

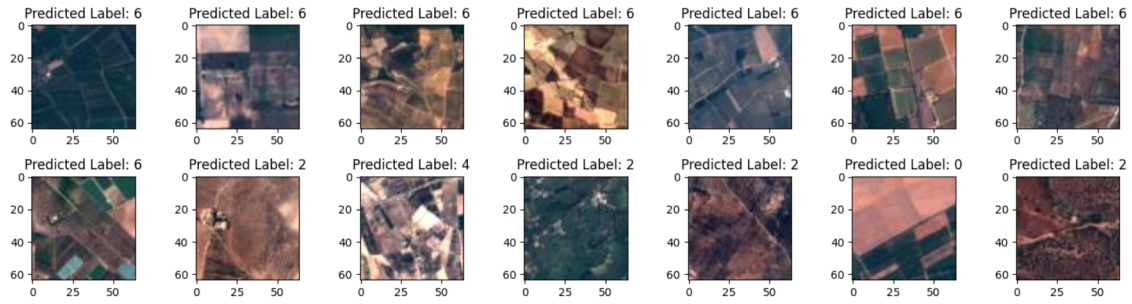


FIGURE 3 – Exemples d’images ayant pour vrai label le label 6

8 Conclusion

On a donc utilisé un modèle ResNet50 pour un problème de classification d’images satellites à 10 classes et on est parvenu, grâce à un choix optimisé des hyperparamètres et de l’augmentation de données, à avoir une précision avoisinant les 96%, ce qui est très satisfaisant au vu de la simplicité à priori du modèle.

Néanmoins, ce modèle pourrait encore être amélioré pour mieux tenir compte de classes qui sont difficiles à distinguer.