Land Cover Classification

EI - PNT - Observing the Earth

Dan Benayoun, Alexandre Gars, Martin Potier
Lucas Da Silva Rocha, Louis-Ferdinand Barbotin

February 2, 2024

CentraleSupelec

Introduction

- Provides a better understanding of land utilization aspect
- Crucial role in formation of policies regarding development planning
- Very powerful intelligence tool

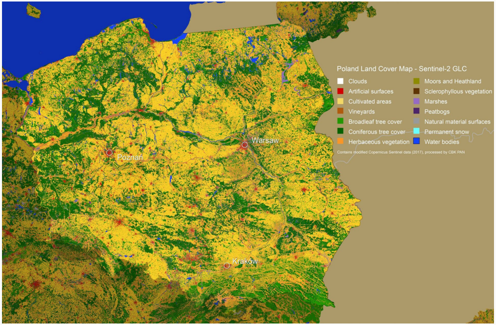


Figure 1: Land Cover of Sentinel 2 data

Table des Matières

Predict land coverage with spectral indexes

Spectral indexes usage

- Leverage spectral response of most common structures
- Often engineered to be specific to one type of structure

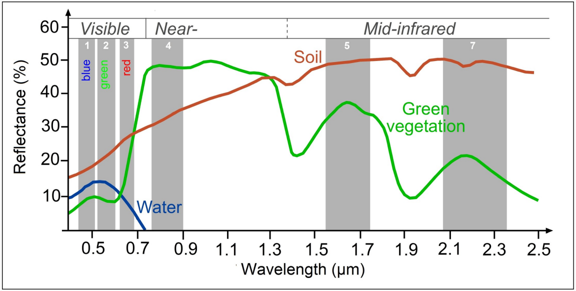


Figure 2: Spectral signature of several common structures

Most common spectral indexes

Index	Equation Using Landsat 8 OLI [35]
Normalized Difference Vegetation Index (NDVI)	$\frac{\text{NIR}-\text{Red}}{\text{NIR}+\text{Red}}$
Enhanced Vegetation Index (EVI)	$2.5 \times \frac{\text{NIR}-\text{Red}}{\text{NIR}+6\times\text{Red}-7.5\times\text{Blue}+1}$
Normalized Difference Water Index (NDWI)	$\frac{(\text{SWIR}_1+\text{Red})+(\text{NIR}+\text{Blue})}{(\text{SWIR}_1+\text{Red})-(\text{NIR}+\text{Blue})}$
Normalized Difference Moisture Index (NDMI)	$\frac{\text{NIR}-\text{SWIR}_1}{\text{NIR}+\text{SWIR}_1}$
Bare Soil Index (BSI)	$\frac{\text{Green}+\text{NIR}}{\text{Green}-\text{NIR}}$

Figure 3: Example of indexes

- Index formula depends on bands availiable

Most common spectral indexes

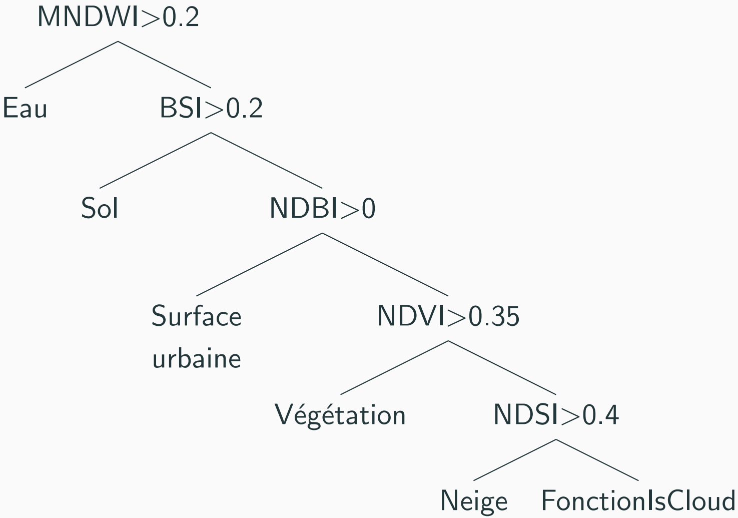


Figure 4: Decision Tree for Land Cover Classification

Example

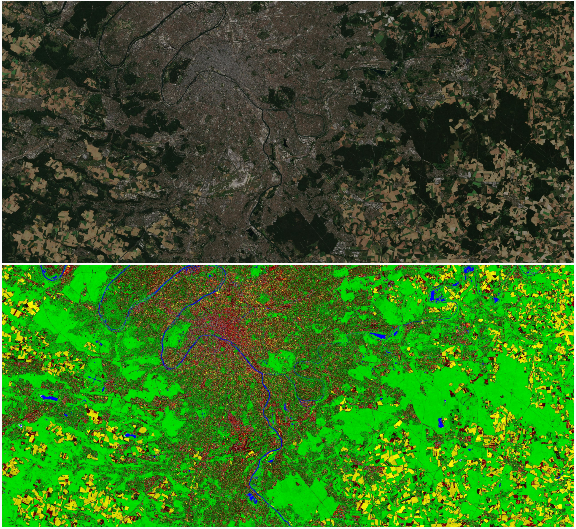


Figure 5: Ile de France

Example

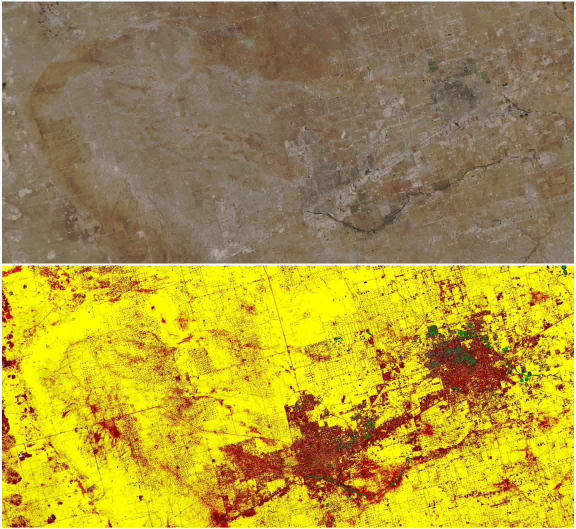


Figure 6: Texas

Example

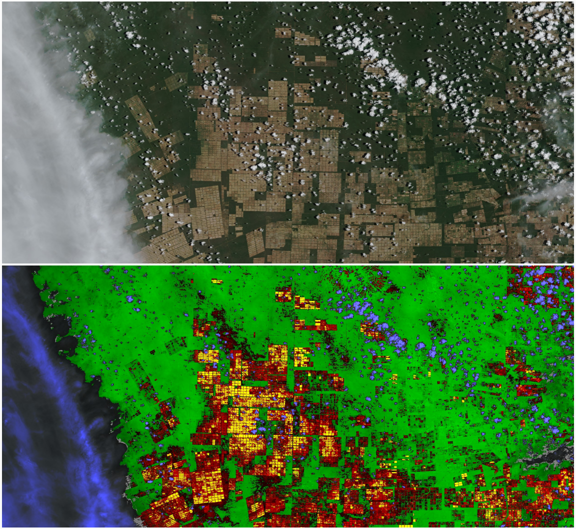


Figure 7: Manaus

Example

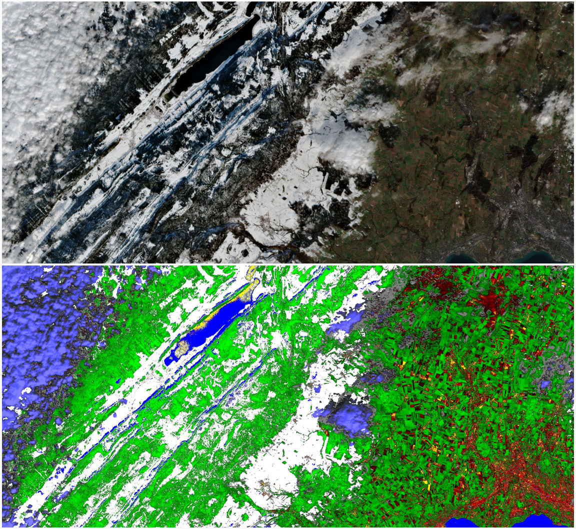


Figure 8: Lausanne

Predict land coverage with AI

K Mean

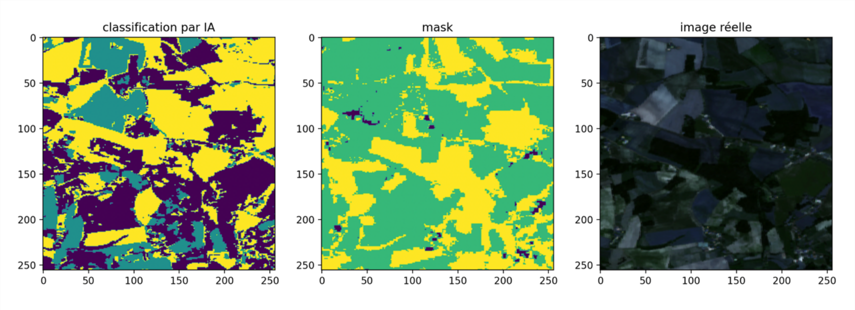


Figure 9: Acceptable but could be improved

K Mean Detection Issue

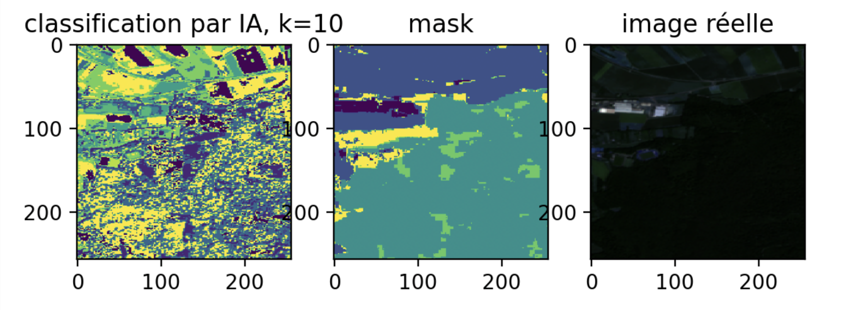


Figure 10: Cloud detection issue

K Mean Revized

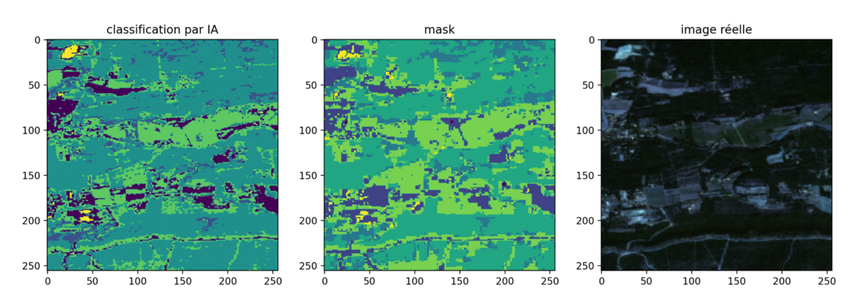


Figure 11: K Mean Applied to all the pixel within the dataset

Convolutional Neural Network

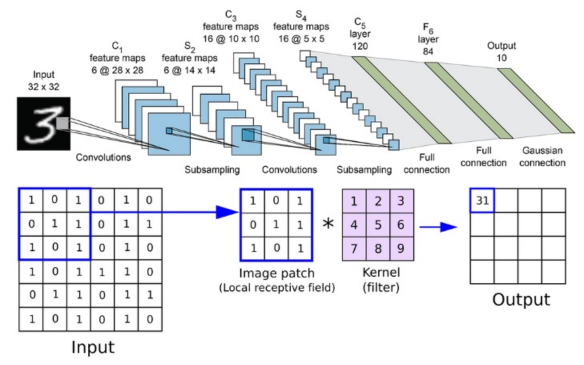


Figure 12: Illustration of the CNN Architecture

Encoder Decoder Architecture

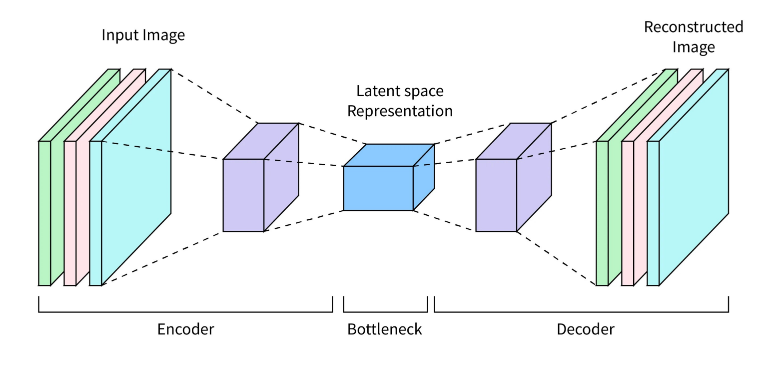


Figure 13: Illustration of the Encoder-Decoder Architecture

U-Net Architecture

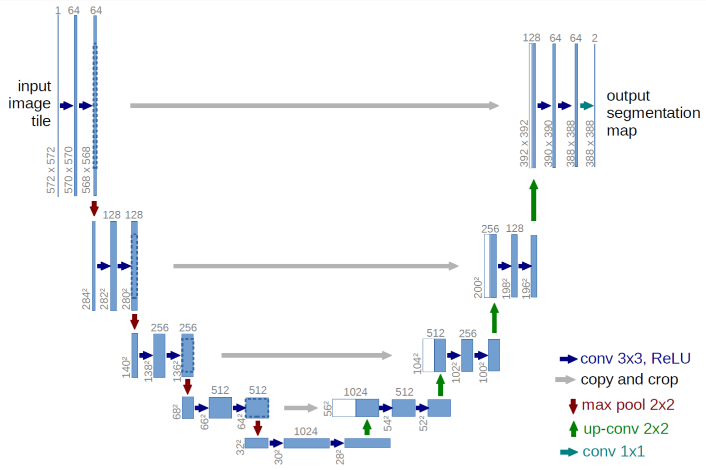


Figure 14: Illustration of the U-Net Architecture

DeepLabv3 Architecture

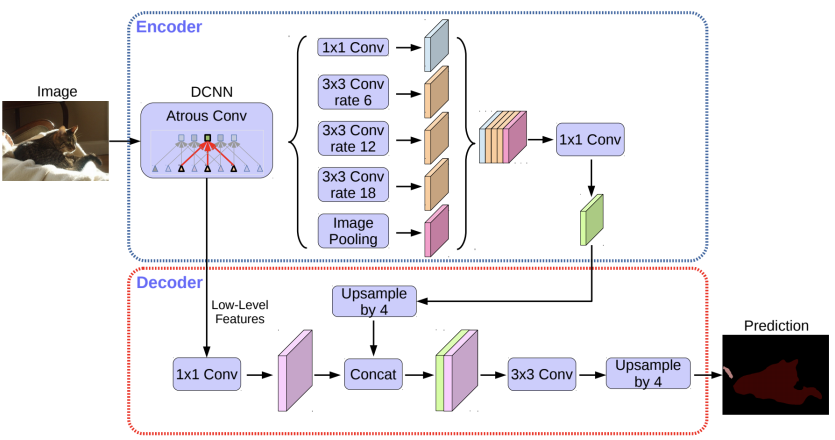
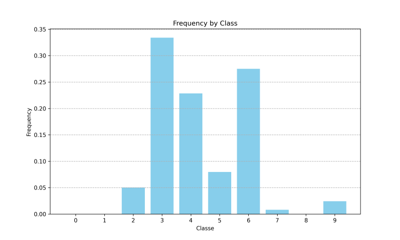


Figure 15: Illustration of the DeepLabv3 Architecture

Pre Processing and implementation

- PyTorch allows for easy implementation and customization (easy to adapt model to 4 channels)
- Pre Process data for better handling (normalization, frequencies...)
- Leverage Kaggle computing power by using CUDA

(a) Weight by class (log scale)



(b) Frequency by class

Figure 16: Class imbalance in Pre Processing

Training Logic

```python
# High-level overview of the deep learning training and validation process

# Initialize model, loss, and optimizer
model = initialize_model()
criterion = define_loss_function()
optimizer = define_optimizer(model.parameters())

# Training loop
for epoch in range(total_epochs):
    train_loss = perform_training_epoch(model, criterion, optimizer, train_data)

    # Validation loop
    validation_results = perform_validation_epoch(model, criterion, val_data)

    # Log results
    log_epoch_results(epoch, train_loss, validation_results)

    # Save model checkpoint if necessary
    save_checkpoint_if_required(model, epoch)

# Final evaluation on the test set
test_results = evaluate_model_on_test_set(model, test_data)

# Save final model
save_final_model(model)
here
```

snappify.com

Loss Function

Generalized Dice Loss (GDL):

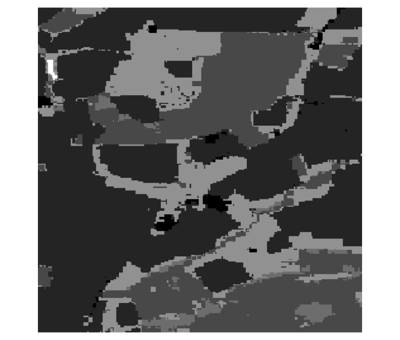
$$GDL = 1 - 2 \cdot \frac{\sum_{l=1}^{2} w_l \sum_n r_{ln} p_{ln}}{\sum_{l=1}^{2} w_l \sum_n r_{ln} + p_{ln}},$$

Weighted cross-entropy (WCE):

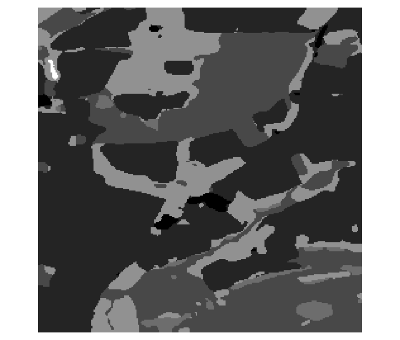
$$WCE = -\frac{1}{N} \sum_{n=1}^{N} w r_n \log(p_n) + (1 - r_n) \log(1 - p_n),$$

Combination of both to take into account class imbalance

Vizualisation



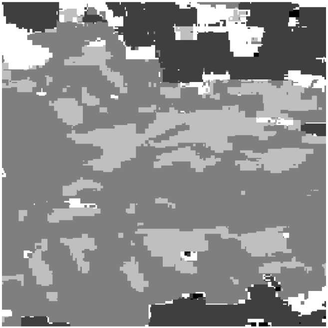
(a) Thruthground



(b) Predicted Mask

Figure 17: U Net

Vizualisation



(a) Thruthground

(b) Predicted Mask

Figure 18: DeepLabv3

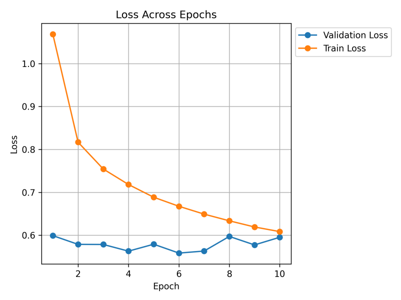
Results

Results K Mean

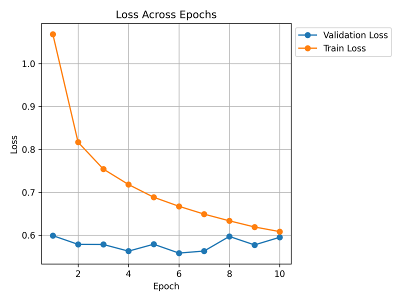
En %	Water	Agricultural	Urban	Total
Water	0.9	0.1	0	1
Agricultural	0	74	2	76
Urban	0	2.5	12	15
Total	0.9	76.6	14.5	92

Figure 19: Confusion Matrix

Results Deep Learning : Loss



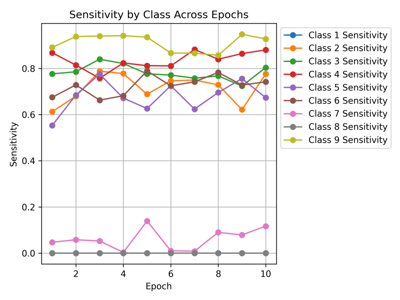
DeepLabv3



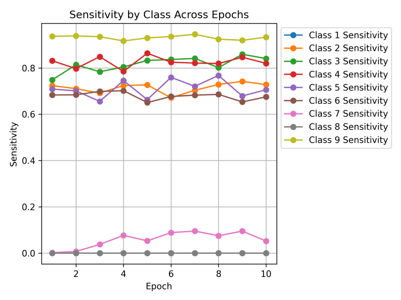
U Net

Figure 20: Loss

Results Deep Learning : Sensitivity



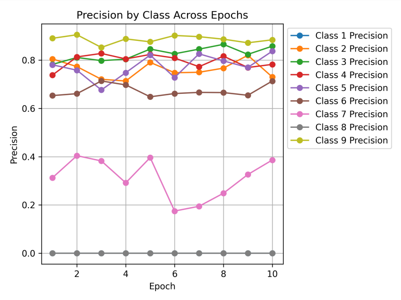
DeepLabv3



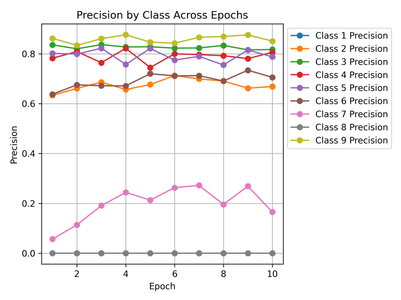
U Net

Figure 21: Sensitivity

Results Deep Learning : Precision



DeepLabv3



U Net

Figure 22: Precision

Conclusion