Deep Learning and Remote Sensing

Romain Wenger

M2 OTG/Geot - 3A EOST :
AI Geosciences

16 and 17 January 2025

Course Plan

Table of Contents

- The amount of acquired data is increasing exponentially
- Diverse data sources :
  - Multi-source, multi-modal, multi-temporal satellite imagery (optical/radar/hyperspectral)
  - 3D Data (LiDAR)
  - Participatory data
  - ... And many more

→Knowledge extraction from data acquired by different types of sensors

Table of Contents

Table of Contents

Vaihingen dataset

Vaihingen dataset containing 7 land cover classes for 33 orthophotos. (Adapted from the work of Audebert et al., 2018, `https://www.sciencedirect.com/science/article/pii/S0924271617301818`)

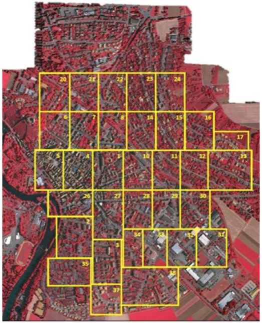
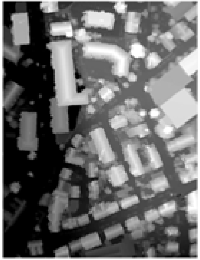


FIGURE – Tiling of the orthophotos for the creation of the dataset

Vaihingen dataset

It contains triplets with (a) an IRRG (Infra-Red, Red, Green) orthophoto, (b) a DSM (Digital Surface Model), and (c) the reference data (I don't like the term *ground truth* :D)


(a)


(b)


(c)

Table of Contents

Objectives

Use a semantic segmentation network (U-Net) for the classification of orthophotos into 7 land cover classes (Adapted from the work of Audebert et al., 2018, `https://www.sciencedirect.com/science/article/pii/S0924271617301818`)

Bonus : Combine the orthophotos with the DSM and/or the NDVI (Normalized Difference Vegetation Index) and demonstrate the improvement (or decline) in classification results.

$$NDVI = \frac{NIR - Red}{NIR + Red}$$

Table of Contents

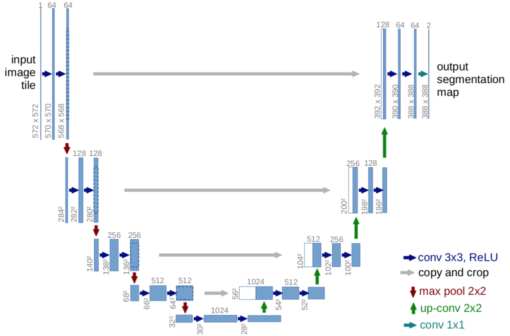


FIGURE – U-Net Network (Ronneberger et al., 2015 https://arxiv.org/abs/1505.04597)

How does convolutions work ?

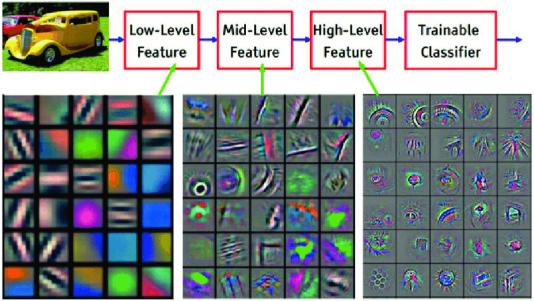


FIGURE – Example of features that the filters in a convolution layer look for at different levels in a network. The deeper into the network (higher level), the more complex the features are. Source : (F.-F. Li & Karpathy, 2015)

DL Workflow

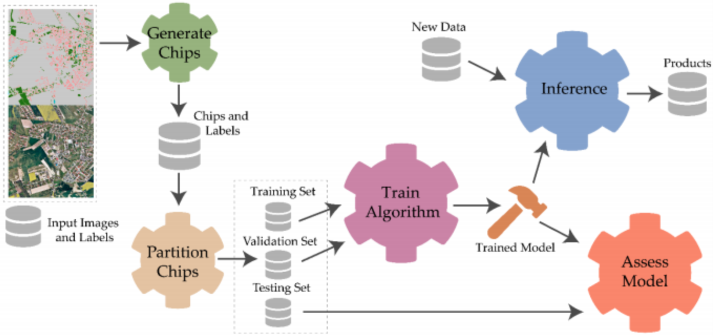


FIGURE – Standard DL Workflow (Maxwell et al., 2021 `https://www.mdpi.com/2072-4292/13/13/2591`)

Image Patches and GPU Efficiency

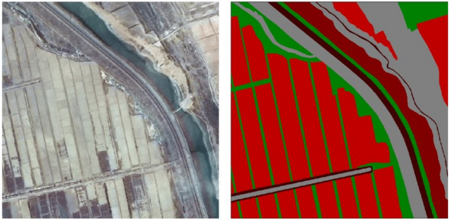

(a) Images (b) Ground truth

FIGURE – Cutting images into smaller patches.

Image Patches and GPU Efficiency

You need to use small subset of your image !

- **Memory Efficiency :** Processing entire high-resolution images can be memory intensive. Splitting them into smaller patches allows training on standard GPUs.
- **Improved Generalization :** Using patches increases the diversity of the training dataset, helping the model generalize better across different regions.
- **Focus on Local Context :** Small patches help the model focus on local structures and textures, essential for tasks like semantic segmentation.
- **Data Augmentation :** Patches can be easily rotated, flipped, or cropped to create variations, enriching the dataset.

How to cut your patches ?

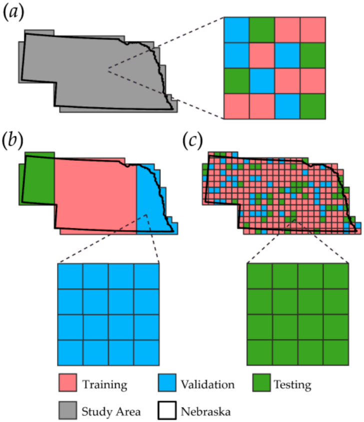


FIGURE – Sampling and patch selection. (a) Random sampling across the entire study area, (b) Geographic stratification, (c) Random sub-tiling and assignment of a category to each sub-tile (Maxwell et al., 2021 `https://www.mdpi.com/2072-4292/13/13/2591`)

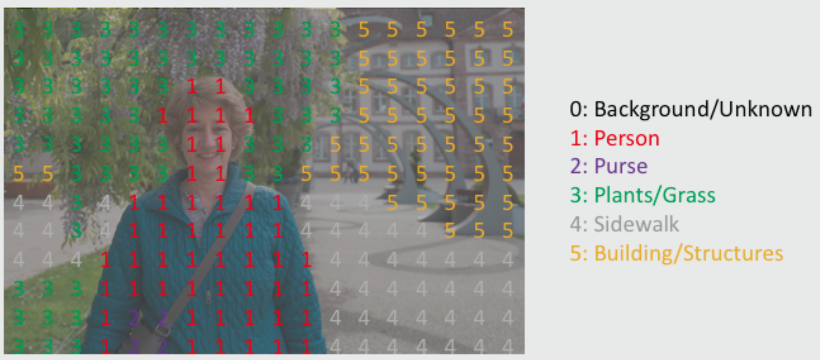


FIGURE – Reference data representation in semantic segmentation.

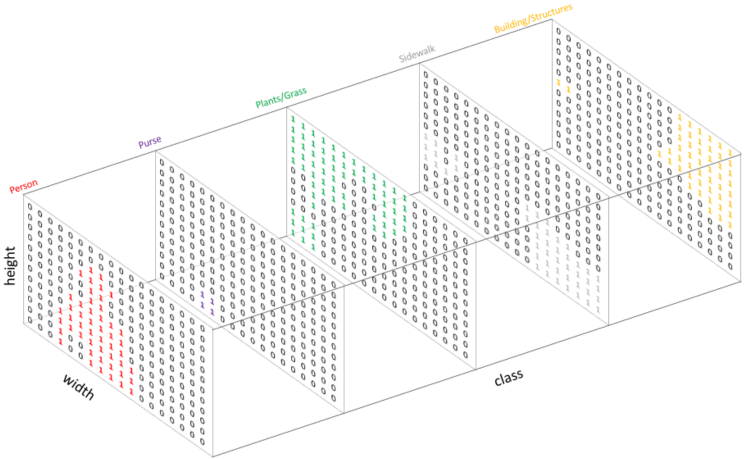


FIGURE – Formatting of reference data using a *One Hot Encoder* for semantic segmentation approaches.

Softmax Function

Definition : The softmax function transforms a vector of values into a probability distribution.

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

- **Probability Distribution :** Each output is in the range (0,1) and sums to 1.
- **Application :** Commonly used in the final layer of classification models to predict class probabilities.
- **Example :** Useful (even needed) in multi-class classification tasks.

Sigmoid Function

Definition : The sigmoid function maps any real-valued number to a value between 0 and 1.

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

- **Range :** Outputs values between 0 and 1.
- **Application :** Often used in binary classification problems.
- **Gradient Behavior :** Can lead to vanishing gradients for large positive or negative inputs.

From Probabilities to Class Prediction

How to choose a class from probabilities :

- For the **softmax**, the class with the highest probability is selected :

  $$\text{Class} = \arg\max(\sigma(z))$$

- For the **sigmoid** in binary classification, a threshold (commonly 0.5) is used :

  $$\text{Class} = \begin{cases} 1 & \text{if } \sigma(z) \geq 0.5 \\ 0 & \text{if } \sigma(z) < 0.5 \end{cases}$$

Note : The choice of the threshold can affect model sensitivity and specificity.

Model evaluation

Predicted Classes

		p	**n**	**Total**
Actual Classes	**p′**	True Positives	False Negatives	P′
	n′	False Positives	True Negatives	N′
	Total	P	N	

TABLE – Theoretical confusion matrix for two classes

Metrics

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1_{Score} = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times TP}{2 \times TP + FN + FP}$$

where TP = True Positive, FP = False Positive, TN = True Negative, and FN = False Negative.

Table of Contents

Important Information

- Import the notebook **TD_DL_2025.ipynb** into your Google Colab session : `https://github.com/r-wenger/Courses_Master`
- The practical session consists of incomplete code snippets. Carefully read the code and comments to understand what you need to do.
- Click on the links in the notebook to understand certain concepts covered.
- You will find "Hidden Methods" tabs. These functions can be complex, and you will need to run the cell for the code to work. If you're curious, feel free to explore them.
- **Do not focus** on the network (UNet) but on how to give your data to the network !