Toward Supervised Classification of Natural Habitats from Very High Resolution Aerial Images with Convolutional Neural Networks

Bertille Temple

Master Thesis Report
April-September 2024
Montpellier, France

Laboratory
Tetis Unity
Evergreen Inria Team
Inrae Fundings

Supervisors
Dino Ienco (Inria)
Cassio Fraga Dantas (Inria)
Diego Marcos (Inria)
Samuel Alleaume (Inrae)

Academic Advisor
Gabriele Facciolo (Borelli)

Ecomed Partners:
Pierre Volte
Marie Pisson
Léo Nery

School
Master Mathematics, Vision, Learning (MVA)
Ecole Normale Supérieure Paris-Saclay

Contents

Acknowledgements

I would like to thank my master thesis supervisors for their regular mentoring and the knowledge they shared with kindness and patience: Dino Ienco, Cassio Fraga Dantas, Diego Marcos, and Samuel Alleaume.

I also wish to express my gratitude to the ECOMED members for their goodwill and availability: Pierre Volte, Marie Pisson, and Léo Nery.

Special thanks to my office mate Malo Desbois for our constructive daily discussions on methodological questions, and to Ananthu Aniraj for his explanations on GPU usage and experiences with SAM. Additionally, I appreciate Christopher Jabeu for our talks about metrics and code, César Leblanc for his generous time explaining his thesis, Quentin Yeche for his advice on data augmentation and SAM, and Matthieu Fauvel for his recommendations on t-SNE. Thanks also to Gabriele Facciolo for his feedback on how to value my work.

Finally, I want to acknowledge all the other colleagues who contributed to making the office environment pleasant: Mireille, Simonna, Pallavi, Eattidal, Pauline, Youssef, Fouzia, Roger, Reno, Akim, Roberto, Raffaele, and many others.

Abstract

This internship took place at the Maison de la Télédétection (MTD) in Montpellier(France), mainly under the supervision of researchers from the EVERGREEN team from Inria, a new branch linked to Sophia Antipolis, with additional guidance from an INRAE supervisor. The EVERGREEN team focuses on advanced Machine Learning(ML) techniques for Earth observation data to address agro-environmental challenges. EVERGREEN, INRAE, along with CIRAD, IRD and AgroParisTech are gathered in the TETIS Joint Research Unit, which is interdisciplinary, working on spatial information for territorial complexity and agro-ecosystems. The internship was funded by INRAE, with data provided by ECOMED, an ecological consultancy focused on biodiversity preservation. This master thesis represents a step toward natural habitats classification with Convolutional Neural Networks (CNN) from aerial images in a supervised setting. Using labelled data provided by ECOMED, the report aims to reveal the dataset's potential and limits while designing pathways to overcome these limits.

1 Introduction

1.1 Motivation

The European Environment Agency's (EEA) latest 'State of Nature in the EU 2020'[1] assessment reveals that 81% of protected habitats are in poor or bad condition, putting them at significant risk. The EU Biodiversity Strategy for 2030 highlights the importance of protecting these habitats, with mapping being a key step in achieving this goal. Automating the mapping of these habitats would allow to cover more habitats and do so more frequently, thus improving the ability to monitor them. For habitat mapping, the European Nature Information System (EUNIS) developed by the European Environment Agency (EEA) and the European Environment Information and Observation Network (EIONET) provides a detailed classification of natural habitats harmonized at an European level, with each primary level subdivided into up to seven lower levels. It includes detailed classes such as: "Western nemoral river bank tall-herb communities dominated by meadowsweet", "Thermo-Mediterranean Jupiter's beard brushes" and "Retuse torgrass swards".[In French: "Mégaphorbiaies occidentales némorales rivulaires dominées par Filipendula", "Fourrés thermoméditerranéens à Anthyllide barbe de Jupiter", et "Pelouses à Brachypode rameux"]. According to EUNIS, a natural habitat is an area characterized by its physical attributes, such as soil acidity, soil humidity, plant physiognomy and the species of plants and animals it supports [22]. Plant physiognomy can be defined as the appearance and architecture of a plant community, with characteristics such as its height and density. A habitat differs from an ecosystem and a biome by its scale, with biomes being the largest, followed by ecosystems, and then habitats. Habitats are sometimes called biotopes, but the distinction lies in the focus: a biotope emphasizes abiotic factors, while a habitat includes plants, animals, fungi, and their interactions into the definition. EUNIS classification is used by various actors in ecology, such as the ecological consultancy firm ECOMED, based in Montpellier.

1.2 Context

ECOMED, short for ECOlogie et MEDiation, was established in 2003 and is specialized in land use planning, ecosystem conservation, and environmental impact assessments. For the duration of the master thesis, it provides access to a valuable dataset consisting of aerial images of Very High Resolution (VHR) and polygons annotated by ecological experts, identifying natural habitats according to the EUNIS classification. Given the availability of Remote Sensing(RS) images, this thesis treats automatic habitat mapping as a Land Cover(LC) classification problem. This approach is supported by the FAO's definition of LC as "the observed (bio)physical cover on the Earth's surface," which can include natural habitats.

1.3 Methodological Background

Before the arrival of Convolutional Neural Networks (CNNs), LC classification was performed by extracting handcrafted features such as spectral and textural information and then using various ML algorithms such as Random Forest (RF) [27] or Support Vector Machines (SVM) [33] using these handcrafted features. Advances in parallel computing and the availability of RS data have enabled the use of CNNs, which were already theoretically developed back in the 1980s [9]. CNNs have the advantage of automatically extracting features without the need for manual design. Coupled with backpropagation [19], CNNs have outperformed traditional handcrafted methods, demonstrating superior performance in LC mapping [24].

1.4 Problem Statement

While CNNs have proven efficient in many LC applications, such as settlement mapping [10] [3], forest area mapping [25] [31], crop type mapping [18], they have not yet been investigated in the context of natural habitat mapping, to our knowledge. The key question addressed in this report is: What is the potential

[1] url_state_nature_EU_2020

of supervised Deep Learning(DL) methods in mapping natural habitats using VHR aerial images? First, an analysis of the data will be conducted to define the project's scope, and the pre-processing steps will be detailed. Then, we will explore the methods and experiments involving pixel classification, patch classification, and data shift mitigation. Finally, we will review the limitations of our current approach and dataset, and recommend next steps, including expanding experiments and investigating alternative methods and data sources.

2 Data Analysis and Pre-processing

In this section, we first explore the data which are made of raster images and vector polygons. Next, we analyze the EUNIS classification along with the expert annotation process and adjust the classification based on this analysis. After that, the pre-processing pipeline is described, starting with the initial step of dataset curation, followed by the second step of data structure modifications for DL processing.

2.1 Data Exploration

2.1.1 Raster images

The tif images provided are ordered by folder zones, with 173 geographical zones from the South of France. Images were captured in 3 days between June and August 2023 using a drone-mounted camera. Each zone is covered by 1 to 3 images in general, as shown in the box plot from Figure 1.

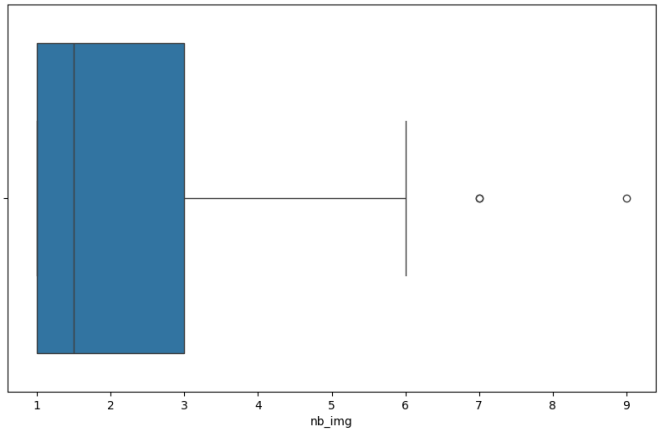

Figure 1: Box plot of number of images per zone

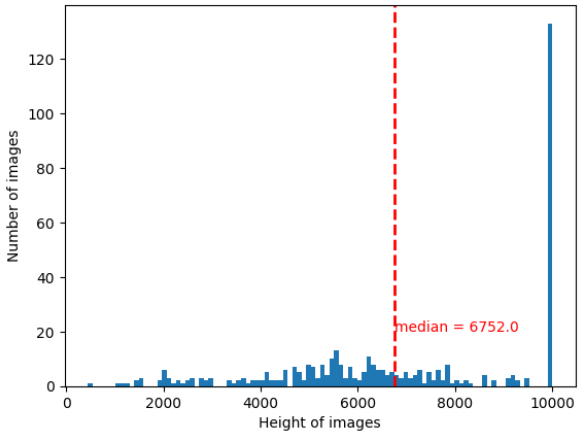

Figure 2: Histogram of height per image

Figure 2 illustrates that images have varying size, with a height between 500 and 10000 pixels. For automating processing, a fix patch size must be chosen. Each image has 4 channels: Red(R), Green(G), Blue(B) and Near InfraRed(NIR) and a resolution of 15 cm per pixel.

2.1.2 Vector polygons

Polygons Attributes Annotated polygons are stored in a vector file as a shapefile, each polygon having 43 attributes. Some of the most interesting are introduced below.

- CDEUNIS_1: The EUNIS code at its most detailed level, made of an initial letter followed by a sequence of integers. The first letter represents the primary level class, the first integer indicates the second level class, and this sequence continues through as many hierarchical levels as required, potentially up to seven. This is the most important field used to supervise the model training,

validation and testing. There are also CDEUNIS_2, and 3 when the polygon is labelled with differents habitats. An example of item at different level of granularity can be found in Figure 3.

- ID: An identifier assigned to each polygon that is not unique. This will be addressed in the Dataset Curation subsection 2.3.1.
- TXRECOUV_1(=recovery rate): Percentage of coverage for class 1 when the polygon is multi-labeled. Half of these polygons have 50% coverage, while the coverage for the other half is unknown.
- LBPYSIO(=physiognomy label): Comments from experts detailing sometimes the EUNIS label, sometimes a recap of what they saw (for eg which plants). This data is not harmonized, making it challenging to process as a feature.
- ENJEU(=issue): Out of 8,271 polygons, 2,845 have this column filled. The items can be categorized into two main groups. The first group consists of adjective-based priority levels, such as Weak("Faible"), Null("Nul"), Very weak ("Très faible", with encoding issues affecting accents), Moderated("Modéré", also affected by encoding issues), and High("Fort"). The second group includes more than 15 out of 20 possible values for this field, which are descriptive habitat categories that are less structured, such as Habitat considered wet (in French: "Habitat considéré comme humide"). Given the predominance of less structured descriptive categories in the second group, it is not straightforward to make a profit of this field. 130 polygons are categorized as high issue, more than half of these 130 polygons are classified as "Tyrrhenian ash-alder galleries forest"[in French:"Forêts galeries tyrrhéniennes à Frêne et Aulne"]. The median surface area of these polygons is 680 m^2, approximately 170 by 170 pixels. So the few polygons marked as high interest require a patch size larger than 170 by 170 pixels to be fully contained within a single patch, assuming the shape of the wooded areas is roughly square. This approximation will be taken into account when selecting patch sizes in the Dataset Pre-processing subsection 2.3.2

Figure 3: Example of EUNIS granularity

Polygons Statistics The pie plot from Figure 4 shows that 27% of the polygons are multi-labels, it happens when a polygon is covered by a mix of different habitats. Moreover, half of the polygons were annotated before 2023 as shown in Figure 5. Images are from Summer 2023, so at least half of the polygons were not annotated considering the relative images. This might result in a mismatch between the annotations and the provided images if the habitats on the ground had changed between the time of annotation and the summer of 2023. That being said, the earlier annotations were made in 2019, and experts believe that a period of less than five years between annotations and image capture is reasonable, as habitats usually do not evolve quickly.

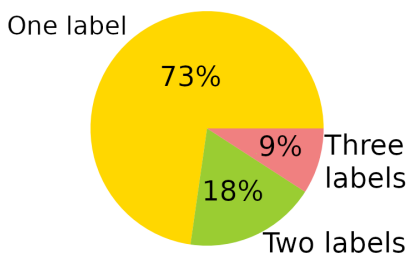

Figure 4: Proportion of areas with 1, 2 and 3 labels

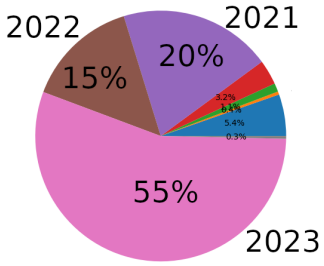

Figure 5: Proportion of annotated surface by year

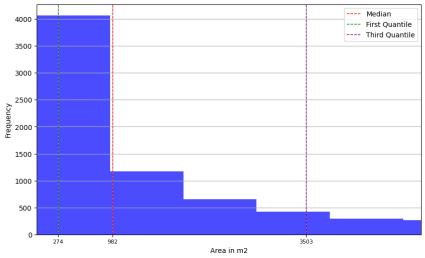

Figure 6: Histogram of polygon areas

The histogram in Figure 6 reveals that the polygon surface distribution is skewed to the left, ranging from 0 to 952,463 m^2, with a median area of 982 m^2. The 25th percentile is 274 m^2, and the 75th percentile is 3,503 m^2, highlighting big variation in polygon surfaces. This variation make more complex the task of choosing the relevant patch size. As it was detailed in the field "ENJEU" from subsection 2.1.2, half of the polygons with high interest have a medium area of 680 m^2, which means there are among the small polygons of the dataset. ECOMED experts confirmed that smaller polygons are usually more important.

The boundaries between polygons of different classes, especially those differentiated only by their physiognomy (e.g., the transition from a matorral to a forest), are not straightforward because the transition between two habitats with the same species but at different stages can be smooth and gradual. This can lead to significant variability in polygons delimitation across experts. To recap, the polygons have varying sizes and polygons delimitation are noisy.

2.2 Classification

2.2.1 Levels Considered

Before trying to automate the classification of habitat cover with CNNs, the feasibility of the task must be evaluated. This requires a look into the annotation process conducted by experts. At ECOMED, natural habitat mapping involves 2 tasks: polygons delimitation and polygons classification.

Polygon Delimitation Experts usually go in the field, identify the areas of habitats easily identifiable from the sky, which they mark by a single geo located point on their QGIS based application. Areas that cannot be distinguished from aerial images are delimited with multiple geolocated points along their boundaries from the field. This highlights that some habitats cannot be delimited using only aerial imagery, particularly when two habitats are distinct but visually similar from above. Consequently, these types of habitats are unlikely to be delimited by applying CNNs to aerial images alone, according to experts estimation, it represents less than 5% of the polygons. Back to the office, each polygon is properly delimited on QGIS generally using Google Earth satellites images, and classified.

Polygon Classification In practice the plant specie and the vegetation physiognomy seen in the ground are the 2 most discriminant features to classify the habitat, but the dataset does not systematically include structured information on plant species nor the physiognomy. Also this point is important: while a CNN might infer vegetation height and density, it seems unrealistic for it to identify plant species on the ground only from an aerial image. Without access to plant names as an independent structured field, or to more data to infer plant specie, the challenge of classifying natural habitats at a fine level becomes very hard. However, classifying habitats at the coarsest level of EUNIS classification might be feasible because an aerial picture alone can provide enough details to distinguish among classes at this level. For example, it is not necessary to know the exact plants specie to delimit a polygon and identify it as "forest", "agricole", "artificial", "meadows" etc.

Recap Our focus will be restricted to coarse level 1 and level 2 classes of EUNIS classification, which are introduced in Figure 7 and 8. It is important to keep in mind that the error model for boundaries might be affected by areas that are challenging to delineate from aerial images, specifically when the expert's visit on the ground was necessary to determine the boundary.

2.2.2 Classes Restriction

Barplot at the left in Figure 7 shows that the dataset is highly imbalanced at level 1. Barplot at the right of the same figure illustrates how the underrepresented classes were combined into an "Other" class to ensure sufficient samples for each class and to prevent the model from being excessively biased towards the dominant class.

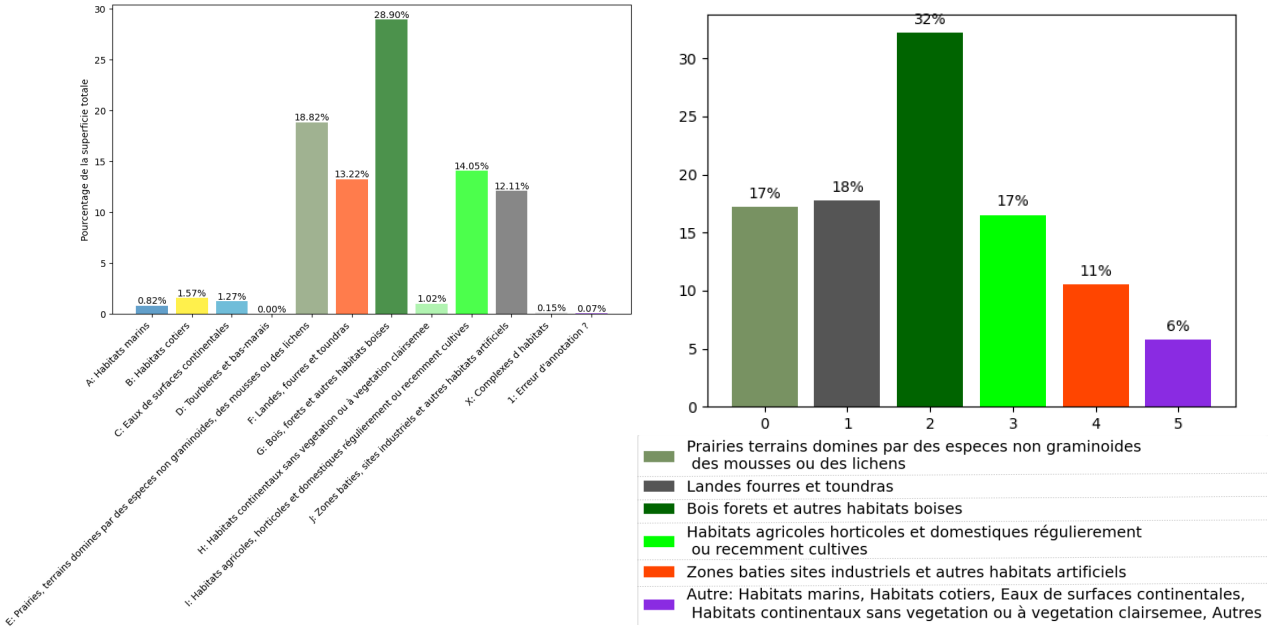

Figure 7: Percentage of area for each class at level 1. Left: before grouping under-represented classes. Right: after grouping

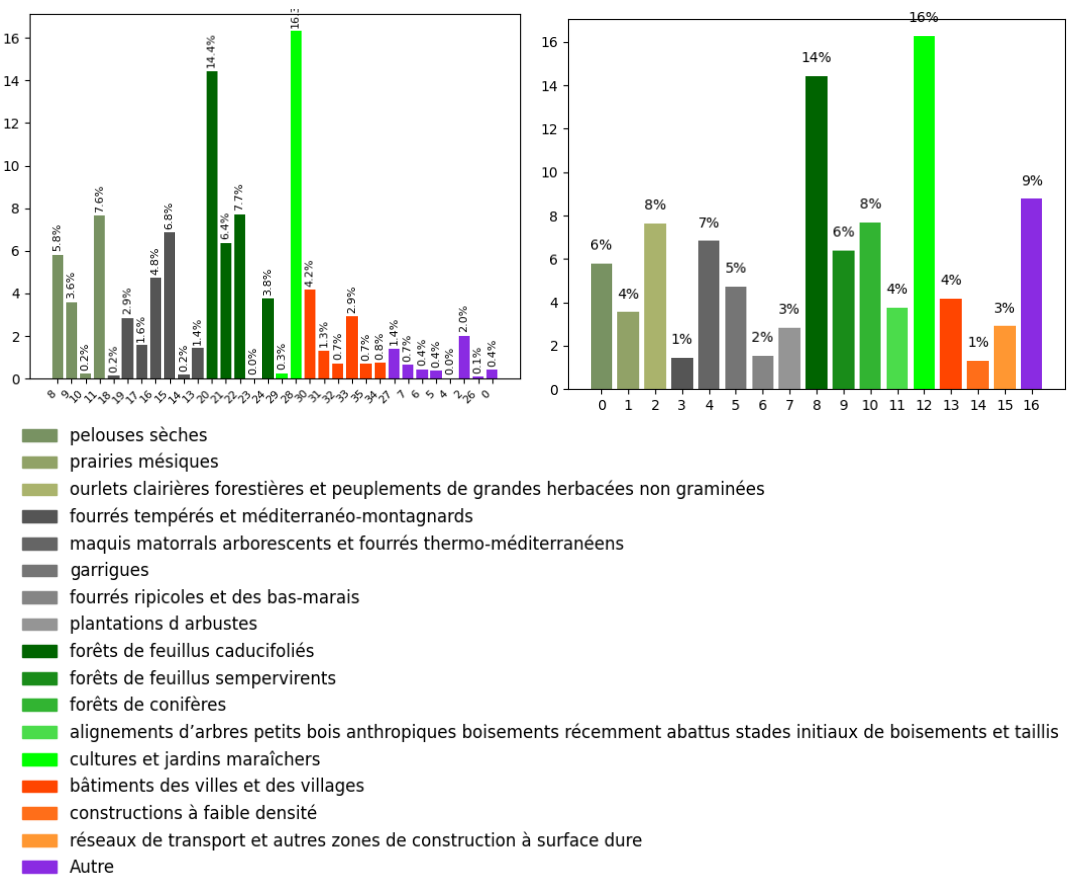

Figure 8: Percentage of area for each class at level 2. Left: before grouping under-represented classes. Right: after grouping

Figure 8 illustrates the same grouping process at level 2 to go from 34 to 17 classes.

In this subsection on classification, we have (1) restricted the classes to level 1 and level 2 based on the available data and expert annotation protocol, and (2) analyzed the class distribution at these levels and balanced the classes. We can now proceed to the next subsection, which covers the preprocessing pipeline.

2.3 Pre-processing

With the recent trend of DL, driven by impressive tools such as ChatGPT and fake image generation with GANs, many companies, public institutes, and laboratories want to benefit from their datasets, which were not originally created for automatic processing, by applying DL techniques. The dataset we have is one of them, as future automatic processing of the dataset was not considered in its initial design. This paragraph provides key steps to bridge the gap between the expected format for DL processing and the format provided by default from the ecological dataset we have. While this dataset is from a specific field, we believe that these points are generalizable to datasets covering other fields since we think that the problems encountered are not field-specific.
This section is split into two parts. First, in the "Dataset Curation for Automatic Processing" subsection, we will discuss the considerations for refining the dataset to ensure it is suitable for automatic processing. This includes addressing issues related to consistency, standard format, and completeness. Second, in the "Dataset Pre-processing for DL" subsection, we will cover the specific pre-processing steps required to prepare the dataset for DL applications. This involves steps such as rasterization and patch splitting.

2.3.1 Dataset Curation for Automatic Processing

The following paragraphs detail the key guidelines in dataset curation and highlight areas where further information may be required.

Key Guidelines

- **Correctness:** Verify that all data values are accurate and correspond to their expected values.
- **Uniform Formatting:** Standardize the formatting of data across the dataset. This includes ensuring that identifiers, dates, and code labels follow a consistent format. For example, compile a list of acceptable values for each field and verify that all entries in the dataset conform to these values. Quantify the extent of formatting inconsistencies and apply necessary corrections or filters to standardize the data.
- **Completeness:** Check for completeness by addressing missing or mismatched data. Quantify missing data and implement measures to fill gaps or remove incomplete entries.
- **Identifiers:** Ensure that all identifiers are unique and properly defined to avoid ambiguity.

In our case, a correctness issue was identified with the TX_RECOUV field, which indicates the coverage rate of the first class specified in the CDEUNIS_1 field. Typically, this field is set to 100% for single-labeled polygons. However, we observed that some single-labeled polygons have TX_RECOUV set to 0 instead of 100%. Additionally, over 50% of multi-labeled polygons also have this field set to 0, which is problematic because, in this context, a value of 0 represents "Unknown". Having a field where a value of 0 indicates either no coverage or full coverage is inconsistent. At the end, TX_RECOUV field was excluded from the study.
For formatting issues, only around 5 badly formatted codes and date were encountered, manual corrections were made.
Regarding completeness, missing data were filtered out, including polygons with missing codes, incomplete codes, or those lacking corresponding tif images. Specifically, 20% of polygons were missing associated tif images, and 45% of .tif images did not have corresponding polygons. Additionally, 2% of polygons were unannotated and 2% had invalid geometries; these were filtered out as well. These issues were reported to the data source ECOMED for future improvement of the dataset.
To address the issue of non-unique identifiers mentioned in Section 2.1.2, new unique identifiers were generated in the original order before any modifications. This step ensures that the IDs can be recreated by anyone using the same original dataset.

Need for Additional Information It might be necessary to request or obtain additional information if it is not present:

- **Temporal Metadata:** Request dates of annotation in the dataset.
- **Acquisition Parameters:** Seek detailed parameters for image acquisition.
- **Documentation:** Obtain complete documentation explaining the data folder structure, the contents of subfolders, and the content of each file.

For temporal metadata, relevant information was found in an Excel file provided together with the data. We matched each polygon with its corresponding image by identifying which image intersects with the polygon's bounding box. Each image was already associated with a zone, as they are organized into subfolders named by zone index. These zones were connected to their annotation dates in the Excel file. By linking polygons to images, images to zones, and zones to annotation dates, we were able to finally

associate each polygon with its annotation date.
Acquisition parameters, including the date and the platform used to capture the images, were obtained during a meeting with the data provider.
Documentation is still pending but could be informed by material from this master's thesis.

2.3.2 Dataset Pre-processing for DL

Step 1: Filtering and Matching Each polygon is linked to an image when their bounding boxes intersect. A pivot table is created to map each polygon to its corresponding .tif image. If no matching image is found for a polygon, the polygon is filtered out.

Step 2: Rasterization In this step, polygons are rasterized to create mask images that align with the original raster images. The mask is a tensor of 2 channels. The first channel stand for the class of level 1, the second for the class of level 2. Each pixel is mapped to an integer between 0 and 5 in channel 1, and to an integer between 0 and 16 in channel 2.

Step 3: Patching Rasterized masks and images are divided into smaller patches of specified sizes: 64x64, 128x128, and 256x256 pixels. Patches containing pixels that are not annotated are filtered out. Table 1 displays the results observed for each patch size:

- 256x256 pixels: 35% of the patches are kept, in other words, 65% of the patches are filtered out because they contain not annotated pixels. From the 35% remaining patches, 57% are homogeneous, meaning they are covered entirely by a single class at level 1.
- 128x128 pixels: 57% of the patches are kept. Of these, 72% are homogeneous.
- 64x64 pixels: 74% of the patches are kept. Of these, 84% are homogeneous.

Patch Size	Per. of Patches Kept	Per. of Homogeneous Patches
256x256	35%	57%
128x128	57%	72%
64x64	74%	84%

Table 1: Patch Size Analysis, Per. stands for Percentage

The choice of patch size depends on the task. For pixel classification tasks, specifically semantic segmentation, one of the goals is to accurately delineate boundaries within a patch. Thus, minimizing the number of homogeneous patches is crucial, which is why a patch size of 256x256 is chosen. Increasing the patch size to 512x512 would lead to an estimated loss of data around 90%, which is considered as too much. 256x256 is seen as a reasonable trade-off between patch homogeneity and the number of patches kept. As noted in Section 2.1.2, polygons of high interest are generally contained in patches of approximately 170x170 pixels, so there is a high probability that these polygons, along with their boundaries, will be included in the 256x256 patches.

For patch classification, we need to consider a small patch size to achieve a high resolution in the reconstructed mask, which leads to a patch size of 64.
Thus, a dataset is built with a patch size of 256 and another with a patch size of 64.
Now that the dataset is prepared and structured for DL, we can move forward with detailing the methods to process it.

3 Method

To evaluate the potential of DL with ECOMED dataset, we study two approaches. The first one involves classifying each pixel in the image with a semantic segmentation approach using UNet and EfficientNet as the encoder, which are described in subsection 3.1. In this same subsection, SAM is also introduced. The second approach implies classifying small images using ResNet, which is detailed in subsection 3.2. This subsection also covers the post-processing step inspired from Conditional Random Fields(CRF) implemented to harmonize patch classification. Finally, the last subsection defines the data shifts that often occur when applying methods originally developed with ideal datasets to real-world datasets, along with methods to mitigate them.

3.1 Pixel/Dense Classification

3.1.1 UNet

Semantic segmentation consists in classifying each pixel of an image and is successfully addressed with CNNs. As detailed in [35], CNNs owe their success to four key inductive biases: locality, multichanneling, weight sharing, and downsampling. The locality property assumes that pixels close to each other are more related than those further apart. This is exploited by aggregating local information (neighboring pixel values) around one pixel into a single value using convolutional filters. The same weights from the convolutional filter are applied to all values from the input image/feature maps with a sliding window, a process called weight sharing, which ensures translation invariance. Multichanneling, seen when increasing the number of channels in feature maps (e.g., from 64 to 128 in Figure 9), prevents information loss when downsampling reduces spatial dimensions (e.g., from 568 to 280 in Figure 9) to expand the receptive field, which is the region of the input image that influences a particular output neuron. As we move deeper into the CNN, the receptive field increases, meaning each neuron in the later layers "sees" a larger portion of the input image. Therefore, downsampling allow for capturing long-range correlations.
After the CNNs success from [17] in 2012 for a natural images classification task, the UNet has emerged as a sucessfull CNN architecture in [30] to segment neuronal structures in electron microscopic recordings. It has symmetric down-sampling and up-sampling paths with skip connections as shown in Figure 9. The up-sampling path and skip connections allow recovering the precise pixel location, essential in a task of pixel-wise classification, skip connections also avoid the vanishing gradient problem.

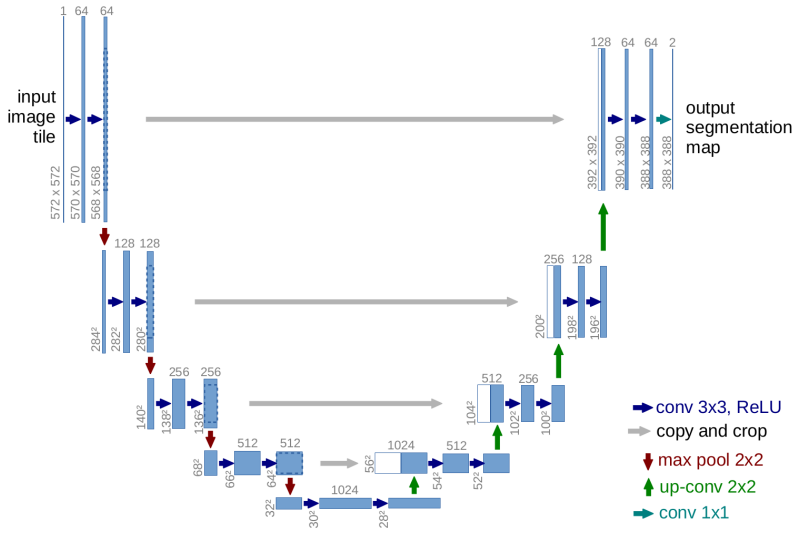

Figure 9: UNet architecture (Taken from [30])

3.1.2 EfficientNet Encoder

EfficientNet-b7 will be used as the UNet encoder, taking the place of the down-sampling path shown in Figure 9. EfficientNet, introduced in [32], is a scaled CNN that uses the same basic block as MobileNet [13], optimized to achieve a good trade-off between accuracy and the number of FLOating Point operations (FLOPs) required to run the model. The main contribution of EfficientNet is its method for scaling small CNNs to handle complex tasks. Large CNNs generally perform better on complex tasks like semantic segmentation with multiple classes, especially with large images such as RS images. This is because such images require many network layers to extend the receptive field and many channels to capture finer patterns. This creates a need to increase the capacity of "small" CNNs like ResNets [12] and MobileNets [13]. However, scaling depth and width relative to each other while maintaining reasonable computational costs is challenging. [32] delivers a method to scale CNNs in depth, width, and resolution using a compound coefficient φ. This user-defined parameter φ controls the extent to which we want to scale the network, where the network depth increases by α^{φ}, width by β^{φ}, and image size by γ^{φ}. The constants α, β, and γ are determined with a grid search on the original smaller model. The scaled EfficientNet-b7 was found to give the best performance, which makes it the best option.

3.1.3 SAM

When addressing a semantic segmentation task in 2024, one cannot overlook the significant impact of the Segment Anything Model (SAM) made of Vision Transformers (ViT) blocks([8]) introduced in [16]. Transformers like ViT lack the inductive biases of CNNs, such as weights sharing (translation equivariance) and locality, meaning they require more data than CNNs to learn spatial relationships and generalize effectively. As we will discuss later in the report, the experiments reveal that CNNs show limited results with the available data. Since ViTs generally require much larger datasets to outperform CNNs, they are likely to perform worse with the same limited dataset. Therefore, training a ViT from scratch will not be investigated. However, SAM presents a new opportunity. Introduced in Fall 2023 by [16], SAM is a pre-trained foundation model for image segmentation that has already learned image-related inductive biases. Key elements to note are that SAM has been pre-trained on a massive dataset (1 billion masks), and has strong zero-shot prediction capabilities. SAM's predictions can be guided through prompts such as points, boxes, or text, and it can compute masks in real-time. The model consists of an image encoder to compute image embeddings and a prompt encoder to compute prompt embeddings. Both embeddings are combined in the mask decoder. SAM can provide several predicted masks in cases of ambiguity about what to segment (e.g., a point on a shirt may indicate either the shirt or the person wearing it), this model has been improved in two notable ways:

1. MobileSAM V1 from [36]: Reduced the time needed by the encoder for segmenting entities related to a selected prompt by distilling the knowledge of the encoder into a smaller encoder.

2. MobileSAM V2 from [37]: Decreased the time needed by the mask decoder for segmenting everything in the image (without prompt guidance) by replacing costly grid search prompts with valid prompts generated by YOLOv8 [29].

When applied to specific fields such as RS, SAM's zero-shot performance tends to decrease, creating a need for fine-tuning. Fine-tuning such a model is computationally costly, and [38] proposes to tackle it by using Low-Rank Adaptation (LoRA) from [14]. LoRA assumes that during task adaptation, weight updates have low intrinsic rank. Given a pre-trained weight matrix $W_0 \in \mathbb{R}^{d\times k}$, we decompose its update as $\Delta W = BA$, where $B \in \mathbb{R}^{d\times r}$, $A \in \mathbb{R}^{r\times k}$, and $rankr \ll \min(d, k)$. During training, W_0 is frozen, and only A and B are trainable. The modified forward pass is:

$$h = W_0 x + \Delta W x = W_0 x + BAx$$

We believe SAM could be effective for delineating polygons of natural habitats, particularly when using the lighter encoders and decoders in MobileSAM V1 and V2, and when fine-tuned with LoRA. However, SAM will likely encounter similar challenges to those faced by CNNs, such as data shift and noisy labels, which may limit its suitability given our current data constraints. Therefore, SAM will be used primarily to demonstrate its zero-shot potential for roads and trees polygon delineation to experts at ECOMED.

3.2 Patch/Scene Classification

While delineating the boundaries of a polygon precisely at the pixel level is very interesting for experts, we must consider the constraints of our dataset: 57% of it consists of homogeneous patches. For these patches, since all pixels belong to the same class, recovering the precise location of every pixel in the mask is unnecessary and running the model will consume time and compute for nothing, the up-sampling path of the UNet is not required for these cases. Also, polygon delineations are noisy at the pixel level, as pointed out in Section 2.1.2, which will probably deteriorate the performances. Considering these two points, we will shift to a task of scene classification using small patches. It will save time and computation efforts allowing to conduct more experiments. The pixel level noise might be skipped as the focus will be on detecting a class of an image patch and not of a pixel, which could increase the performance.

3.2.1 ResNet

The model used is ResNet-18, presented in [12], which is a CNN architecture detailed in Figure 10. We modified the final Fully Connected Layer(FCL) to output 6 or 7 classes. Similarly to UNet and EfficientNet, ResNet has residual connections(also called skip connections) to avoid vanishing gradients. We select ResNet18 because it is the smallest and least computationally heavy.

Layer Name	Output Size	ResNet-18
conv1	$112 \times 112 \times 64$	7×7, 64, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	$1 \times 1 \times 512$	7×7 average pool
fully connected	1000	512×1000 fully connections
softmax	1000	

Figure 10: ResNet18 architecture from [12]

3.2.2 Stepwise ResNet Design

As detailed in Section 2.2.1, natural habitat mapping consists of two conceptually distinct steps for an expert: polygon delineation and polygon classification. In the pixel classification model, these sub-tasks were merged, but in the patch classification model, we decided to handle them incrementally to manage

complexity step by step. In the first phase, we only focus on semantic classification—identifying the class within patches—which simplifies the training process and model design. In the second phase, we incorporate boundary detection into the model. Below, we provide the specific configurations for each step.

The first step consists in classifying only homogeneous patches, adopting a single-label configuration-each patch can be predicted one single label-with an output vector targeting 6 classes, listed in Figure 7. The output of the last FCL is passed through a softmax function to produce a probability distribution over all classes (where the sum of the probabilities equals 1). This step is temporary as it requires prior knowledge of whether a patch is homogeneous or not to build the dataset, which is an information provided by the groundtruth mask.

The second step consists in classifying all patches, adopting a multi-label configuration-each patch can be predicted multiple labels-with a supplementary class dedicated to predict if the patch is homogeneous or heterogeneous, which we will call fronter class. In this case, each output of the last FCL is passed through a sigmoid function independently to produce a probability between 0 and 1 for each class. In this configuration, 2 thredholds α_1 and α_2 are needed. When the 7th probability exceeds α_1, the frontier class is set to 1 which means that the patch is predicted as heteregenous. In such case, all classes with probabilities greater than α_2 (obviously, the frontier probability is excluded) are selected. Otherwise, the patch is predicted as homogeneous and the predicted class is the one predicted with the maximum probability, excluding the probability of the frontier class.

In this second step, we also tried another setup with two output heads: one for predicting the habitat class(es) (multi-labels) with 6 classes in output, and another for determining whether the patch is homogeneous or heterogeneous (single-label) with one class in output. This approach aligns more closely with the conceptual distinction experts make between the two sub-tasks.

3.2.3 CRFs Post-processing

In the patch classification model, neighboring patches are treated separately and not considered in relation to each other. However, a commonly accepted assumption in both image processing and geography is that closer objects tend to be more similar. This idea was implemented at the pixel level with convolutional filters, as detailed in Section 3.1. To incorporate it at the patch level and harmonize patches with their neighbors, a post-processing step was introduced that injects prior knowledge by using CRFs.

CRFs are a type of discriminative probabilistic graphical model defined as undirected graphs, where each node corresponds to a random variable or group of random variables, and the edges represent probabilistic relationships between these variables. To bridge the gap between CRFs and our configuration, let us consider a mask of patches as a grid I, and each patch in it as a node. CRFs are said discriminative because they focus on the posterior probability $p(y|X)$, with X being the observations and y the labels. The posterior distribution for a patch i $p(y_i|X)$ can be given by any classifier, ResNet in our case. Usually, a CRF is determined by an energy function with two terms: the unary potential and the pairwise potential.

$$U(Y|X) = \sum_{i \in I} D_i(y_i|X) + \lambda \sum_{i \in I} \sum_{j \in N_i} V_{i,j}(y_i, y_j|x_i, x_j)$$

Here, N_i are the neighborhood patches around patch i in I, x_i and x_j are the probability vectors of patches i and j, respectively. In [5], [34], and [23], the unary potential is set as the negative log-posterior probability predicted on patch i:

$$D_i(y_i|X) = -\log p(y_i|X)$$

The second term of the energy is the pairwise potential, it can be defined to enforce consistency among neighboring pixels as it was done in in [5], [34], and [23]:

$$V_{i,j}(y_i, y_j \mid x_i, x_j) = [1 - \delta(y_i, y_j)]\, K(x_i, x_j)$$

where $\delta(\cdot)$ is the Kronecker symbol (i.e., $\delta(y_i, y_j) = 1$ for $y_i = y_j$ and $\delta(y_i, y_j) = 0$ otherwise). The kernel K is a Gaussian kernel defined as:

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

$K(x_i, x_j)$ is high when x_i and x_j are similar.

The pairwise potential increases when two neighboring patches are predicted with different classes, and it increases more when these patches have similar probability vectors.
To minimize the overall energy of the grid-CRF, multiple algorithms can be applied, such as graph cuts [4].

The first step of the CRF algorithm we designed is to reconstruct the original mask from the patches. We define the set of neighborhoods of a patch located at position (i, j) in I as:

$$\{(i + x, j + y) \mid x \in \{-1, 0, 1\}, y \in \{-1, 0, 1\}, (x, y) \neq (0, 0)\}$$

The ResNet model outputs a vector with a frontier class and predicts frontier patches with multiple labels. The goal of the post-processing is to smooth regions with single-labeled patches, therefore the second step of our CRF algorithm focuses on patches predicted as homogeneous by filtering patches with frontier class probability above α_1. Then, the frontier class is removed from the probability vector output by the ResNet model, and we finally obtain the posterior $p(y_i|X)$ for each homogeneous patches. The unary potential is defined identically as in [5], [34], and [23]. The pairwise term is more simply defined as:

$$V(y_i, y_j \mid x_i, x_j) = \lambda \sum_{i \in I} \sum_{j \in N_i} \mathbb{1}(y_i \neq y_j)$$

λ is a hyperparameter controlling the weight of the pairwise term. This term penalizes the energy each time a neighborhood j is predicted with a class different from patch i. Contrary to the pairwise term described previously, this one does not weigh the penalty according to the distance between x_i and x_j.

We can finally express the energy of a single patch as:

$$E_i = -\log p(y_i|X) + \lambda \sum_{j \in N_i} \mathbb{1}(y_i \neq y_j)$$

To minimize the energy, we adopted a sequential approach(over graph cuts), where we minimize the energy of each patch sequentially, assuming it will reduce the overall energy, while accepting that there is no guarantee of reaching a minimum. The idea is to reconstruct the original mask from the predicted patches, to scan them one by one, starting from the patch in the upper left of the mask, and to update the predicted value by taking the class with the minimum energy.

The pseudo-code can be found below:

```
for patch m in mask:
    if x_i[-1] > alpha1:  # If patch is predicted as homogeneous
        x_i = x_i[:-1]
        # Remove the last probability, x_i has now 6 values
        # For now, Ym = max(x_i)
        E = []  # Initialize an empty energy vector E

        for k = 0 to |x_i| - 1:
            E_k = -log(x_i[k]) + lambda * sum(Z_kj for j in N_i)
            with Z_kj = 1 if patch j is predicted with another class than k
            E.append(E_k)

        Ym = class corresponding to min(E)
```

3.3 Data Shift Mitigation

A data shift happens when there is a change in the joint probability distribution of the covariates X and the labels y. In [28], A. Storkey categorizes causes of data shift into 6 groups: simple covariate shift, prior distribution shift, sample selection bias, imbalanced data, domain shift, source component shift. In contrast, [26] organizes data shift based on its effects on conditional and unconditional probability distributions into four non-overlapping groups: covariate shift, prior distribution shift, concept shift, and other types of shift. Each of these types of shift is briefly detailed below:

- Covariate shift is the most common and happens when $p_{train}(y|X) = p_{test}(y|X)$ and $p_{train}(X) \neq p_{test}(X)$.
- The prior probability shift or labels shift happens when $p_{train}(X|y) = p_{test}(X|y)$ and $p_{train}(y) \neq p_{test}(y)$.
- The concept shift happens when $p_{test}(y|X) \neq p_{train}(y|X)$ with $p_{train}(X) = p_{test}(X)$.

In computer vision, Domain Adaptation (DA) addresses data shift by adapting models to a target domain using available target data. Domain Generalization (DG), on the other hand, focuses on building models that generalize well to unseen domains without access to target domain data, which is much more challenging. To tackle DG, a first family of methods aim to expose the model to a broad range of source domains, which is done with data augmentation for example. Another one of these approaches combines feature statistics from different source domains, as done in [39], while [21] increases domain diversity through data augmentation directly in the feature space. Another family of approaches focuces on the learning part of the model (and not on the data). For example, [15] proposes a model that learns both domain-invariant and domain-specific representations. In this method, features are polarized by minimizing the cosine similarity between domain-specific and domain-invariant features. Classification is then performed using the domain-invariant features.
In this section, we presented the pixel classification model, as well as the patch classification model, which was built incrementally with a post-processing step implemented to smooth the predictions. We also reviewed the typical data shifts and some methods used to tackle them. Next, we will report on the experiments conducted and present the results.

4 Experiments

In this section, we will first detail the tools used for implementing the methods, the training details, the metrics used to quantify performance, and the data splitting setting. Following this, we will present

the experiments conducted with the pixel classification model and the results obtained. Finally, we will analyse the results from the experiments conducted with the patch classification model.

4.1 Settings

4.1.1 Hardware and Software Tools

Hardware The code was parallelized on GPUs with CUDA. Initially, two GPUs were used: the NVIDIA RTX 4000 Ada with 20 MB of memory and the NVIDIA GeForce RTX 2080 Ti with 11 MB of memory. Due to frequent GPU crashes and CUDA errors, the setup was upgraded to NVIDIA GeForce RTX 3090 GPUs, each with 24 GB of memory, which were used for the remainder of the project.

Software UNet was implemented using the Segmentation Models PyTorch(smp) library, ResNet was imported from PyTorch, and the post-processing algorithm was coded from scratch. The IoU and F1 metrics presented below in subsection 4.1.3 were coded from scratch because the existing implementations did not match our configuration.

4.1.2 Training Details

To optimize the full UNet with EfficientNet as the encoder, we use the Dice Loss as the cost function and the Adam Optimizer for minimization. For optimizing the ResNet, we use the Cross Entropy for the single-label setting and the Binary Cross Entropy for the multi-label setting, with Adam as the optimizer.

4.1.3 Evaluation Metrics

The metrics used are the mean Intersection over Union(mIoU) and the mean F1(mF1). IoU is the division of the area of intersection between the predicted and Ground Truth(GT) regions by the area of their union. F1 is the harmonic mean between precision and recall. In term of True Positive(TP), False Positive(FP), True Negative(TN) and False Negative(FN), IoU and F1 are defined as follow:

$$\text{IoU} = \frac{\text{Intersection}}{\text{Union}} = \frac{TP}{TP + FP + FN}$$

$$\text{F1} = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} = \frac{2}{\frac{1}{\frac{TP}{TP+FP}} + \frac{1}{\frac{TP}{TP+FN}}} = \ldots = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

In general, F1 scores are higher than IoU because it penalizes less FP and FN. While IoU is the standard metric in semantic segmentation, probably because this task comes from medical applications where it is crucial to avoid FP and FN, F1 is more used in image classification tasks. Image classification standard methods is generally applied to natural images to classiffy cats and dogs or digits, the importance of FP and FN is therefore less pronounced. To compare the results from semantic segmentation with those from existing literature, we will compute the mIoU for these experiments. For comparing results from patch classification with the literature, we will calculate the F1 score. Although a direct comparison between pixel and patch classification is not possible, they may be compared indirectly. To facilitate this, we will also compute the F1 score for the pixel classification model and use F1 as a common metric across all experiments.

4.1.4 Data Splitting

It was detailed in Section 2.1.1 that patches come from images, which come from geographical zones. The naive splitting to be conducted as a baseline is to randomly shuffle the patches whatever the images and zones they come from and distribute them into a training(60%), a validation(20%) and a test(20%) set.

This setting is called random shuffling, it ensures a same balance of classes across sets but does not allow to evaluate the model capability to generalise to unseen image or zone.
The second approach is to split the data in a stratified manner by image, ensuring that all patches from the same image are assigned to the same set (train, validation, or test) and not split among them. The third approach stratifies by zone, where all patches from the same zone are assigned to the same set. These two approaches allow us to measure the model's ability to generalize to unseen images or zones but do not preserve class balance across sets.
Ideally, we would like to split the dataset while maintaining both (1)similar class balance across sets and (2)the integrity of zone groups—assigning entire zones to a single set without splitting them. This problem could be approached as an optimization problem, as described in [2]. However, since this approach is beyond the scope of this report, we must choose between these two characteristics. It is more important to build a model that can generalize to unseen zones, even if it is biased toward the most represented classes, than to create an unbiased model that fails to generalize. Therefore, the chosen splitting process is stratified by zone, which does not guarantee equal class balance across sets.
A map of the stratification by image and by zone can be seen in Figures 11 and 12.

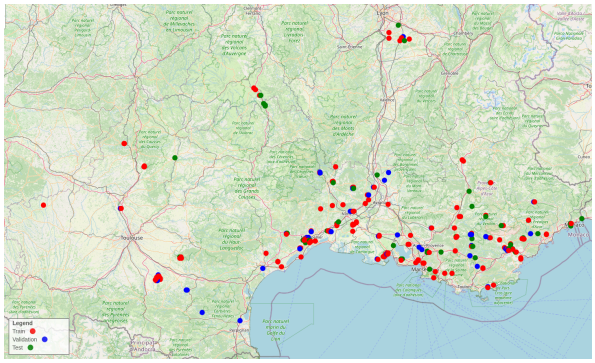

Figure 11: Map of data stratification by image

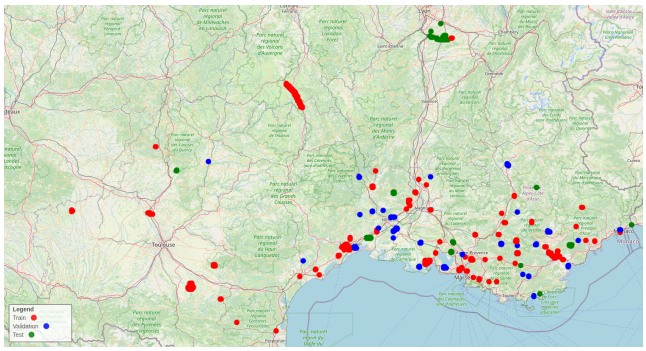

Figure 12: Map of data stratification by zone

Now that the hardware, software tools, training parameters, evaluation metrics, and splitting strategy have been set, let us move forward with the pixel classification experiments.

4.2 Pixel/Dense Classification Experiments

The first experiment focuses on ensuring that any mismatch between the annotations and the provided images does not affect the relationship between a label and its representation. Next, we test the UNet model's ability to generalize across different zones. In the second experiment, we identify a significant data shift across zones, which we will try to describe more precisely in the third point. To address it, in the fourth experiment, we explore using a pre-trained model, augmenting the data, and restricting the study to the Mediterranean region. Additionally, as a side experiment with UNet, we conduct a NIR ablation study to assess the utility of the NIR channel. Finally, we applied SAM to assess its zero-shot potential for delineating the polygons. The experiments of dense classification can be categorized into six distinct groups:

1. UNet: Annotation-ground time consistency
2. UNet: Spatial generalisation
3. UNet: Identifying data shift
4. UNet: Mitigating data shift: pre-training, data augmentation, Mediterranean focus

[2] url_Stratified_Splitting_Grouped_Dataset

5. UNet: NIR Ablation

6. SAM: Preliminary Zero-shot Results

4.2.1 UNet: Annotation-ground Time Consistency

As explained in Section 2.1.2, more than half of the polygons were annotated before 2023, the year of image capture. It might have happened that for these polygons, at the time of the picture capture, the natural habitat on the ground had changed and no longer corresponds to its corresponding label, which could mislead the model and degrade performance. To test if this issue is present, we compare the results for random shuffling with annotations from all dates and with annotations only from 2023.

Shuffling setting	**Year**	**mIoU**	**mF1**
Random	All	0.56	0.71
	2023	0.47	0.64

Table 2: UNet, stratified shuffling by zone, all vs 2023 year

The table 2 shows that restricting the dataset to 2023 results in a performance drop, with the mIoU decreasing by 0.09. This decline is likely due to a significant reduction in the amount of data (and, in particular, training data), with approximately 40% of the dataset being lost under the 2023 restriction. This reduction in data outweighs any potential benefits from improved alignment between annotations and the images. Given that the loss of data has a greater impact than correcting the potential mismatch between annotations and ground truth, we will continue using the full dataset.

4.2.2 UNet: Spatial Generalisation

Shuffling	**Setting**	**mIoU**	**mF1**
Random	-	0.56	0.71
Stratified	by zone	0.24	0.36
	by image	0.26	0.4

Table 3: UNet, random vs stratified shuffling

Table 3 shows that the mIoU drops to 0.26 with stratification by zone. In a study made by the French National Institute of Geographic and Forestry Information(IGN) in [11], a similar model, task, and data as us were used: LC mapping in France with UNet on VHR images at 20 cm/pixel, including RGB-NIR channels. They achieved an mIoU of 0.56 with the stratified by zone setting (with zones representing actual regions in France). Therefore, our results with stratification by zone are notably lower than those reported in the existing literature. The performance gap between the random and stratified settings, as well as between our stratified setting with ECOMED data and the stratified setting with IGN data from [11], may be attributed to a data shift, suggesting that the images distribution from training, validation, and test sets differ a lot from each other.
The confusions matrices from Figure 13 reveal that class 1 “Heathland, scrub and tundra” and class 2 “Woodlands, forests, and other wooded habitats” are often confused in stratified setting. These classes differ often based only on phisiognomy and its is probable that a human too would struggle to differentiate them from RS images. Also, class 3 (“Regularly or recently cultivated agricultural, horticultural and domestic habitats”) is often predicted as class 0 (“Grasslands and lands dominated by forbs, mosses or lichens”), which is not absurd as we saw in subsection 4.3.2 that many samples in “Grasslands and lands dominated by forbs, mosses or lichens” have long range dependencies similarly to samples from “Regularly or recently cultivated agricultural, horticultural and domestic habitats”.

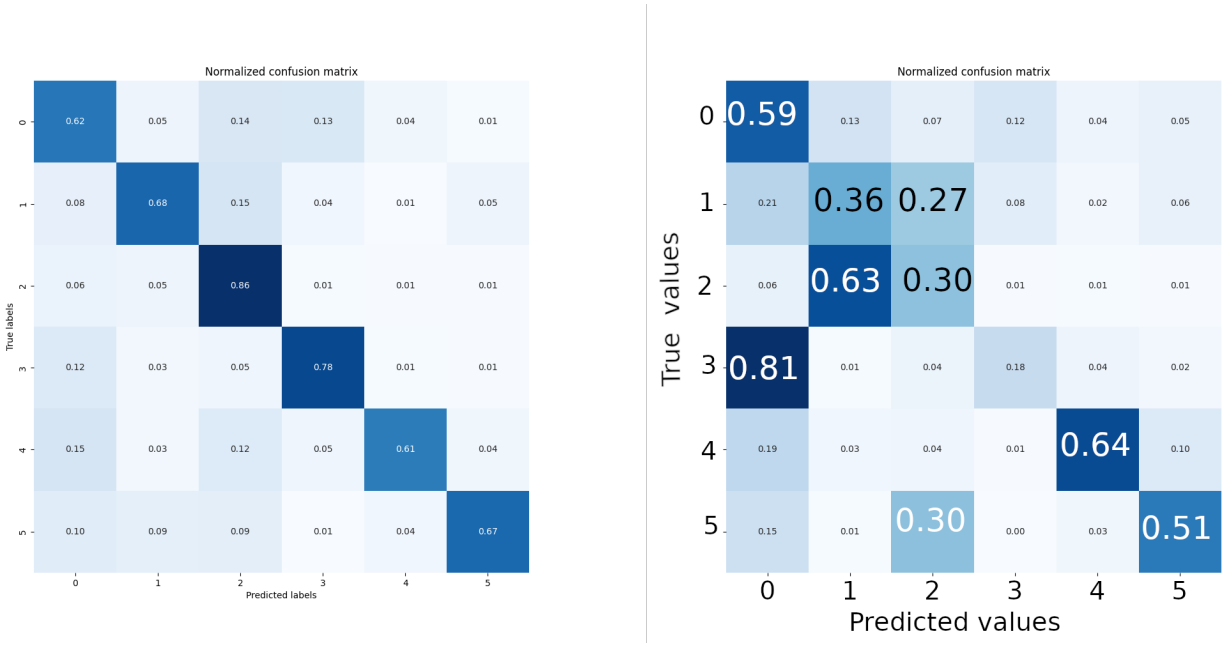

Figure 13: Confusion matrices. Left: Random shuffling. Right: Stratified shuffling by zone.

4.2.3 Identifying Data Shift

Let us try to identify the types of data shift we might face, relatively to the ones exposed in subsection 3.3:

- We think the covariate shift is present, because the balance of level 2 classes ar quite different across set for each level 1 class. Figure 14 shows for example the balance of sub-classes inside "Constructed, industrial and other artificial habitats" [in French: "Zones bâties, sites industriels et autres habitats artificiels"] class in stratified sampling by zone. The sub-class "bâtiments des villes et des villages" is much more represented than the sub-class "réseaux de transports" in the training set compared to the testing set. In Figure 34, we can see that these 2 subclasses have underlying probability distribution very different. As a result, the probability distribution of X for the same class "Constructed, industrial and other artificial habitats" in training might be quite different from the one in the testing.

- We believe prior distribution shift is present too, as the balance of classes at level 1 is not ensured. The stratification was made in subsection 4.1.4 to ensure zone group integrity, at the cost of equal balance of level 1 classes across sets. The unbalance of classes among sets for stratification by zone setting can be viewed in Figure 15.

- We do not think the concept shift happens as the descrition of EUNIS classes does not change across the sets (relationship between a label and the data).

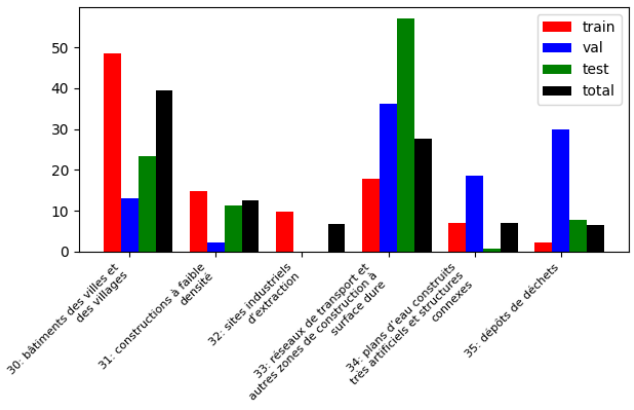

Figure 14: Stratification by zone, level 2 classes balance for level 1 class "Artificial"

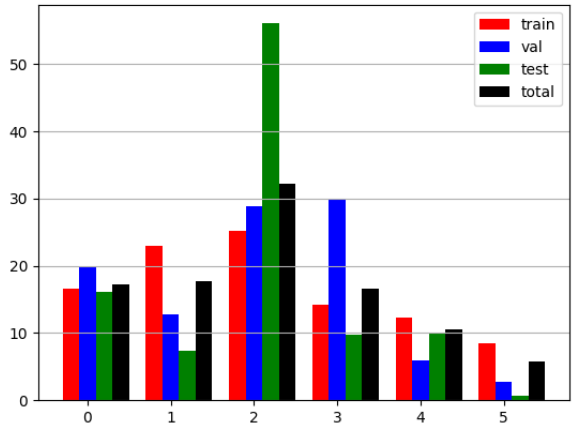

Figure 15: Stratification by zone, level 1 classes balance

4.2.4 UNet: Mitigating Data Shift

Pre-training To check if the pre-training with ImageNet is correctly set in place, we ran it with the baseline random shuffling configuration. Table 4 shows that it works as expected, with an increase of 0.1 in mIoU. However, we are particularly interested to see if using pre-trained weights improves performance for stratified shuffling. Tables 5 and 6 show that it does not considerably improve it; the distribution shift is too significant to benefit from transfer learning.

Shuffling setting	Configuration	mIoU	mF1
Random	From scratch	0.56	0.71
	Pre-trained	0.66	0.78

Table 4: UNet, random shuffling, from scratch vs pre-trained

Shuffling setting	Configuration	mIoU	mF1
Stratified by image	From scratch	0.26	0.4
	Pre-trained	0.3	0.43

Table 5: UNet, stratified shuffling by image, from scratch vs pre-trained

Shuffling setting	Configuration	mIoU	mF1
Stratified by zone	From scratch	0.24	0.36
	Pre-trained	0.24	0.36

Table 6: UNet, stratified shuffling by zone, from scratch vs pre-trained

Data Augmentation The transformations applied to address the data shift issue include random adjustments to brightness and contrast, modifications to hue (the color tint), saturation values, and the addition of Gaussian noise. Again, we applied these augmentations in random shuffling scenario to see if they were correctly set in place. Table 7 confirms that they are correctly applied, even though the improvement is modest with an increase of 0.05 in mIoU. This small improvement was expected because the random shuffling setting already has a training set with considerable diversity and high variability. When we focus on stratified shuffling by image (Table 8) and stratified by zone (Table 9), augmenting

the data in this way does not lead to substantial improvements. The variation across zones seems to be too high to be mitigated by data augmentation.

Shuffling setting	Configuration	mIoU	mF1
Random	Original data	0.56	0.61
	Augmented data	0.61	0.64

Table 7: UNet, random shuffling, original vs augmented data

Shuffling setting	Configuration	mIoU	mF1
Stratified by image	Original data	0.26	0.4
	Augmented data	0.27	0.41

Table 8: UNet, stratified shuffling by image, original vs augmented data

Shuffling setting	Configuration	mIoU	mF1
Stratified by zone	Original data	0.24	0.36
	Augmented data	0.26	0.38

Table 9: UNet, stratified shuffling by zone, original vs augmented data

We can observe that the results for stratified by zone and by image are consistently aligned with each other. One can be inferred from the other; therefore, we will not continue running both settings and will stick to the one most relevant to us: stratified by zone.

Mediterranean Zones When looking at the maps in Figure 11 and 12, we can see that the natural habitats are sourced from very different bioregions. For instance, some samples come from zones around Lyon, while others are from areas near the Mediterranean Sea. The climate and landscapes vary a lot between these regions. Therefore, we can assume that the poor results with stratification are due to a geographical shift. A class at level 1 might have different representations in Lyon compared to the Mediterranean. To address this, we restricted the study to the Mediterranean basin located in the purple oval in Figure 16

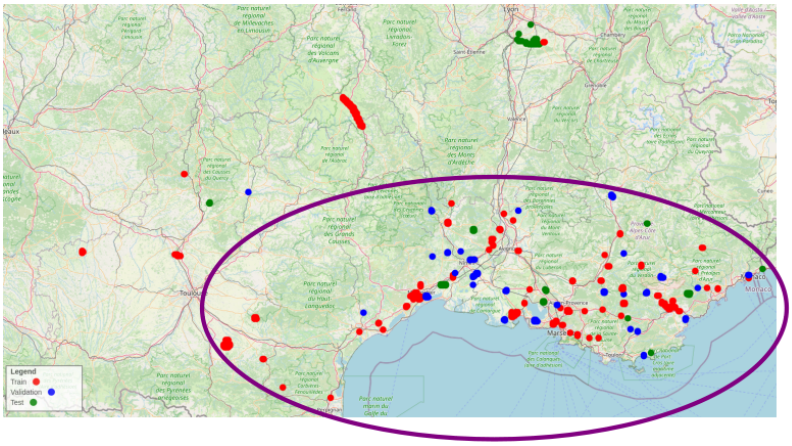

Figure 16: Mediterranean basin

Shuffling setting	Zones	mIoU	mF1
Stratified by zone	All	0.24	0.36
	Mediterranean	0.37	0.52

Table 10: UNet, stratified shuffling by zone, all vs Mediterranean data

Indeed, focusing on the Mediterranean zones led to notable improvements compared to using all zones as shown in Table 10. Yet, the differences in mIoU remain significant between random shuffling and stratified shuffling, with a difference of 0.2 in mIoU (0.56 vs. 0.37). Restricting the study to Mediterranean zones attenuates the data shift but does not eliminate it entirely.

4.2.5 UNet: NIR Ablation

Given the high cost associated with acquiring NIR data, as expressed by ECOMED, this experiment is done to evaluate its utility. Table 11 reveals that the NIR channel does not appear to contribute to reaching our target performance metric of mIoU around 0.5. Despite this finding, additional experiments are needed to validate the result before we can conclude that the NIR channel is not necessary.

Shuffling setting	Configuration	mIoU	mF1
Random	RGB-NIR	0.55	0.71
	RGB	0.52	0.68

Table 11: UNet, random shuffling, RGB-NIR vs RGB

4.2.6 Preliminary SAM Zero-shot Results

We evaluate SAM's zero-shot performance by applying it directly to large images and corresponding masks. The qualitative results in Figures 17 and 18 show that SAM delineate well the trees and roads but not the habitats with unclear boundaries, indicating the need for fine-tuning. But fine-tuning SAM is likely to face the same challenges as before, including noisy pixel-level labels, insufficient diversity in data sources, and partially resolved data shifts across different zones. Therefore, we decided to focus on addressing the inherent problems with the dataset rather than continuing with SAM.

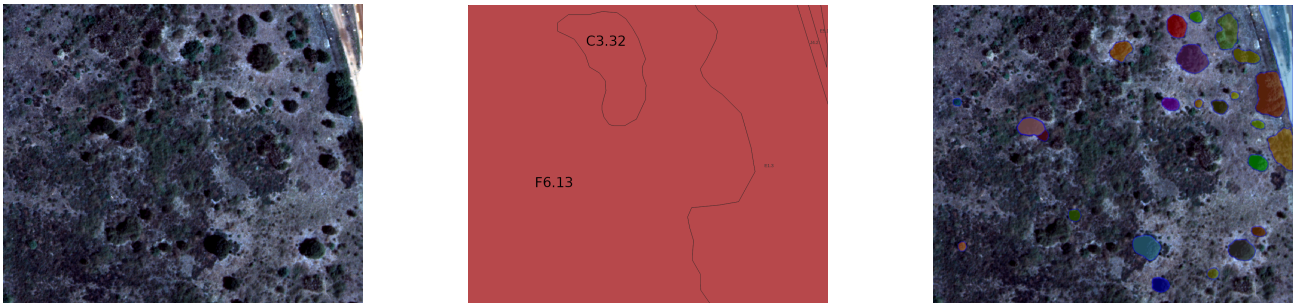

Figure 17: Left: Image, Middle: GT delineation, Right: SAM delineation prediction(zero-shot)

Figure 18: Left: Image, Middle: GT delineation, Right: SAM delineation prediction(zero-shot)

During this experiment, we also identify 2 previously unnoticed issue: in some cases, trees are delineated as in the GT from Figure 18, while in others, they are not as in the GT from Figure 17. Discussions with experts confirmed that this inconsistency is widespread across the dataset. Additionally, the GT seems quite difficult to establish only from the provided RS image, such as distinguishing a "Provence cane beds"[code C3.32, in French: "Formations à Arundo Donax"] from a "Western cistus garrigues" [code F6.13, in French: "Garrigues occidentales à Cistus"] in Figure 17. This observation will be analyzed later in subsection 5.1.3.

In the first subsection of the dense classification experiment, we saw that the loss of data outweighs the benefits of correcting date annotation mismatches. We also evaluated the UNet model's ability to generalize to unseen zones by shuffling the data in a stratified way and identified a significant data shift related to a covariate and a prior distribution shit. Among the approaches we tried, restricting the study to Mediterranean zones was effective in partially addressing this issue. Additionally, we found that the NIR channel was not helpfull to achieve our target performance of mIoU 0.5 in the random shuffling setting. Finally, we found that SAM performs well for delineating roads and trees in zero-shot. More than 60% of the dataset consists of homogeneous patches, for which the downsampling path of the UNet and the decoder of SAM are unnecessary. The disproportionate computational and time requirements ($\sim$6 hours per experiment with UNet) in relation to our dataset's constraints(noisy labels at pixel levels) justify the transition to patch classification.

4.3 Patch/Scene Classification Experiments

The patch classification experiments are divided into three categories. First, we will explore the single-label ResNet model designed for classifying homogeneous patches. Next, we will evaluate the multiple-label ResNet model, which is used for classifying all patches, including heterogeneous ones. Finally, we will present the results obtained with the post-processing applied.

4.3.1 ResNet: Homogeneous Patches

This setting is the first step of the incremental process to build the whole model. In this subsection, we will cover optimization experiments, assess the generalization ability of the model, and apply the same techniques as before to mitigate the data shift, including pre-training, data augmentation, and focusing on Mediterranean zones.

Optimization Experiments We evaluate various learning rates and optimizers to maximize the results with random shuffling. Specifically, we will test learning rates ranging from 10^-4 to 10^-3 and optimizers including Adam and AdamW. Once the best learning rate and optimizer(among the ones tested) is found, we will use it for the remainder of the experiments.

Learning Rate Table 12 shows that a learning rate of 10^{-3} results in better performance. The training and validation curves in Figure 19 indicate that the model overfits very quickly with a learning rate of 10^{-4}, converging to a score of 0.6 from epoch 20. It converges later with a learning rate of 10^{-3} but reaches a score of 0.7. With 10^{-4}, the model likely got stuck in a local minimum. From now on, we will use 10^{-3}.

LR	mF1
10^{-4}	0.60
10^{-3}	0.71

Table 12: ResNet, homogeneous patches, random shuffling, lr 10^{-4} vs 10^{-3}

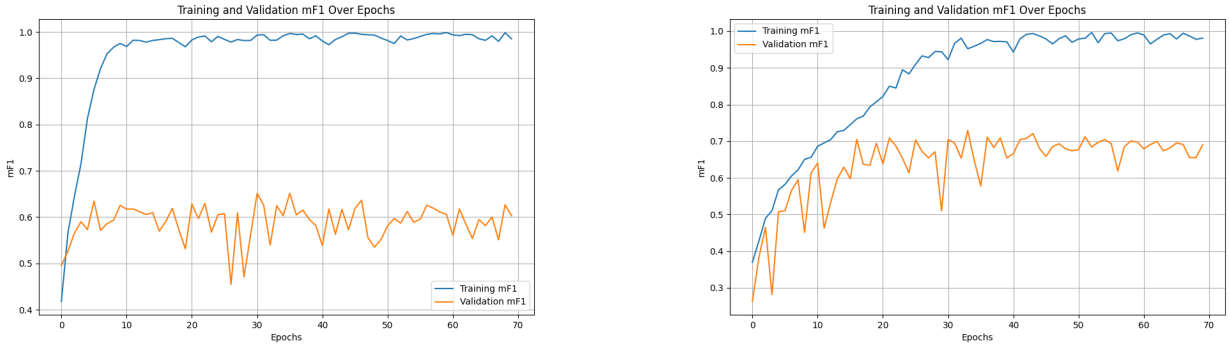

Figure 19: ResNet, homogeneous patches, loss curves. Left: Learning rate 10^{-4}. Right: Learning rate 10^{-3}.

Optimizer AdamW is a variant of the Adam optimizer which regularizes the network by penalising the big gradient update (weight decay), it should help to reduce overfitting. In our experiments, this advantage did not translate into better results as shown in Table 13 so we will continue using the Adam optimizer.

Optimizer	mF1
Adam	0.71
AdamW	0.69

Table 13: ResNet, homogeneous patches, random shuffling, Adam vs AdamW

Based on the results of the last two paragraphs, we will use a learning rate of 10^{-3} and the Adam optimizer for the remainder of the experiment.

Generalisation As reported in Table 14, the stratified-by-zone setting results in a big drop of performances probably due to a data shift, similarly to what was observed for semantic segmentation. The difference lies in the time needed to run the experiments. Runing the ResNet model with 256*256 patches need between 1 and 2 hours with patches of 256 compared to the 4 to 6 hours needed for the UNet model. This economy of computational ressources and time allows us to run more experiments. Also it is in line with the fact that the polygons delimitations are noisy and thus that the delimitation of polygons are not precise enough at the pixel level.

Shuffling	mF1
Random	0.71
By zone	0.29

Table 14: ResNet, homogeneous patches, random shuffling vs stratified shuffling

Mitigating Data Shift

Pre-training Since pre-training was already set in place with the UNet, we will not check it with Random Shuffling but rather focus directly on what it can bring us for the stratified shuffling. The pre-training increases only slightly the results as shown in Table 15.

Pre-trained	mF1
No	0.29
Yes	0.33

Table 15: ResNet, homogeneous patches, stratified shuffling by zone, pretrained

Data Augmentation The data augmentation applied here includes the same techniques as in subsection 4.2.4, with additional geometric augmentations such as random vertical and horizontal flips, rotations up to 45 degrees, and random resized cropping. As we added new transformations, we check the new setting with Random Shuffling first, and then apply in on stratified shuffling. Augmenting the data in this way increases the mF1 of 0.1 in Random Shuffling as seen in Table 16 which confirms it is correctly set in place, but it does not for the stratified shuffling (Table 17).

Augmented	mF1
No	0.71
Yes	0.81

Table 16: ResNet, homogeneous patches, random shuffling, data augmentation

Augmented	mF1
No	0.29
Yes	0.29

Table 17: ResNet, homogeneous patches, stratified shuffling by zone, data augmentation

Mediterranean Zone Restricting the analysis to Mediterranean zones surprisingly result in worse performance for stratified shuffling, as shown in Table 18. We suppose that the expected reduction in data shift could not compensate for the reduction in the amount of training data.

Zone	mF1
All	0.29
Mediterranean	0.2

Table 18: Mediterranean zones

At the end of the study on 256×256 homogeneous patch classification experiments, we learned that a learning rate of 10^{-3} is more effective, and Adam works well. The model continues to struggle with generalizing to unseen zones. Data shift is not mitigated by data augmentation, restricting to Mediterranean

zones, and slightly by using a pre-trained model. Next, we will move towards building a model that can also predict if a patch is heterogeneous or not: the multi-label ResNet model with a seventh class frontier.

4.3.2 ResNet: All patches

In this subsection, we will tune the thresholds α_1 and α_2, evaluate whether splitting the output head into two heads improves performance, analyze how the performance of frontier classes decreases when the patch size is reduced, measure the generalization capability of the model, test a pre-trained model published by the IGN, run t-SNE on the features extracted and experiment the classification at level 2.

Threhold Tuning The multi-label nature of this model imposes two thresholds: one to decide at what probability we consider the patch to be homogeneous, and one to decide at what probability we consider a class to be present in the patch. A naive threshold of 0.5 led to inconsistencies in the predictions. For example, patches predicted as homogeneous still had various classes predicted. To address this, it was decided that for patches predicted as homogeneous, only the class with the highest probability would be selected, while for those predicted as heterogeneous, classes with probabilities above a certain threshold α_2 would be kept. This approach requires tuning two thresholds: one to determine whether a patch is homogeneous or heterogeneous, and another to decide which classes to retain for heterogeneous patches. 1) Threshold tuning of α_1 (for the frontier class) by examining the precision vs recall curve and the F1:

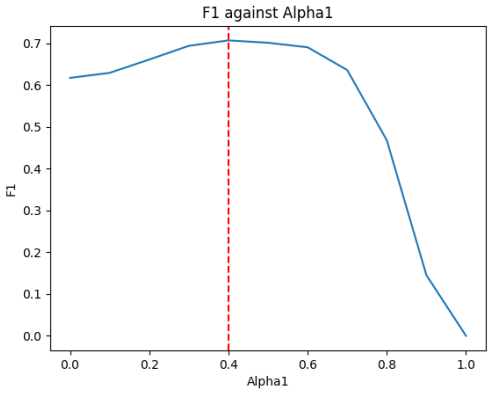

Figure 20: F1 for several alpha1

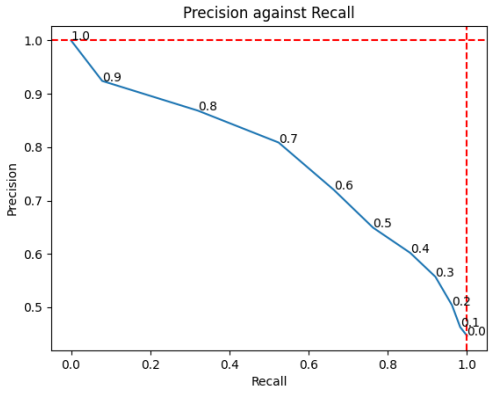

Figure 21: Precision vs Recall for several alpha1

The F1 score is maximized when the threshold α_1 is 0.4, yet the precision vs recall curve indicates that the Euclidean distance between points on the blue line and the top right corner (where precision = 1 and recall = 1) is minimized at a threshold of 0.6. The F1 score with $\alpha_1 = 0.6$ is very slightly lower than with $\alpha_1 = 0.4$, but it results in a better trade-off between precision and recall, so we will use alpha = 0.6.

2) Threshold tuning of α_2 (for the habitats cover classes):

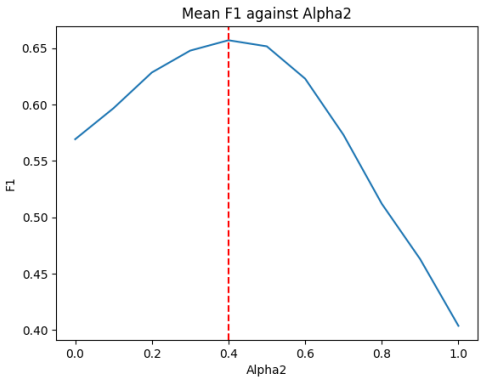

Figure 22: Mean F1 for several alpha2

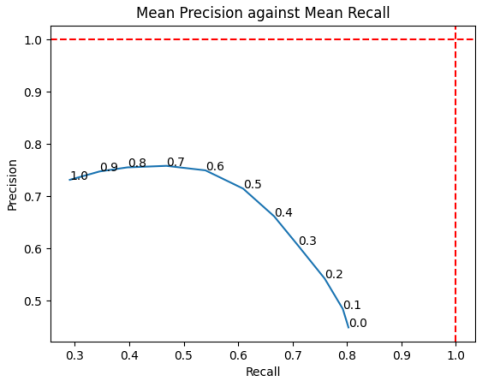

Figure 23: Mean precision vs recall for several alpha2

A threshold of 0.4 maximizes the F1 score and results in a good trade-off between precision and recall.

Thresholds Fine-tuned?	**Alpha1**	**Alpha2**	**mF1**
No	0.5	0.5	0.6
Yes	0.6	0.4	0.61

Table 19: Thresholds fine-tuning relevance

In terms of F1, the mean score is not significantly affected by fine-tuning the threshold. Considering the time required to tune both thresholds for all settings, we will stick with the basic thresholds of 0.5 and not perform further tuning for the rests of the experiments.

One Head vs. Two Heads From an expert point of view, the task of delineating the polygons (determining whether a patch is homogeneous or not) is different from the task of classifying the polygons(identifying which habitat is in the patch). To align with this conceptual split, we can divide the end of the network into two heads: one providing a binary classification to indicate if the patch is a frontier one, and the other outputting the class(es) present on the patch. This approach requires optimizing the network twice, once for each head, and results in worse performance for the frontier head, as shown in Table 20. This means that, by learning which class is in the patch, the model learns important information to predict the frontier probability, which makes sense. The conceptual distinction between boundary detection and habitat classification made by experts does not persist with the approach we have. Therefore, we will stick to our single-head model with 7 classes.

Head	**mF1 for frontier head**
1	0.36
2	0.13

Table 20: ResNet, multi-labels, one vs two heads

256-128-64 When we decrease the patch size, the model loses ability to detect frontiers accurately as it appears in Table 21. It can be explained by the fact that reducing the patch size results in reducing the ratio of frontier patches in the dataset.

Patch size	F1 frontier class
256	0.71
128	0.57
64	0.36

Table 21: ResNet, multi-Labels, frontier classification with carying patch size

Stratified Shuffling Similar to UNet and ResNet single-labels, ResNet multiple-labels results in a drop in performance when switching to stratification by zone, as shown in Table 22.

Shuffling	mF1
Random	0.71
Stratified by zone	0.29

Table 22: ResNet, multi-labels, random vs stratified shuffling

The techniques applied to mitigate data shift with ResNet single-label in subsection 4.3.1 were not successful, and there is no justification that they would work better with the addition of a seventh class. Therefore, we will not reproduce these experiments with ResNet multi-labels.

Pre-training ImageNet consists of natural images, which differ a lot from the RS images we are working with. This difference might explains why using a pre-trained model on ImageNet resulted in only a slight performance improvement as observed in subsection 4.2.4 and 4.3.1. In contrast, the IGN provides a pre-trained model trained on VHR RS images, as described in [11], which makes it much more relevant to our data. The pre-trained model is a UNet with a ResNet34 encoder. We used only the encoder weights and conducted an experiment to compare the results between ResNet34 with and without pre-training from IGN.

Pre-trained(IGN)	mF1
No	0.6
Yes	0.63

Table 23: ResNet34, all patches, from scratch vs IGN pre-trained

Table 23 shows that the pre-trained model from IGN does not significantly improve performance.

Intra-class Diversity Scores at level 1 are not convincing, and we assume that this is because the level 1 EUNIS classification aggregates sub-classes that are too different from each other. By focusing on classes from level 1, we consider that subclasses at level 2 within the same level 1 class are generally more visually similar to each other than to level 2 subclasses from a different level 1 class. This assumption is only partially confirmed by figures from 30 to 35 in the Appendix. These figures illustrate that:

- Intra class variability is high. For example, Figure 30 shows that the level 2 class "Woodland fringes and clearings and tall forb stands"[in French:"Ourlets clairières forestières"] has little in common visually with "Dry grasslands"[in French:"Pelouses sèches"], even though both are grouped under the same level 1 class, "Grasslands and lands dominated by forbs, mosses or lichens" [in French: "Prairies; terrains dominés par des herbacées non graminoïdes, des mousses ou des lichens"]. High intra-class diversity can be an advantage for better generalisation ability of the model, but it has its limits. While the representations of the same class might varies a lot in term of distribution, we should still be able to visually group together the items of the same class, which brings us to the second point.

- The assumption made that level 2 classes from a same level 1 class are more similar to each other than to sub-classes from others classes is not always confirmed. Figure 31, 32 and 33 reveal that "Shrub plantations"[in French: "Plantations d'arbustes"], with its long-range dependencies, which represent 16% of the level 1 class "Heathland, scrub, and tundra" [in French: "Landes, fourrés et toundras"] is visually more similar to items from the level 1 class "Regularly or recently cultivated agricultural, horticultural and domestic habitats"[in French:"Habitats agricoles, horticoles et domestiques régulièrement ou récemment cultivés"]. The same happens for some images of "Dry grasslands"[in French:"Pelouses sèches"] that are closer to the level 1 class "Regularly or recently cultivated agricultural, horticultural and domestic habitats" than to the level 1 class "Grasslands and lands dominated by forbs, mosses or lichens" or for images of "Maquis, arborescent matorral and thermo-Mediterranean"[in French: "Maquis, matorrals arborescents et fourrés thermo-méditerranéens"] classified as "Heathland, scrub, and tundra" but indistinguishable from images classified in the level 1 class "Forest". The classification was not designed by thinking to its potential use in LC processing, so classes are not thought to be visually consistent from RS images.

T-SNE Analysis t-Distributed Stochastic Neighbor Embedding(t-SNE) is an algorithm used to visualize high-dimensional data by reducing it to two or three dimensions. When applied to features from the last layer of a CNN, t-SNE can reveal how the network has clustered and differentiated classes, it ensures that similar data points remain close together in the lower-dimensional space.
The t-SNE visualization in Figure 24 appears to support quantitatively this idea that intra-class diversity is too high at level 1, as multiple clusters are observed for the same class. Descending to level 2 is expected to increase inter-class diversity while improving intra-class consistency, which may help address the identified issue.

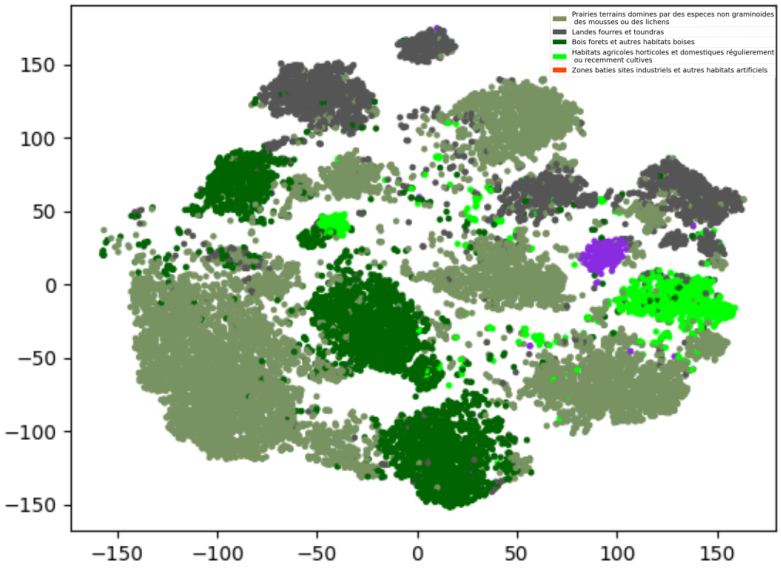

Figure 24: t-SNE ran on features extracted from the training data before the last FCL of Resnet18 single-label, stratified by zone. The purple points should be red.

Level 2 of Classification Figures from 30 to 35 in the Appendix and the class balance analysis from Figure 8 reveal that classes are not uniformly consistent. For instance, the level 1 "Woodlands, forests, and other wooded habitats" class is visually consistent, so moving to level 2 likely will not improve performance. Yet, level 2 could help distinguish the level 2 "Shrub plantations" from other level 2 classes inside the level 1 class "Heathland, scrub, and tundra".
Moving to level 2 comes with the cost of reducing the number of samples per class, which tends to lower

performance, as shown in Figure 25 of the F1 score by class. We think improvement occurs when moving to level 2 involves splitting subclasses that both (1) are visually different and (2) have relatively large amounts of data compared to other level 2 classes. Notably, the F1 curve by class mirrors the dataset's class proportions (Figures 25 and 8, plotted again alongside the level 2 results in Figure 26), reflecting a bias towards dominant classes due to the dataset's imbalance.

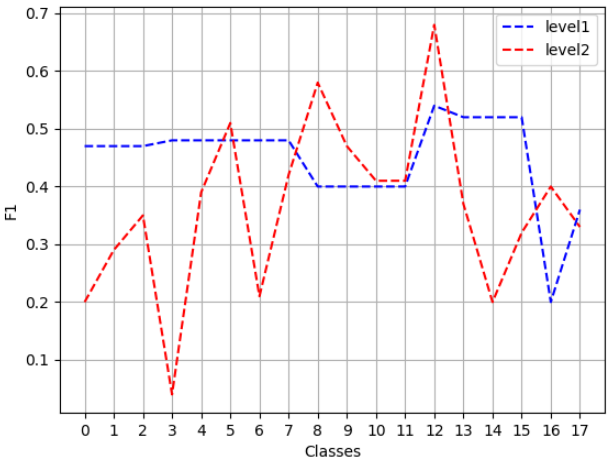

Figure 25: ResNet, multi-labels, level 2 of EUNIS classification

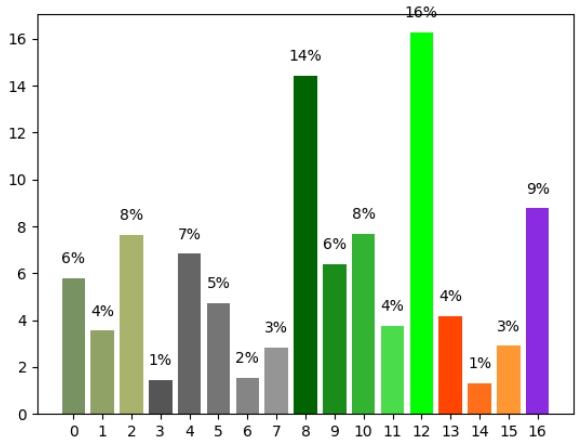

Figure 26: Percentage of area for each class at level 2

To recap this subsection's findings, we observed that tuning α_1 and α_2 reesulted in negligible performance improvements. Additionally, splitting the output head into two increased optimization time and led to decreased performance for frontier classes. We also found that predictions for frontier classes gets better with bigger patches, the model continues to generalize poorly, performance gains with the IGN pre-trained model are limited, t-SNE analysis reveals clusters intra-class, and level 2 classification does not effectively improve the results. Moreover, there is a strong correlation between classes distribution and the results obtained, raising the concern that the model may simply be learning the classes distribution rather than distinguishing classes effectively.

4.3.3 CRFs Post-processing

Before and after the post-processing step, the observed mF1 does not improve and sometimes decrease. However, post-processing is not aimed at improving the classification of each patch (measured by mF1) but rather at smoothing the area to create a reconstructed image with coherent regions. Therefore, the quality of post-processing is better evaluated by examining the plots of the reconstructed images. The three reconstructed image plots from Figure 27, 28 and 29 contain fewer isolated patches of a single color, revealing qualitatively that the post-processing step seems effective.
Note that the three zones plotted below were excluded from the training data with the random shuffling setting to prevent overlap between the training and test sets. Consequently, the setting is not purely random anymore.

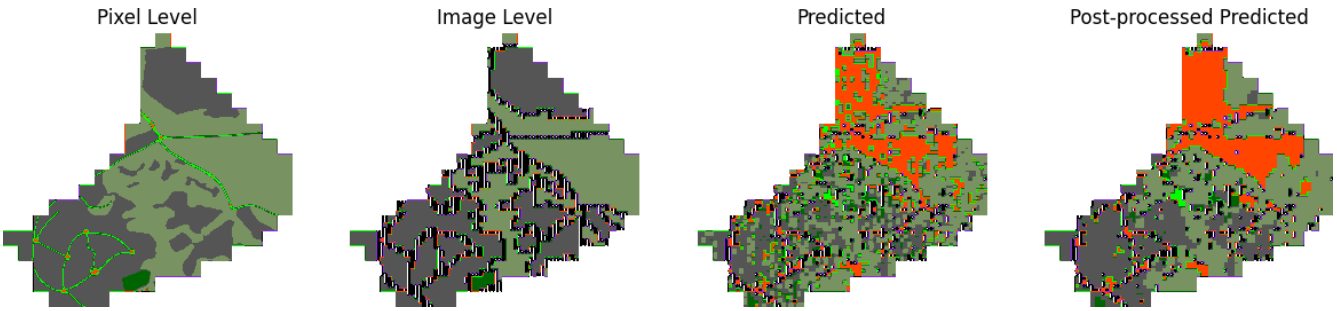

Figure 27: Zone 1 patch size 64, random shuffling, no data augmentation, no pre-trained, post-processed. Two Left images: GT. The black and white striped patches correspond to boundary patches

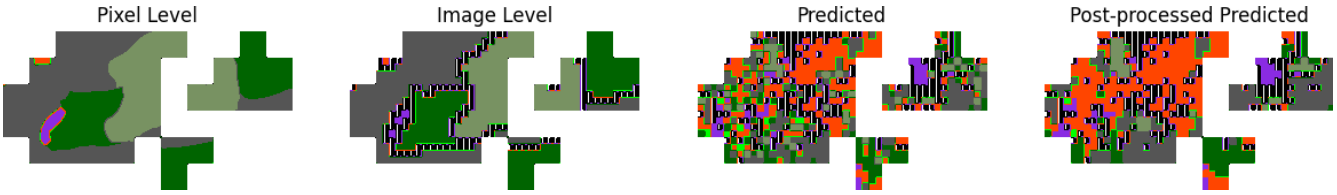

Figure 28: Zone 100 patch size 64, random shuffling, no data augmentation, no pre-trained, post-processed. Two Left images: GT

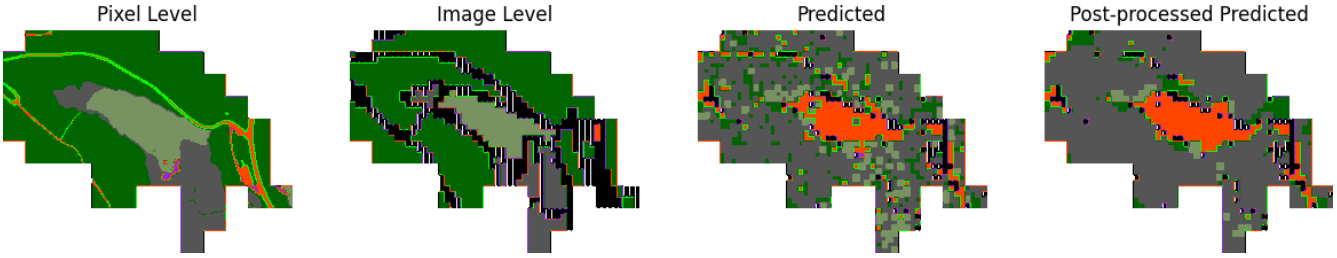

Figure 29: Zone 133 patch size 64, random shuffling, no data augmentation, no pre-trained, post-processed. Two Left images: GT

4.4 Key Results

Let us recap the section experiments in a few lines: In the UNet experiment, the benefits of adjusting date image-annotation mismatch are overshadowed by the consequences of data loss. Stratified shuffling revealed a significant distribution shift, mainly covariate and labels shifts, which were partially addressed by focusing on Mediterranean zones. The NIR channel did not contribute to achieving the target mIoU of 0.5. SAM performed well for delineating roads and trees in zero-shot. The high compute cost of UNet led us to switch to patch classification. Experiments with 256×256 patches showed that a learning rate of 10^{-3} and the Adam optimizer were effective, though generalization to unseen zones remained a challenge. Very high intra-class diversity was suspected by analyzing a few images from the same level 1 class, and was revealed by running t-SNE on extracted features. But the model's performance did not improve significantly by descending to level 2 of the classification; and there was a strong correlation between class distribution and results, indicating the model might be learning distribution patterns rather than distinguishing classes effectively. Finally, the applied post-processing step seems to successfully smooth areas.

5 Discussions

In this section, we will first extend the analysis with the adopted approach. It will cover evaluating class distribution bias in the model predictions, refining dataset quality through expert geo-data filtering, assessing expert performance with restricted information, consider DG perspectives and class unmixing. Then, we will explore new approaches, such as using Species Distribution Models(SDM), expert systems and Large Language Models (LLM).

5.1 Extended Analysis with the Adopted Approach

5.1.1 Evaluating Class Distribution Bias in Model Predictions

It was noted in the subsection 4.3.2 that the model might learn the distribution of the data, such as predicting the forest class 30% of the time simply because they make up 30% of the dataset, rather than discriminating classes based on the pixel values in input. To evaluate this hypothesis, we propose an experiment: to train the model on the unbalanced dataset, and then to test it on an (artificially) balanced validation set. We can count the predictions of Positive for each class and compare it to the training data distribution. If the prediction counts mirror the training set balance, this would confirm our concern.

To address this issue, we suggest several approaches: downsampling the dominant classes like forest, upsampling the minority classes like artificial, combining both techniques, assigning class-specific weights in the cross-entropy loss function, or moving towards the focal loss, which improves performance on difficult examples by reducing the impact of easy examples.

5.1.2 Refining Dataset Quality via Expert Geo-data Filtering

The error may increase for polygons where expert visit on the ground was crucial for accurately delineating their boundaries as pointed out in subsection 2.2.1. To distinguish polygons that are easily delimited with images from those requiring expert ground visits, it would be useful to have access to the geo-located points drawn by experts during the step of polygon delineation. Filtering out polygons that cannot be annotated only from aerial images (polygons with multiple geo-located points) might increase the performances for the frontieer class. Additionally, these geo-located points could serve as prompts to guide the segmentation with SAM.

5.1.3 Assessing Expert Performance with Restricted Data

Until now, we have evaluated the model's performance with the assumption that human performance was achieved with an mF1 score of 1. However, expert performance is based on annotations made with more information than what is provided in the delivered dataset, including access to plant physiognomy and names from field visits. This creates a methodological issue in comparing expert and model performance since the tasks and available information were not the same for both. We believe an experiment is needed to determine how effectively an expert can classify habitats at the level 1 (and possibly level 2) using only VHR images. This would provide clarity on how the model's performance compares to that of experts under equivalent conditions. To address this, a study could be conducted in collaboration with ECOMED experts. For example, we could present 10 samples from each level 1 class to the experts and ask them to classify these samples only by looking at VHR images. We would then compare their classifications with the model's predictions. This comparison would show if experts can do better than the model or if the model performs as well as or better than the experts with the same data.

5.1.4 Domain Generalisation Perspectives

DG, as discussed in subsection 3.3, implicitly assumes that classes are visually consistent across different domains. For example, a dog depicted in both a sketch and a photograph is considered the same semantic

entity, despite the differing domains (sketch vs. photograph). However, this assumption is not validated in our context. In our dataset, a class at level 1 can include sub-classes that have nothing in common with each other from an aerial perspective, such as vineyards and Mediterranean scrublands. Our intuition is that DG methods will not effectively address this challenge as long as classes from the dataset are not consistent. To validate or invalidate this intuition, the POEM method from [15] or Mixtyle from [39] both mentioned in subsection 3.3 could be applied.

5.1.5 Class Unmixing

To create visually consistent classes, we need to reconsider the EUNIS classification hierarchy. Our analysis from subsection 4.3.2 demonstrated that this system does not effectively group visually similar samples from aerial images at level 1. We believe that by sticking with the EUNIS classes at level 1, as they are, we will not improve results much more than what we alrady did. Allowing more flexibility and manually constructing visually consistent classes could improve classification accuracy. However, this approach may result in losing expert semantic knowledge that is implicitly embedded in the original classification system, which is not acceptable. Another alternative is to keep the EUNIS classes and to incorporate implicit information that is not seen in the images we have. We suggest transitioning to a new approach that include additional sources of data to solve the task.

5.2 Exploring New Approaches

We believe that prioritizing natural habitat mapping at a fine level is essential because aggregating habitats at levels 1 or 2 results in the loss of crucial information, such as plant species. Instead of inferring natural habitats directly from the data, we propose dividing the process into two sub-processes: first, determining the plant species names (or their probabilities of presence) for a given polygon, and second, inferring the habitat from these plant species names and additional information about vegetation height. This two-step approach shifts from LC mapping to the domain of SDM.

5.2.1 SDM to Infer Plant Species

SDM can be defined as the task of predicting the probability of a species' distribution in a specific location. This approach, as detailed in [2], involves combining satellite images, both time series and single snapshots with environmental data, including time series data such as temperature and humidity, and single-time variables like soil characteristics. To validate such a model, we could utilize the dataset from the same study [2], which incorporated annotations made by citizens in the collaborative application Pl@ntnet; by giving weights to users according to their level of expertise as described in [20]. Also, we could use the SIMETHIS dataset from [1] to focus on the South of France. The model would apply DL techniques, such as CNN or ViT, to predict plant probabilities in a given polygon.

5.2.2 Expert Systems and LLMs to Infer Habitats from Plant Species

From the set of plant species probabilities, we would integrate additional data on vegetation height, potentially using LIDAR HD data published by the IGN. Combining information on plant species and plant physiognomy, the two key features used by experts to determine habitats, as described in 2.2.1, we could infer the habitat by using an expert system(based on if-else rules), similar to the approach used in [6].

Another alternative to using an expert system is to apply a LLM, as proposed by C. Leblanc [article to be published soon; GitHub code available at: cesar-leblanc/hdm-framework]. To validate the results, we could use the European Vegetation Archive (EVA), which provides the full list of plant species, estimates of cover abundance for each species, location details, and EUNIS classes. The approach would involve using BERT introduced in [7] as a foundational model and fine-tuning it with EVA data. The model

could learn to map plant species to an embedding based on the co-occurences with others plant species in the same location, and then classify these embeddings into natural habitat classes with the final FCL.

Formation and Knowledge Transfer

In the course of this project, I took part in several learning and knowledge-sharing activities to expand my expertise and communicate my work.

- **Formation:**
  - **Field Visit:** Took part in a one-day field visit near Mireval with ecologist from ECOMED Léo Nery, which helped me understand the labelling protocol and the experts needs.
  - **Pl@ntNet Summer School:** Attended a 3 days summer school on "AI and interoperability for agroecology", where I learned about the Pl@ntNet dataset, Species Distribution Modeling (SDM), and the use of large language models (LLMs) for habitat mapping.
- **Knowledge Transfer:**
  - **Lab Presentation:** Gave a 5-minute presentation in English to about 40 lab members, sharing the progress and key findings of my work.
  - **ECOMED Presentation:** Presented a 30-minute talk to the data provider ECOMED partners Pierre Volte and Marie Pisson along with my 4 supervisors, explaining the study's methods and results.

Also, to support transparency and reproducibility, the code for this study is publicly available and documented, following the FAIR (Findable, Accessible, Interoperable, Reusable) principles. Access it at github.com/BertilleT/habitat_mapping. Note that the data is private and can not be shared without explicit consent from ECOMED.

6 Conclusion

In this master's thesis, we tackled the problem of mapping natural habitats using VHR images in the South of France, with the EUNIS classification and data provided by the ecological consulting firm ECOMED. We delivered guidelines to bridge the gap between the original ECOMED dataset and a pre-processed version suitable for automation, which can be applied to other datasets that were not originally designed for DL but are intended to be used with it. We approached the problem as one of LC mapping, focusing on levels 1 and 2 from the EUNIS hierarchical classification system. Several challenges were faced, such as a data shift between study areas, noisy annotations, and significant intra-class diversity. The data shift was partially addressed by restricting the study to the Mediterranean region, and the issue of noisy pixel-level annotations was mitigated by transitioning from a pixel-based classification model to a patch-level one.

That said, the main problem with the adopted approach lies in the aggregation of subclasses, necessitated by the focus on levels 1 to ensure sufficient samples per class. The EUNIS classification was not designed to maintain high intra-class consistency and inter-class diversity at level 1 from an aerial perspective. This aggregation is also problematic because plant species are entirely absent from the results, which makes the findings of limited interest to ecological experts.

To maintain a focus on objectives that are relevant to experts, it seems pertinent to address the problem by targeting fine-level habitat classification. This could be achieved by incorporating environmental data and changing the perspective from LC classification to SDM.

References

[1] Olivier Argagnon, Guilhem De Barros, and Virgile Noble. Simethis-flore-cbnmed - database of southeastern france vegetation. *Vegetation Classification and Survey*, 3:119–120, 2022.

[2] Christophe Botella, Benjamin Deneu, Diego Marcos, Maximilien Servajean, Théo Larcher, César Leblanc, Joaquim Estopinan, Pierre Bonnet, and Alexis Joly. Overview of GeoLifeCLEF 2023: Species Composition Prediction with High Spatial Resolution at Continental Scale Using Remote Sensing. In *CLEF 2023 - Working Notes of the Conference and Labs of the Evaluation Forum*, volume 3497 of *CEUR Workshop Proceedings*, pages 1954–1971, Thessalokini, Greece, Sept. 2023.

[3] Wadii Boulila, Hamza Ghandorh, Mehshan Ahmed Khan, Fawad Ahmed, and Jawad Ahmad. A novel cnn-lstm-based approach to predict urban expansion, 2021.

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP, 06 2016.

[6] Milan Chytry, Lubomír Tichý, Stephan Hennekens, Ilona Knollova, John Janssen, John Rodwell, Tomáš Peterka, Corrado Marcenò, Flavia Landucci, Jiří Danihelka, Michal Hájek, Jürgen Dengler, Pavel Novák, Dominik Zukal, Borja Jiménez-Alfaro, Ladislav Mucina, Sylvain Abdulhak, Svetlana Aćić, Emiliano Agrillo, and Joop Schaminée. Eunis habitat classification: Expert system, characteristic species combinations and distribution maps of european habitats. *Applied Vegetation Science*, 23:648–675, 12 2020.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[9] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.

[10] Anatol Garioud, Stéphane Peillet, Eva Bookjans, Sébastien Giordano, and Boris Wattrelos. Flair# 1: semantic segmentation and domain adaptation dataset. *arXiv preprint arXiv:2211.12979*, 2022.

[11] Anatol Garioud, Stéphane Peillet, Eva Bookjans, Sébastien Giordano, and Boris Wattrelos. Flair# 1: semantic segmentation and domain adaptation dataset. *arXiv preprint arXiv:2211.12979*, 2022.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[14] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[15] Sang-Yeong Jo and Sung Whan Yoon. Poem: Polarization of embeddings for domain-invariant representations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(7):8150–8158, June 2023.

[16] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[18] Nataliia Kussul, Mykola Lavreniuk, Sergii Skakun, and Andrii Shelestov. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(5):778–782, 2017.

[19] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2, 1989.

[20] Tanguy Lefort, Benjamin Charlier, Alexis Joly, and Joseph Salmon. Weighted majority vote using Shapley values in crowdsourcing. In *CAp 2024 - Conférence sur l'Apprentissage Automatique*, Lille, France, July 2024.

[21] Yingnan Liu, Yingtian Zou, Rui Qiao, Fusheng Liu, Mong Li Lee, and Wynne Hsu. Cross-domain feature augmentation for domain generalization, 2024.

[22] J. Louvel, V. Gaudillat, and L. Poncet. *EUNIS, European Nature Information System, Système d'information européen sur la nature. Classification des habitats. Traduction française. Habitats terrestres et d'eau douce.* MNHN-DIREV-SPN, MEDDE, Paris, 2013.

[23] Luca Maggiolo, Diego Marcos, Gabriele Moser, Sebastiano B. Serpico, and Devis Tuia. A semisupervised crf model for cnn-based semantic segmentation with sparse ground truth. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–15, 2022.

[24] Masoud Mahdianpari, Bahram Salehi, Mohammad Rezaee, Fariba Mohammadimanesh, and Yun Zhang. Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery. *Remote Sensing*, 10(7), 2018.

[25] Antonio Mazza, Francescopaolo Sica, Paola Rizzoli, and Giuseppe Scarpa. Tandem-x forest mapping using convolutional neural networks. *Remote Sensing*, 11(24):2980, 2019.

[26] Jose G. Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V. Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530, 2012.

[27] M. Pal. Random forests for land cover classification. In *IGARSS 2003. 2003 IEEE International Geoscience and Remote Sensing Symposium. Proceedings (IEEE Cat. No.03CH37477)*, volume 6, pages 3510–3512 vol.6, 2003.

[28] Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil Lawrence, editors. *Dataset Shift in Machine Learning*. The MIT Press, Cambridge, MA, 2009.

[29] Dillon Reis, Jordan Kupec, Jacqueline Hong, and Ahmad Daoudi. Real-time flying object detection with yolov8, 2024.

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[31] Jonathan V Solórzano, Jean François Mas, J Alberto Gallardo-Cruz, Yan Gao, and Ana Fernández-Montes de Oca. Deforestation detection using a spatio-temporal deep learning approach with synthetic aperture radar and multispectral images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 199:87–101, 2023.

[32] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[33] Phan Thanh Noi and Martin Kappas. Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using sentinel-2 imagery. *Sensors*, 18(1), 2018.

[34] Devis Tuia, Michele Volpi, and Gabriele Moser. Decision fusion with multiple spatial supports by conditional random fields. *IEEE Transactions on Geoscience and Remote Sensing*, 56(6):3277–3289, June 2018.

[35] Zihao Wang and Lei Wu. Theoretical analysis of the inductive biases in deep convolutional networks. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 74289–74338. Curran Associates, Inc., 2023.

[36] Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight sam for mobile applications, 2023.

[37] Chaoning Zhang, Dongshen Han, Sheng Zheng, Jinwoo Choi, Tae-Ho Kim, and Choong Seon Hong. Mobilesamv2: Faster segment anything to everything, 12 2023.

[38] Zihan Zhong, Zhiqiang Tang, Tong He, Haoyang Fang, and Chun Yuan. Convolution meets lora: Parameter efficient finetuning for segment anything model. *arXiv preprint arXiv:2401.17868*, 2024.

[39] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle, 2021.

7 Appendix

Prairies

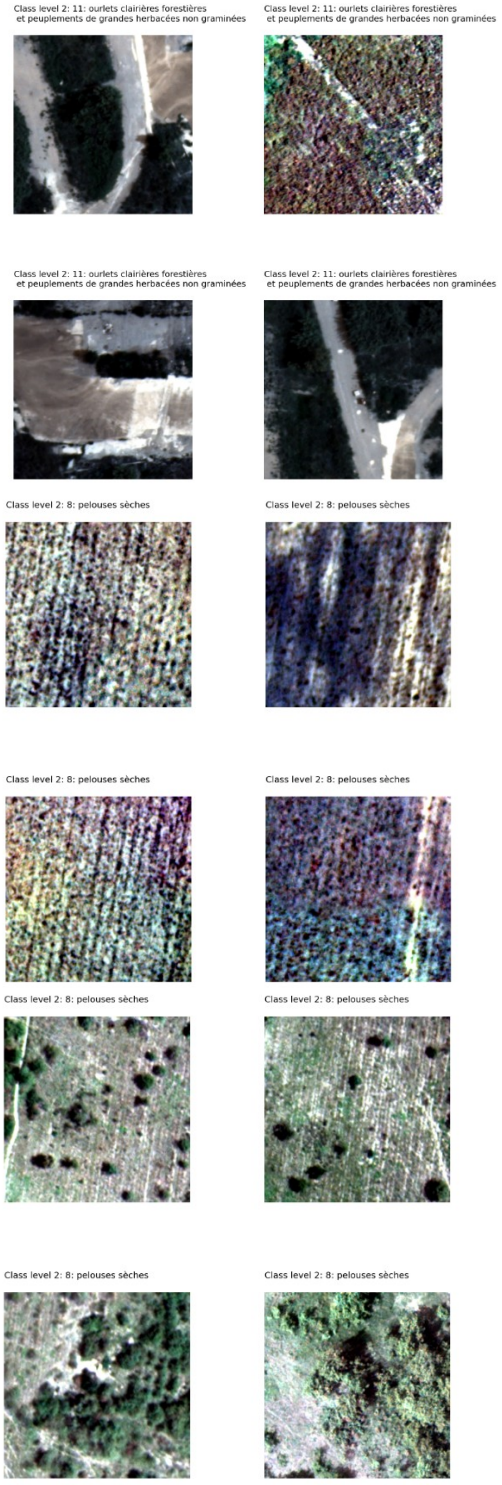

Figure 30: Visual of patches from class 0

Landes

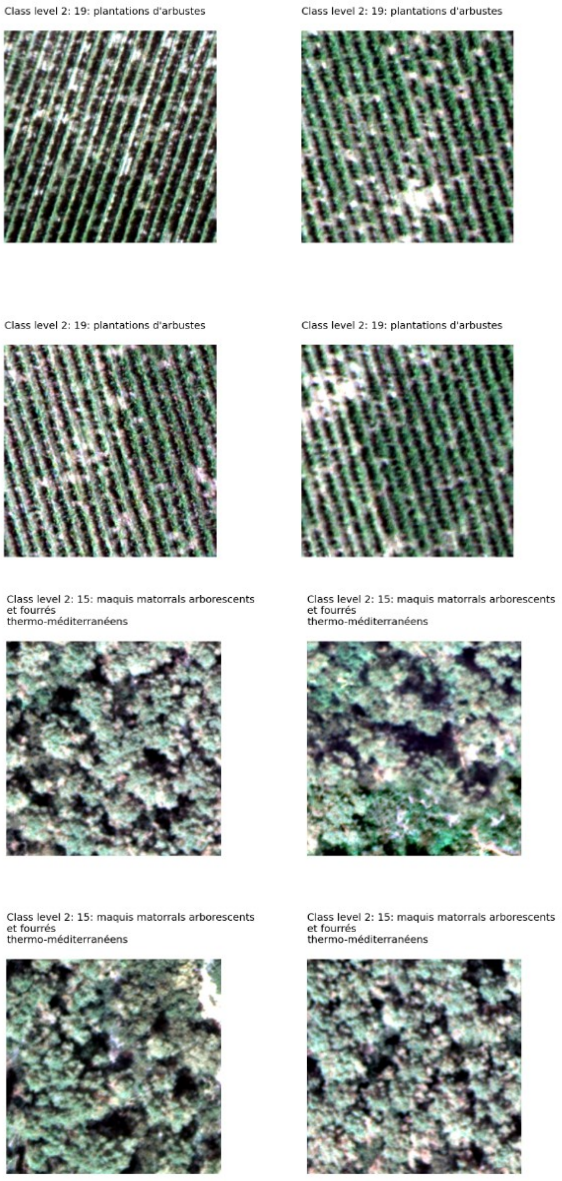

Figure 31: Visual of patches from class 1

Bois

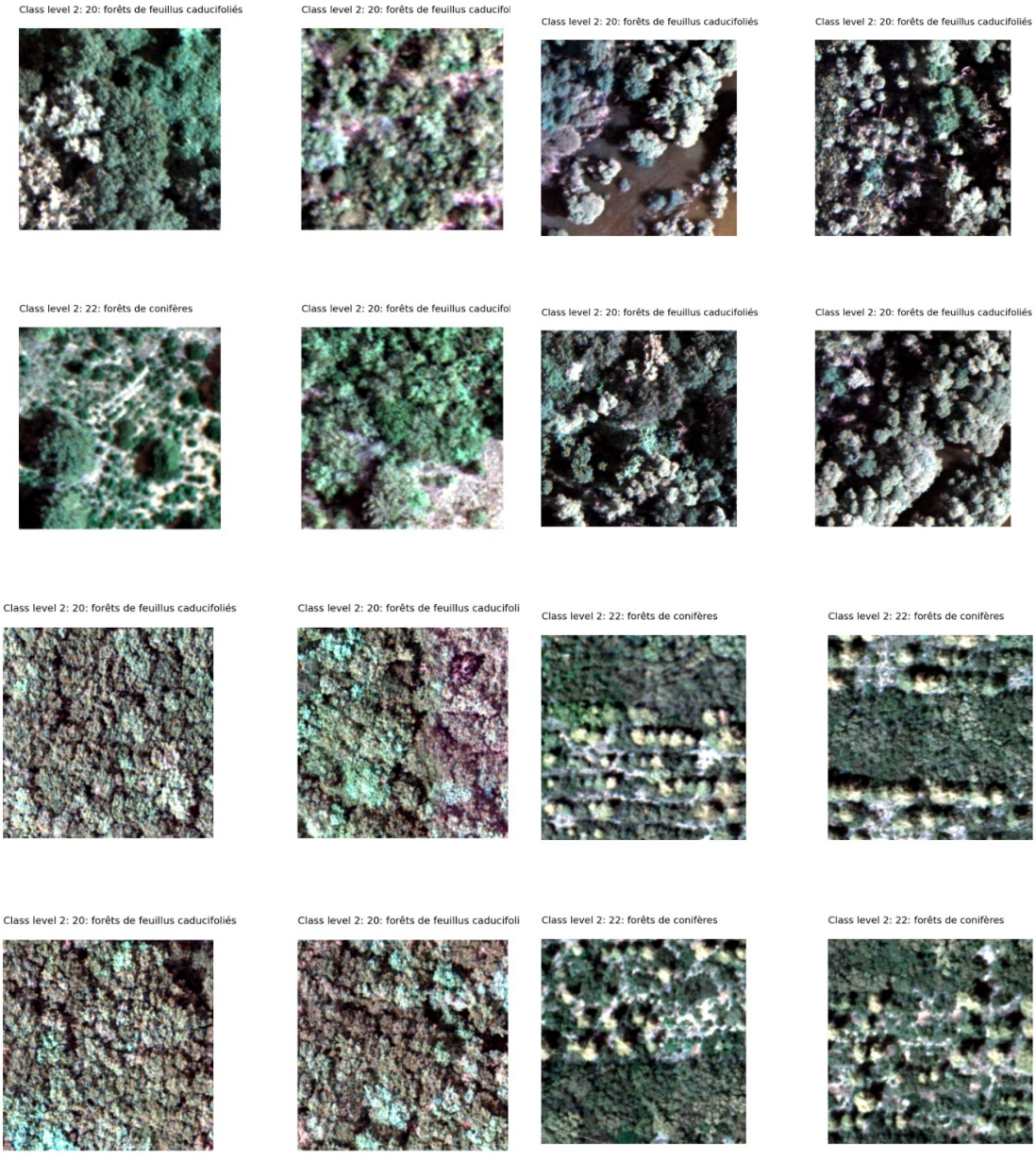

Figure 32: Visual of patches from class 2

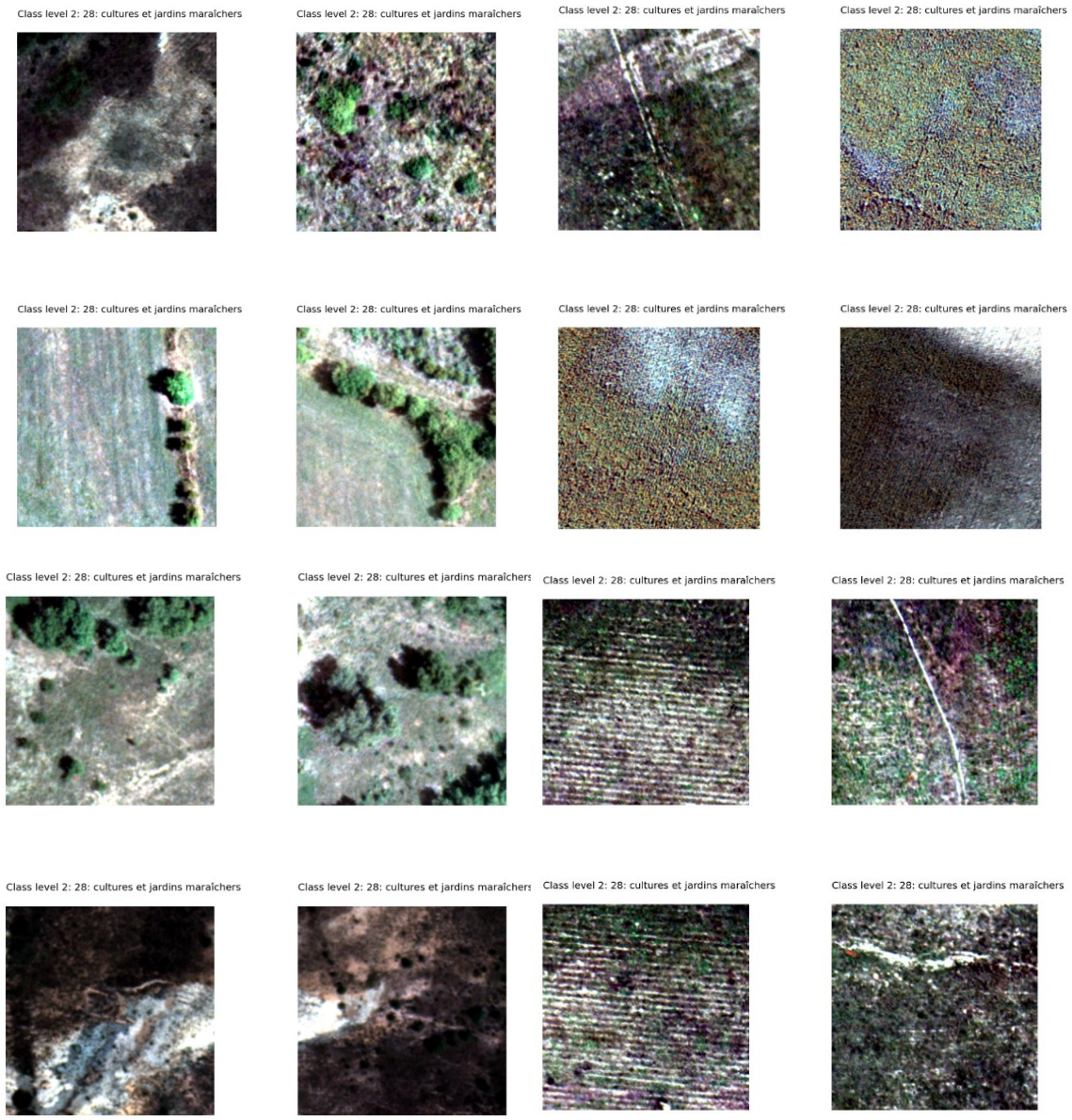

Figure 33: Visual of patches from class 3

Artificiel

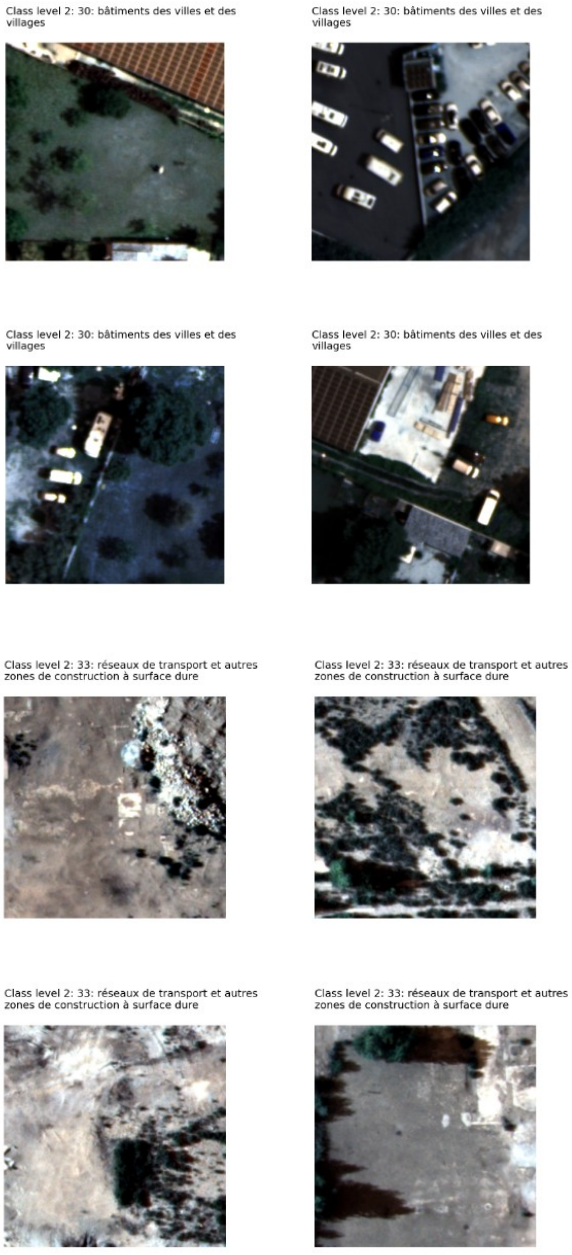

Figure 34: Visual of patches from class 4

Autre

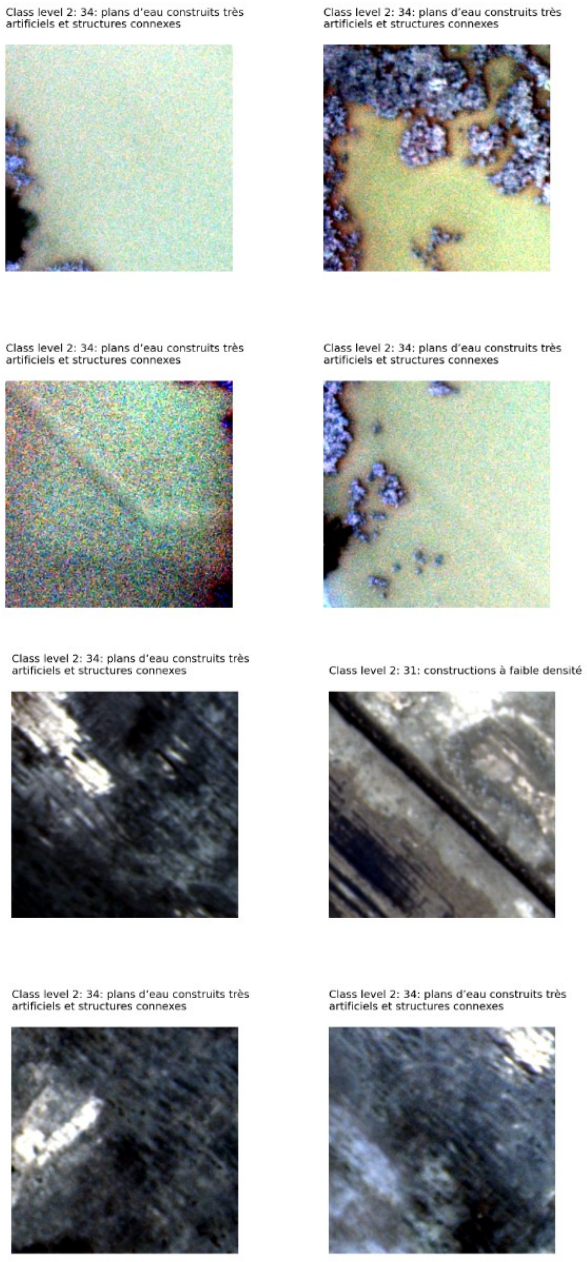

Figure 35: Visual of patches from class 5