Wetland Localization

Using Data Science on Satellite Imagery

Master in Data Science & Artificial Intelligence
Data Science Mission
Group 3

emlyon business school
Abhishek THOMAS
Erwin BYLL
Henry TURNBULL
Siwen LU
Dilara TOYGAR

March 25, 2025

I. Abstract

Wetlands play a crucial role in the environment; they help filter water, support biodiversity, store carbon, and even reduce flood risks. But unfortunately, they are disappearing at a rapid pace, especially because of urban development, farming, and climate change. One of the biggest problems is that in many areas, including in France, we don't really know exactly where all the wetlands are. Field based monitoring is resource consuming and impacted by limited access and seasonal changes. To try and support this project investigates the potential of data science and the use of satellite images.

The project was conducted in collaboration with datacraft who provided us with satellite data and guidance. The main goal was to see if wetlands could be segmented in satellite imagery, for which Sentinel-2 images were provided. The nature of the data allowed for time-series analysis and feature identification beyond the capabilities of the human eye. We built a full pipeline from scratch: starting with data processing, cloud detection analysis (because clouds can badly interfere with satellite images), patch creation, data augmentation, and finally semantic segmentation using different models. Five models were tested: a U-Net trained from scratch, a pre-trained U-Net using fine-tuning, the same pre-trained U-Net but using transfer-learning, a pre-trained DeepLabV3 Plus model using fine-tuning, and that same model but with transfer-learning implemented as well. Due to the imbalance nature of the data, the model performances were evaluated with the mean IoU, while Dice Loss was used as the loss function. Due to the dataset being small and low in quality, the results were not very strong although the fine-tuned models did show the best performance out of the five models used. The fine-tuned DeepLabV3+ model achieved the best score on the test set, however all models showed signs of overfitting. In the end, the project provides a robust and a working pipeline that can already be helpful, and has a lot of potential for improvement.

This project demonstrates potential implementation to support wetland monitoring using satellite images. The results are not very performant for effective implementation currently, however the pipeline does offer a scalable foundation for future research and practical implementation, especially if complemented by additional data and spectral bands.

II. Table of Contents

Table of Contents

III. Introduction & Context of the research

Wetlands help with filtering water, controlling floods, storing carbon, and supporting a huge variety of plants and animals. They do a lot for the planet and play a crucial role in a stable ecosystem required on earth. However, in many places including in France wetlands are disappearing, which poses a big issue. For instance urbanisation, farming, climate change, among many others are putting more and more pressure on these areas. And the reality is, we don’t yet have a clear picture of where all the wetlands are located. That makes it really hard to protect them properly, or even to know when something is going wrong.

Usually, finding wetlands is done by field experts who go out into the area and study it directly. It’s a super valuable method, but requires plenty of work and resources. And it's not always possible, especially in places that are hard to access or in bad weather conditions. Plus, wetlands can change a lot depending on the season. Sometimes they’re full of water, sometimes dry, it really depends on many factors. So, going there just once doesn’t always tell us the full story. That’s why there’s a real need to find other ways to support this kind of work, especially ways that are faster, cheaper, and easier to repeat over time.

That’s where the datacraft wetlands research project comes into play. The target is to understand whether data science can help identify wetlands using satellite imagery. More specifically, Sentinel-2 images were provided by the client and cover a lot of useful information. These images have different bands, which means they capture more than just what the human eye sees. They also highlight things like vegetation and moisture, which are useful when trying to identify wet areas.

The aim is for this project to be a foundation of a solution for monitoring and protecting wetlands in France and beyond. The dataset includes the same image across a span of 36 months allowing for the notion of time to be captured. Because wetlands change a lot depending on the season, this helps the model understand their behaviour better. With time series data, the model becomes more robust and less likely to be fooled by temporary conditions like a rainy week or an unusually dry period.

This project is to support an important field that is crucial for nature preservation and is impactful on countering global warming. Knowing where wetlands are and how they change is important for environmental protection. The methods analysed in this research could be reused in other regions and can be improved upon over time. It’s a starting point for such research, and still has plenty of potential moving forward.

The report starts with some background on wetlands and previous research on similar topics, before explaining the data used and how it was handled, to the modelling part, and how they trained and tested, with evaluation method chosen. Finally, showing the achieved results and discussion on how they can be interpreted and look into what it means in comparison to previous research. We’ll also consider the limitations and challenges faced, and explore the areas of improvement and potential for future research.

IV. Literature review

This literature review explores the different elements required in an image segmentation tasks of satellite images for detecting wetlands. It explores previous literature on similar tasks specifically focusing on the type of data used, data processing techniques, models used and evaluation metrics.

Dataset for Satellite Imagery

Sentinel 1

"Sentinel-1 is an imaging radar satellite at C-band (~5.7 cm wavelength) consisting of a constellation of two satellites, Sentinel-1A and Sentinel-1B, also part of the European Copernicus program created by the European Space Agency (ESA). Their main cover applications are […] Mentoring land surface motion risks; Mapping of land surfaces: forest "(Attema, Davidson et al. 2008, Torres, Snoeij et al. 2012).

It plays a pivotal role in wetland mapping projects. Its ability to penetrate cloud covers and operate even in night conditions make it particularly valuable for monitoring wetlands, which are often located in regions with frequent cloud cover and experience rapid weather changes.

One of the primary advantages of Sentinel-1 is its capacity to provide valuable data even with bad weather conditions. This all-weather capability gives uninterrupted monitoring, which is important for capturing wetlands, as they change frequently. For instance, a study demonstrated that combining Sentinel-1's radar data with optical data from Sentinel-2 achieved an overall classification accuracy of approximately 90% in wetland mapping.

One example of Sentinel-1's utility in wetland mapping is shown in a study focused on the Great Lakes region. In this study, researchers used Sentinel-1 datasets platform to produce a detailed wetland map. This approach utilized an object-based supervised machine learning classification workflow, with an overall accuracy of 93.6%. Specifically, wetland classes were identified with an overall accuracy of 87%/ This shows the effectiveness of SAR for wetland classification (*(PDF) Sentinel-1 and Sentinel-2 Data Fusion for Mapping and Monitoring Wetlands*, 2024)

However, the use of Sentinel-1 data is not without challenges. The preprocessing of radar data can be complex and computationally intensive, requiring specialized software and expertise. Additionally, radar signals can be affected by surface roughness and vegetation structure, potentially leading to misclassifications. For example, areas with dense vegetation might exhibit similar backscatter characteristics to open water, complicating the differentiation between these land cover types citeturn0search1.

Despite these challenges, the integration of Sentinel-1 data into wetland mapping projects offers significant benefits. Its ability to provide timely and reliable data enhances the monitoring and management of these critical ecosystems. Although, in several articles it is stated that Sentinel 1 and Sentinel 2, work in partnership, to capture the best results. The integration of both data sources has been shown to enhance classification accuracy, particularly in regions with frequent cloud cover where optical data alone may be insufficient.

Sentinel-2: Why It Is Used, Strengths, and Limitations

One of the things that makes Sentinel-2 really useful is that it captures images in thirteen different spectral bands. That includes both visible light (like what we see with our eyes) and shortwave infrared, which gives extra information that we normally can't see. These bands are super helpful when it comes to understanding what kind of land we’re looking at like forests, water, or wetlands. Wetlands in particular have their own kind of “spectral signature,” meaning they reflect light in a specific way depending on how wet or vegetated they are. Some studies (Campos-Taberner et al., 2020) emphasize how these bands improve the ability to differentiate wetlands from other land cover types. The use of shortwave infrared bands is especially useful in distinguishing between dryland and wetland vegetation, which can appear similar in visible light (Amani et al., 2017).

However, Sentinel-2 isn’t perfect. One of the main problems is that it only uses optical imagery, which basically means it needs clear skies to work properly. If there are clouds, they can cover parts of the image, and that makes it really hard to use the data. In places where it's often cloudy, this becomes a big issue, because we can’t get consistent observations over time. And that’s especially annoying when we’re trying to monitor wetlands, since they need to be tracked regularly to really understand how they change. To deal with this, some researchers have tried using cloud-masking methods, which are basically ways to detect and remove the cloudy parts of the image. Others suggest combining Sentinel-2 with Sentinel-1, which is another satellite that uses radar instead of optical sensors. The cool thing about radar is that it can see through clouds, so it works even when the weather isn't great. That's why combining both satellites can be super helpful, it gives a more complete and reliable view of the wetlands, even in bad weather (Hosseiny et al., 2021).

Another limitation is the trade-off between spectral and spatial resolution. The satellite gives really good spatial resolution for some of the bands, like the visible ones and the near-infrared, with 10 meters per pixel, which is pretty detailed. But for the shortwave infrared bands, the resolution drops to 20 meters per pixel, which means we lose some detail. This lower resolution can make it harder to see smaller wetland features, especially when they're narrow or mixed with other land types. Also, before we can even use the data for machine learning, there's a lot of preprocessing that needs to be done. Sentinel-2 images don't come ready to use. We need to apply atmospheric corrections, geometric adjustments, and radiometric calibration. Basically, we have to clean and align the data properly to make sure that what we see is accurate and that the images can be compared over time. This step is super important, especially when we're working with time series and training models that are sensitive to small variations in the input (Rumora et al., 2020).

Time Series Imagery: Why It Improves Classification Accuracy

Using time-series imagery has been shown to really help when it comes to classifying wetlands more accurately. The thing is, if you only use a single image taken at one moment in time, you just get a snapshot and that doesn't always reflect how the landscape truly behaves. Wetlands are very dynamic environments. They change a lot depending on the season, with vegetation growing or disappearing, water levels rising or dropping, and even the soil moisture changing. Because of all these seasonal variations, it's easy to misclassify a wetland if you only rely on one image. For example, something that looks dry in the summer might actually be a wetland that just happens to be less active at that time. That's why having access to multiple images taken at different times of the year makes a big difference. It helps the model learn the seasonal patterns of wetland areas, and that way, the classification becomes more stable and reliable over time(Temporal Convolutional Neural Network for the Classification of Satellite Image Time Series, n.d.).

One of the main reasons why using time-series imagery improves classification is because it helps capture how wetland vegetation changes throughout the year. Wetlands follow seasonal cycles, and some types of plants grow more during certain months and then slow down or become dormant during others. If we only use one image, we might catch them at a time when they don't look like wetlands at all which can lead to mistakes. In the study Understanding Deep Learning in Land Use Classification Based on Sentinel-2 Time Series(Campos-Taberner et al., 2020). The researchers showed that deep learning models trained on time-series images performed much better than models trained on just one image. That's because time-series data helps the model understand how wetlands change over time, which makes it easier to tell them apart from other vegetation that might look similar in some seasons. According to the same study, summer images tend to be the most useful, since the differences between land cover types are more obvious during that time of year.

Time-series imagery also helps avoid mistakes that can happen because of temporary environmental conditions. For example, after it rains a lot, some areas that are usually dry might look flooded in a satellite image, and that can make the model think it's a wetland when it's not. The opposite can happen too during dry seasons, actual wetlands might seem less visible or look like normal land. If we only use one image, we might misclassify these cases. But by looking at several images from different times of the year, models can better understand which areas are real wetlands and which ones just change temporarily. A study called WetNet (Hosseiny et al., 2021) showed that combining data from different dates and even from different satellites like Sentinel-1 and Sentinel-2 helps a lot in detecting the real shape of wetlands and how they evolve over the seasons.

Another important advantage of time-series imagery is that it helps make the classification models more reliable. It means the models won't be too sensitive to environmental conditions that change depending on the season. For instance, some wetlands are much easier to detect during summer, because that's when the vegetation is more developed and water bodies stand out more clearly in the images. A study called *Coastal Wetland Classification with Deep U-Net Convolutional Networks and Sentinel-2 Imagery* showed that models trained on summer data tend to perform better, since the contrast between wetlands and other land types is stronger at that time. But when it comes to wetlands that flood only during certain times of the year, it's really important to include images from different seasons. That way, the model can learn the full water cycle and doesn't miss important wetland features that only appear for a few months(Dang et al., 2020).

Data processing for image classification or segmentation tasks

Spectral bands and indices

Bands are a crucial part of preparing data for satellite imagery especially in vegetation detection and are measured in nanometers (nm) (Liu et al., 2021). Sentinel-2 imagery being equipped with multi-spectral instruments means data can be captured across 13 spectral bands with between 10 to 60-meter pixel size Sentinel-2 carries the Multispectral Imager (MSI). This sensor delivers 13 spectral bands ranging from 10 to 60-meter pixel size with the channels blue (B2), green (B3), red (B4), and near-infrared (B8) all with a 10-meter resolution. While a 20 meter ground sampling resolution is found in red edge (B5), near-infrared (NIR) (B6, B7, and B8A), and short-wave infrared (SWIR) (B11 and B12) bands. Finally the 60 meter pixel size is with coastal aerosol (B1) and cirrus band (B10). (GISGeography, 2019).

When we look into the effects of bands sentinel-2 imagery for vegetation, marsh or wetland detection there are plenty of examples. With sentinel 2 images It was shown that using Multispectral bands improved the detection of vineyards in satellite imagery segmentation in comparison to a single high-resolution band, and the importance of its implementation stated to differentiate vegetation based attributes (Liu et al., 2021). The Red band is the one that is most used in the analysis of vegetation due to its sensitivity to chlorophyll absorption.Vegetation easily absorbs the red light, so the areas with healthy plants will have a light red reflectance, and barren lands or water surfaces will have a red reflectance much higher(Ji et al., 2009). In the research from Jones et al. (2020) the multispectral bands used are the following: coastal (400–450 nm), blue (450–510 nm), green (510–580 nm), yellow (585–625 nm), red (630–690 nm), red edge (705–745 nm), NIR1 (near-infrared 1, 770–895 nm), and NIR2 (near-infrared 2, 860–1040 nm) (Jones et al., 2020).

The NIR band is shown to be fitting when considering vegetation and distinguishing land from water achieving higher accuracy (Gasteiger et al., 2014). A study conducted using recurrent neural networks designed to analyze sequential data on Sentinel 2 imagery showed that red and near-infrared (NIR) bands are the most important for distinguishing land use types. It founds these bands to be particularly effective for analyzing vegetation health and soil composition (Campos-Taberner et al., 2020). Regarding marsh detection the accuracy of deep-water marsh vegetation classification gradually increased as spatial resolution improved with the NIR band. However, what strongly improved marsh vegetation classification accuracy was combining these spectral bands with spectral indices. (Liu et al., 2021).

The two main spectral indices looked into for this report are NDWI and NDVI as they are fitting in Sentinel-2 imagery. NDVI is a combination of Red (B4) and NIR (B5) band (NDVI=Band 5 – Band 4/Band 5+Band 4), and is generally good at differentiating vegetation from non-vegetation areas (Shukla et al., 2021).

(Jones et al., 2020) found that adding NDVI to the greyscale bands of their images for vegetation identification based on the U-net model, increased the accuracy from 83% to 84%. However, this combination performed a little worse than the Multispectral bands in satellite image segmentation (Jones et al. (2020). In the case of Kaplan & Avdan (2017) regarding specifically identifying information related to wetlands, NDVI was not capable, but was able to identify vegetation present within the wetlands. In contrast (Liu et al., 2021) found that adding spectral indices with NDVI, NDWI and EVI improved accuracy in distinguishing marsh vegetation types.

Another important index is the Normalized Ratio Water Index (NRWI), which refines NDWI by differentiating deep and less deep water bodies (Xie et al., 2018). This is particularly useful for wetlands where seasonal water depth variations occur, helping models better distinguish between temporary and permanent water bodies. Finally, choosing the right spectral bands and indices is crucial for improving wetland classification accuracy. While RGB bands provide a basic visualization, NIR, SWIR, and indices like NDWI, MNDWI, and NRWI offer deeper insights into wetland hydrology and vegetation health, leading to more precise classification results.

Classification, Detection vs Segmentation

Image classification refers to classifying a predefined label to an entire image and determining whether it appears there or not. Object detection refers to identifying and locating different things within an image, while image segmentation identifies the entity in an image and assigns each pixel to a label (Image Classification vs Image Segmentation vs Object Detection | Kaggle, n.d.). For satellite imagery most often classification and segmentation methods are used for identification, with classification easiest to achieve but segmentation giving clearer results

Rasterization

Unlike vector data, rasters are made up of pixels or grid cells. Further to that, they are usually regularly spaced and square (but they don’t have to be) and often look pixelated.

Raster Overview

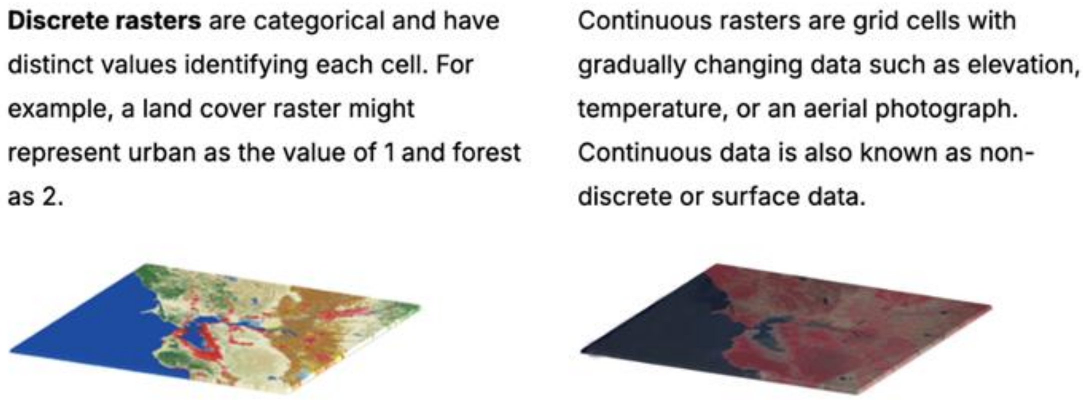

Figure 1: Raster Overview (GISGeography, 2015)

A geospatial raster is accompanied by spatial information that connects the data to a particular location. This includes the raster's extent and cell size, the number of rows and columns, and it's coordinate reference system (or CRS).

Rasterization is the process of converting vector data into raster format. This involves overlaying a grid onto vector features and assigning values to the resulting pixels based on the underlying vector data. Rasterization is useful when integrating vector features into raster-based analyses or when preparing data for certain image processing techniques that require raster inputs. The process is the following.

- **Defining a Grid:** Overlaying a grid with a specified cell size over the vector data. The resolution of the resulting raster is determined by the cell size; smaller cells yield higher resolution.
- **Assigning Values to Cells:** Determining how each cell's value is derived from the vector data it overlaps. Common methods include:
- **Binary Encoding:** Assigning a value (e.g., 1) if the cell is occupied by a vector feature and another value (e.g., 0) if not.
- **Attribute Mapping:** Assigning cell values based on specific attributes of the vector features, such as land use type or elevation.
- **Proportional Allocation:** Assigning values based on the proportion of the cell covered by different vector features.
- **Handling Overlaps:** Implementing rules to manage cases where multiple vector features overlap a single cell, such as prioritizing features based on a hierarchy or combining attribute values. (Barrie, 2022)

Shapefiles

Shapefiles have a central function in image segmentation and satellite imagery as a gateway to geospatial analysis. They facilitate machine learning model training by providing annotated datasets and enabling integration of segmentation results into GIS platforms to conduct additional analysis. For instance, YOLOv8 has been used in building segmentation in satellite imagery with results being converted to shapefiles to be utilized in spatial applications. (Alosius, 2023) The Segment Anything Model (SAM) also enables geospatial analysis through segmentation of satellite imagery and providing results in shapefiles. Shapefile format does have a limitation in processing large and complex datasets. Despite this limitation, shapefiles have a central function in transforming satellite imagery to usable geographic information.(*Segmenting Satellite Imagery with the Segment Anything Model (SAM) || Image Segmentation Using AI*, 2024)

Augmentation techniques used (resizing vs padding)

In the domain of imagery data augmentation, resizing and padding are fundamental techniques used by actors to standardize the input dimensions for machine learning models. Resizing involves scaling images to a uniform size, ensuring compatibility with model structure. This method maintains the aspect ratio, but excessive resizing can lead to loss of mandatory details, potentially reducing model performance. On the contrary, padding adds pixels to images, preserving original dimensions without altering content. While padding maintains spatial information, it introduces non-informative regions that models must learn to ignore, which could increase the training complexity. (*Augmenters.Size — Imgaug 0.4.0 Documentation*, n.d.)

In the context of wetland mapping projects, such as the DataCraft Wetlands Project, these augmentation techniques could be relevant. Wetland imagery often varies in resolution because of many aspects, such as: different satellite sources or environmental conditions. As a consequence, implementing resizing ensures uniform input sizes, facilitating efficient processing and model consistency. However, it is crucial to avoid excessive resizing that could hide important details for accurate wetland boundaries. Padding, on the other hand, keeps original image features but requires models to distinguish between actual content and padded areas, which could toughen the learning processes.

Clouds

Detecting clouds is a really important step when working with satellite images, especially with Sentinel-2. That's because clouds can hide parts of the land and make it harder for the model to classify things correctly. There are some traditional methods like Fmask, Sen2Cor, and MAJA that are used to spot clouds by looking at specific spectral thresholds. But these methods aren't perfect; they can get confused when the image has bright surfaces like snow or sand, which sometimes look too similar to clouds and end up being misclassified(Baetens et al., 2019).

To get better results, more recent methods based on deep learning have been developed like Convolutional Neural Networks (CNNs) and Fully Convolutional Networks (FCNs). These models are trained on large datasets and can adapt to different types of weather or atmospheric conditions. Because of that, they're usually much more accurate than the older, traditional methods when it comes to segmenting clouds in satellite images(Li et al., 2019). Multi-temporal filtering also helps a lot with cloud detection. The idea is to use images taken at different times and then keep only the clearest pixels from each one to create a sort of cloud-free version of the scene. This method is especially useful when monitoring wetlands, because it helps get a more complete and clean view of the area, even if some images were originally covered by clouds (Gao, 1996).

Sentinel-1 offers a good alternative because it uses radar instead of optical sensors, and radar waves can go through clouds. That means we can still get useful data even when the weather isn't clear, which allows for more regular and consistent monitoring. Some studies (Fuentes-Peñailillo et al., 2018), showed that combining data from Sentinel-1 and Sentinel-2 leads to better classification results. So this mix of both satellites works really well, especially in areas where cloud cover is a common problem.

Python libraries such as Rasterio for raster data manipulation, GeoPandas for geospatial analysis, OpenCV for image processing, and Albumentations for data augmentation play key roles in automating cloud detection and improving satellite image quality.

Libraries

Python libraries are really important when it comes to working with satellite images. They make it a lot easier to handle, analyze, and improve remote sensing data, especially for tasks like classifying wetlands. Some libraries are especially helpful for managing raster and vector files, doing image processing, or applying data augmentation to improve the performance of machine learning models(Gorelick et al., 2017).

Rasterio is really useful when working with geospatial raster data like Sentinel-2 images. It lets us read, write, and transform raster files easily, which saves a lot of time when handling satellite data. For example, we can use it to extract specific spectral bands, change the projection of an image, or mask out parts we don't need, all of which are important steps before using the data in a classification model. On the other hand, GeoPandas builds on top of Pandas and adds features to work with geospatial vector data. It's especially handy when dealing with shapefiles, like the ones showing wetland areas. With GeoPandas, we can do spatial operations like overlaying the shapefiles on top of raster images, which helps us match the satellite data with the actual wetland boundaries(Clewley et al., 2014).

OpenCV, also known as CV2, is a library that's widely used for image processing. It offers a lot of helpful tools like filtering, edge detection, and contrast enhancement. These functions are really useful when working with satellite images because they help clean up the data by reducing noise and making important features stand out more clearly(Gangal et al., 2021). Albumentations is a really useful library when it comes to deep learning, especially for image classification tasks. It allows us to apply data augmentation techniques like rotation, scaling, or flipping, which helps create more varied training datasets. This makes the model more robust because it learns to recognize wetlands even when the images look slightly different, for example, due to changes in orientation, lighting, or other conditions (Buslaev et al., 2020).

By using all these libraries together, the whole remote sensing workflow becomes much more efficient. They make it easier to prepare the data properly, which is super important for getting good results. In the end, this helps improve the accuracy of the classification models, especially when it comes to environmental monitoring tasks like wetland detection.

Modelling and Evaluation

This part of the review will look into the models that can be used for image segmentation tasks for satellite imagery. It will first compare literature on machine learning versus deep learning, before considering specific models often used for segmentation tasks in satellite imagery, and finally also looking into transferred learning and the documentation of two specific pre-trained models.

Machine Learning vs Deep Learning image segmentation

Looking at a 40 year Meta study on wetland classification showed that for past wetland mapping studies machine learning models such as random forest, support vector machines and decision trees were mainly used and achieved good accuracy, due to their ability at handling diverse features contributing to better distinction with random forest often performing best. However, from around 2015 onwards more and more deep learning models like CNNs, U-Net architectures and transformer-models being used, due to their ability to better extract spatial and spectral features. It showed that specifically when combined with Sentinel 1 and Sentinel 2 images superior classification performance could be achieved with CNNs, however the common challenge of class imbalance was found with wetlands often getting under represented and leading to higher miss classification result (Mahdianpari et al., 2020)

Looking into a time-series example, the study "Understanding Deep Learning in Land Use Classification Based on Sentinel-2 Time Series" explores how deep learning models can be used to classify land use based on time-series data from Sentinel-2 images. The research focuses on improving the interpretability of machine learning models by identifying which spectral bands and seasonal periods contribute the most to classification accuracy. By applying a recurrent neural network designed to analyze sequential data, determine which bands of the electromagnetic spectrum and which time periods provide the best information for classifying different types of land cover. The BiLSTM model achieves strong classification performance, demonstrating that analyzing time-series data improves accuracy compared to using single-date images (Campos-Taberner et al., 2020). Interestingly a later study on using sentinel 1 and sentinel 2 images for image classifcatoin in France, found that a random forest model achieved the highest Kappa coefficient and overall accuracy with a similar F1 score to CNNs. However the authors do suggest that this lower CNN performance can be due to the image patches used in training of (16x16), as well as the small number of images in the time series data used (Le Guillou et al., 2023).

U-Net

The Unet model is a fully convolutional network that can achieve precise segmentations with few image inputs (Ronneberger et al., 2015). It is often used for Sentinel-2 image classification and segmentation tasks. By downsampling it captures context information, while the decoder then up samples to ensure spatial resolution and the skip connection ensures different levels of fusion of the feature mapping reducing information loss when downsampling (Yao & Jin, 2022). The U-Net has a relatively small model size due to the number of learned parameters (Jones et al., 2020). In a specific segmentation study it showed decent performance levels despite CNNs difficulties with medium scale imagery used. It was however outperformed by the Swin Unet method that could better capture the information (Yao & Jin, 2022). In comparison to a DeepLab-V3 model it performed equally for simple binary segmentation tasks in crop detection (Jones et al., 2020).

In the use of Unet for segmenting different land-cover types in satellite imagery it performed better on a VGG backbone rather than a ResNet50 backbone. However, lightweight backbones like VGG fit faster with stable accuracy increase, with deeper networks fitting slower with fluctuation in accuracy, which could be a reason for this difference (Yao & Jin, 2022)

DeepLabV3 Plus

First looking into DeepLabV3, a network utilising a spatial pyramid pooling module with the Atrous Spatial Pyramid Pooling (ASPP). This is enhanced by adding image level features further contributing to encoding of the entire context, capturing multi-scale contextual information for semantic image segmentation. The final feature map is a powerful encoder output with rich semantic information (Chen et al., 2018). This is extended with the DeepLabV3 Plus by adding a simple decoder module ensuring refinement in segmentation on the object boundaries in turn improving segmentation results (Chen et al., 2018). As Liu et al. (2021) found, the DeepLabV3 Plus model is strong for vegetation classification on Sentinel-2 imagery and is especially good at dealing with multi-scaling problems of image segmentation. They found that the Atrous Spatial Pyramid Pooling (ASPP) module upsampling after a 3 x 3 convolution restored spatial information resulting in pixel-level prediction. However, in the case of (Jones et al., 2020) it was found that DeepLabV3 showed no benefits in a more simple satellite imagery segmentation task, but it should be reflected that they did not analyse the effects of DeepLabV3 Plus.

Transfer Learning and Fine-tuning Models

With the aim of improving performance and generalisation on related tasks, transfer learning tries to achieve it by using pre-trained models allowing previously gained information to be used for the benefits of similar future tasks and is particularly helpful for deep learning tasks where a lot of data is needed for producing meaningful outputs. It uses a pretrained model and then adapts the final layer or layers to adapt for the new task, whereas fine-tuning simply uses the entire model as it is. The major benefits come from reducing computational costs, requiring smaller dataset size and due to the retraining on multiple datasets it is very generalizable. However it is most effective when the tasks do not have to much variety, the source and target datasets are similar and the comparable model can be applied for both tasks (What Is Transfer Learning?, 2024)

The study "Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks" compares transfer learning models to deep learning models used without transfer learning, for the semantic segmentation task of mapping slum areas. The transfer learning model used is a Fully Convolutional Network (FCN) which is based on the architecture of the VGG19 CNN. To be able to use this for semantic segmentation (slum mapping), the final classification layer was replaced with a 1x1 convolution and incorporates skip connection combining predictions from different layers. For Sentinel-2 imagery, the results do show a significant increase in accuracy when applying transfer learning from the QuickBird-trained FCN compared to the FCN trained directly using ImageNet weights (Wurm et al., 2019). In a final step of exploring transfer learning and fine-tuning the structure of two pretrained models based on PyTorch models will be delved into (Qubvel-Org/Segmentation_models.Pytorch, 2019/2025).

The PyTorch U-Net segmentation model fine-tuning uses the previously mentioned U-Net fully conventional neural network architecture and is designed for semantic image segmentation. The encoder part extracts increasingly abstract features and the decoder recovers spatial details moderately, with the skip connection between these layers allowing extraction of the fine-grained details. This way important spatial information can be preserved (Segmentation Models — Segmentation Models Documentation, n.d.). DeepLabV3 Plus is an alternative to U-Net that uses Atrous Separable Convolution for Semantic Image Segmentation. (Segmentation Models — Segmentation Models Documentation, n.d.). All encoders have pretrained weights meaning preparing data the same way during the weights pre training can lead to better results. The encoders already come with pre-trained weights allowing for speedier and more robust convergence in segmentation model training and lightweight or high-capacity encoder can be used depending on preferences. The depth allows to specify the quantity of encoder downsampling operations. To take it further one can then adapt the final layer to fit the new dataset and make it a transfer learning model (Qubvel-Org/Segmentation_models.Pytorch, 2019/2025).

Evaluation Metrics and Loss Function

The Intersection over Union (IoU) is the most common metric for segmentation evaluation, and it measures the area of overlap (TP) over the area of Union (TP+FP+FN) and calculates how well a prediction matches the ground truth (Demir et al., 2018). It is less sensitive to class imbalance than accuracy and is therefore crucial in imbalanced segmentation tasks like wetlands. This can then be broken down into the background IoU measuring how well the model segments backgrounds and the wetland IoU measuring how well wetlands are segmented. The mean IoU ensures both classes are considered equally and is often used as the final metric in image classification, and image segmentation tasks (Demir et al., 2018; Long et al., 2015). As the mean IoU involves calculating the IoU for each class, and the IoU calculation requires the confusion matrix, all these elements are part of the evaluation. When considering a stronger imbalance of the dataset a mean IoU of 0.6-0.7 can be considered a good performance.

Regarding the loss function, typical loss functions such as cross-entropy are not great at dealing with class imbalance allowing the background to dominate (Cabezas & Diez, 2024). This is where the Dice Loss can come in, as it is calculated (1-(2*TP/(2*TP+FP+FN)) measuring the overlap between predicted and ground truth segmentation tasks laying a focus on the correctly segmented area (Lima et al., 2023).

V. Methodology

In the coming section we will explore the methods used in this project to solve the below research question:

How can satellite imagery and image segmentation models be used to generate an initial assessment of wetland presence in France to support more efficient environmental monitoring?

The Methodology is broken into data and exploratory analysis, data processing and feature engineering, the modelling section, and finally the evaluation section.

Data and Exploratory Data Analysis

This part of the methodology incorporates everything from the data provided to the details of that data. It considers data specifics and how to combine the satellite images with the given shapefiles to then create new images that can be best used for wetland segmentation.

Provided Images and Shapefiles

The raw data provided for this project contains sentinel-2 satellite images, wetland shapefiles and the study area shapefiles. Made up of 36 separate images, each Sentinel-2 image shows the same area in France containing wetland and non-wetland surfaces. The data was provided by the client and which data was to be used was decided from the client's side. From different possible quality levels of Sentinel-2 images the provided data has the lowest spatial resolution with 62 pixels height and 357 in width with one pixel making up just under 54.85 meters squared. For overall details on the sizes see the table 1 on page 22. Each image provided represents the satellite image of each month, spanding January 2021 until December 2023, that contained the least cloud interference of that month. Therefore, the notion of time is captured in the images, allowing for chronologically ordered dates. The wetland shapefile shows the exact outline of the wetland for that specific area, whilst the study area shows the examined area for wetlands conducted for this research. For this project the client informed us to focus on the wetland area shapefile and not necessarily to consider the study area.

RGB image, RGB image adapted, Shapefile

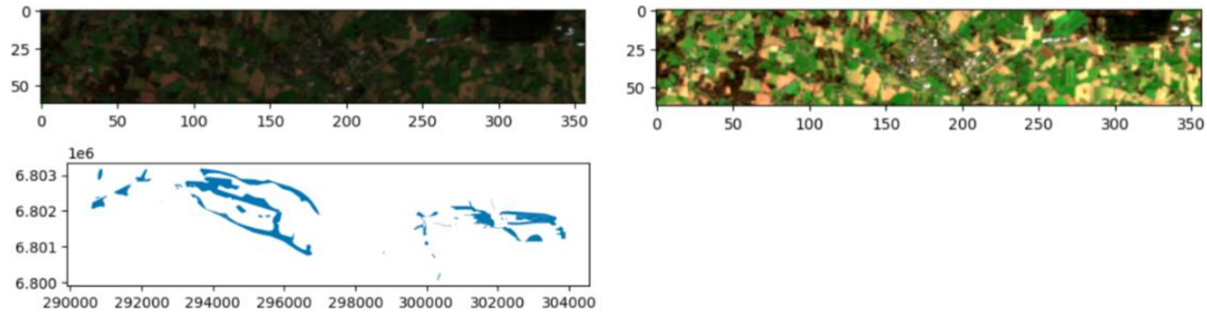

Figure 2: RGB image, RGB image adapted, Shapefile

The image datafiles stored in a GeoTIFF (.tif) format also contained separate geospatial metadata information of width, height, band-quantity, coordinates, and EPSG coordinate system value among many others. This metadata also contains information on the individual bands, however regarding these 36 images from the 10 bands available there is only metadata on the three RGB bands in red, green and blue. This information includes the statistics maximum, statistics mean and statistics minimum as well as the standard deviation. The Rasterio library was used to open the GeoTIFF files, after which each of the bands had to be normalised ensuring alignment and comparability. Finally by stacking the three RGB bands on top of eachother the full coloured satellite image could be generated, as can be seen in the example figure 2 above. The lighting could then be adapted also depicted in figure 2 which is clearer for the human eye.

Metadata was also included in the shapefiles, including shape length, shape area as well as geometry which informs of the coordinates in the shapefiles coordinate reference system (CRS). By plotting it, the above shapefile in figure 2 is depicted, which can be plotted over each of the 36 images available. To open the shapefiles geopandas (gpd) was used.

Overlaying, Mask Creation and Projection

On top of this a wetland mask was also created by filling the inside part of the shapefile, allowing for the entire wetland space of pixels to be covered by the mask. Having access to the shapefiles fitting to satellite images allows to lay the wetland shapefile over each satellite image showing the outline area the wetland is expected to be in. This could be achieved by firstly placing a raster on the satellite images allowing to locate the areas on the images and allow the coordinate reference system (CRS) with the EPSG code to be matched on the image with the shapefile. In order to add the shapefile and the mask on top of the various satellite images, the overlap needed to be checked. When it could be identified that the EPSG code was not matching, the EPSG code of the shapefile could be adapted to match the code of the satellite image. Furthermore simple overlap checks of the wetlands on the images were made comparing the bounds projections of the two. This allowed reprojecting the shapefile and mask considering its CRS to match the rasters CRS on the image. Plotting the wetland boundaries and creating the wetland mask allowed to overlay the shapefile mask (in red) onto the images precisely, ensuring correct placement of the shapefiles onto the wetland area in the satellite image as exampled in figure 3 below. This was iterated over all images, allowing for visualisation. However to ensure simple and efficient visualisation of each image considering the various bands, shapefiles and masks, a widget was created allowing for plotting on top of each other. Due to the ordering of all images by date, the notion of time could also be captured.

Sentinel-2 image with shapefile overlay

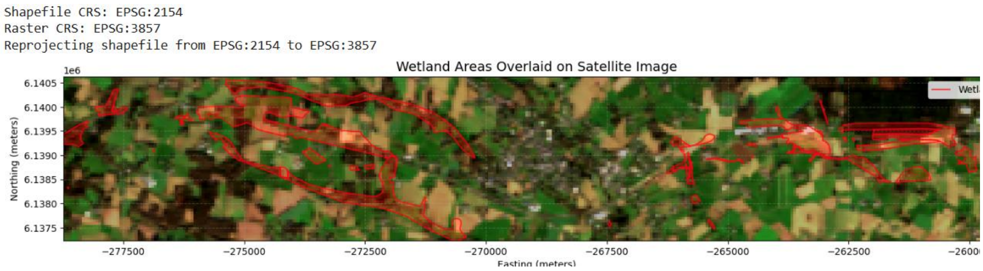

Figure 3: Sentinel-2 image with shapefile overlay

Image Metadata	
Image shape: (62, 357) Total pixels: 22134 Number of bands: 10	Pixel Resolution: Width: 54.85 meters Height: 54.85 meters Area per pixel: 3008.49 square meters
Pixel Counts: Total pixel count: 22,134 Study area pixels: 9,205 Wetlands area pixels: 1,977	Area Measurements: Total area: 66,589,927.01 square meters Study area: 27,693,154.34 square meters Wetlands area: 5,947,785.57 square meters

Table 1: Image Metadata

Class Imbalance Between Wetlands And Non-Wetlands

With the data available having been combined and the wetlands per satellite image area being known, as can be found in the above image metadata table 1, it could be identified how much of the land covered in the image is wetland or non-wetland. As can be seen in the measurement area in the table, the total area makes up around 66.5 million square meters and the wetlands area only making 6 million square meters, we can see that the wetland area covers only 8.93%, with the non-wetland area therefore making up 91.07% of the area depicted in the image.

Clouds Handling Method

Clouds are a major challenge in this case because they can obscure ground features, alter spectral signatures and introduce errors into the models, which reduces accuracy. To address this challenge, a cloud detection method was developed. This method was created separately from the other segmentation models.

This method began with a visual inspection of all tiles in the dataset. It was conducted using a custom-developed interactive tool that displayed visualisations of each image tile, allowing us to manually mark tiles as 'cloudy' or 'cloud-free'. After this process, 35 tiles (19.4% of the dataset) were identified as containing significant cloud cover, while 145 tiles (80.6%) were classified as cloud-free. This initial labelling provided the ground truth for our automated detection methodology.

Rather than using raw pixel values, which can vary significantly across different dates and atmospheric conditions, we focused on the statistical variation within each band as a discriminative feature for cloud detection. For each image tile and each band, we calculated the Coefficient of Variation (CV), defined as the ratio of standard deviation to mean ($CV = \sigma/\mu$), resulting in a ten-dimensional feature vector representing each tile. To determine which bands provided the most discriminative power, we conducted a one-way Analysis of Variance (ANOVA) for each band's CV values across the cloudy and non-cloudy classes. This analysis was complemented by calculating Cohen's d as a measure of effect size to quantify the practical significance of the observed differences between classes.

Based on the identified discriminative features, we developed several machine learning models to automate the cloud detection process: Random Forest Classifier, XGBoost, LightGBM and CatBoost. They were trained to predict the presence of clouds based on the ten-band CV features, learning to replicate our manual classification decisions, with an 80:20 train-test split. Appropriate evaluation metrics were employed to address the class imbalance, including precision, recall, and F1-score, rather than relying only on accuracy.

For the best-performing model, we conducted feature importance analysis using SHAP (SHapley Additive exPlanations) values. The SHAP analysis provided both global insights into feature importance across the entire dataset and local explanations for individual predictions, enhancing the interpretability of the cloud detection system. This model could be used before the segmentation task to remove cloudy patches from the dataset, however as previously mentioned this was a separate exploration and not part of the first implementation covered in the rest of this methodology.

Data Processing And Feature Engineering

For this part it was crucial to consider the modelling approach that wanted to be used. As it was clear that with such little data it would be difficult to get a performant model trained on solely the data provided. For this reason it was decided to not only create our own model, but also run our data on pre-trained existing models in image segmentation fine-tuned models. This consideration meant that the data processing needed to be done a little differently depending on the models used, as pretrained models expect a specific input which needs to be adhered to. The explained methods below are describing the process used for the data processing for these pre-trained models, unless specifically stated that this part of the processing was for solely the small Unet model which was only trained on our data.

Creating Patches

As there was not a lot of data to work with, more image sets were required. For this reason patches were created, so these multiplied images could be used for modelling and allowed for more precise data to work on, as well as ensuring we have a train, validation and test set for each month, where there is no leakage. This process also needed to be implemented for the wetland masks ensuring images and masks could still be used together. To achieve these splits into smaller patches, whilst still preserving the spatial structure, the patchify library was used.

Original and Cropped image and mask

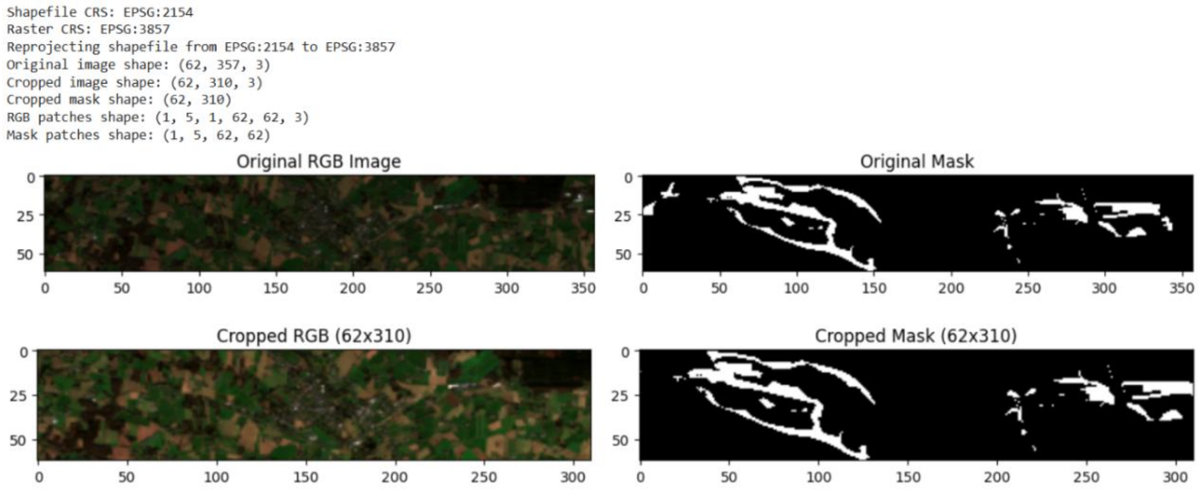

Figure 4: Original and Cropped image and mask

Firstly, to have enough patches for splitting the data into the three sets, it was decided to split each satellite image (input files) and wetland mask (output files) into a total of five patches. They were to be divided into five equal square parts split vertically. However, considering the height of the image is 62 pixels times 357 pixels in width, this does not divide into even squares of 62 by 62. To make it add up a width of 310 pixels was needed, and as per figure 4 above, achieved by cropping out on both sides of the image, allowing for 5 squares of 62x62 pixels in size to be patched on each image and each mask. To ensure consistency and easily being able to work with the data we therefore were left with 36x5 patches of the satellite images as input data, as well as 36x5 patches from the masks as output data, all of which stored in GeoTIFF format. The patched sentinel-2 images, wetland masks, and the patched images with the wetland images are depicted in figure 5 below.

Patches

Figure 5: Patches

Train, Valid, Test Split

Separating the data in this way allowed us to then split the 5 patches per image and mask into three training, one validation and one testing patch per image and masks for all 36 overall images and masks. What needed to be ensured was that with all 36 images and masks, the patches in the validation set were all from the same area and the same goes for the test set ensuring the prevention of data leakage. Therefore for each image and mask the first three patches were separated into the training set, the fourth patches into the validation set and the fifth patches respectively into the test set. Each of the patches was then saved into their perspective separate train, validation and test folders.

Image Bands

From the 10 original bands available, metadata could only be provided on the red, green, blue bands. As the pre-trained models deal with data containing these three bands, these were considered and implemented for this model.

With the Small Unet model however all 10 bands were used including the imported NIR band as well as the the two spectral indices of NDVI and NDWI discussed in the literature review previously, allowing for more information to be captured. The NDVI was calculated from near-infrared and red bands, whilst the NDWI was calculated from green and near-infrared bands.

Image Size

Pretrained models also need the image size of the input data to fit the size required. Padding or resizing the images were both options here, however although resizing works fine for images, there is a scaling issue with masks, leading to worsening of the quality of the mask and in turn worsening image segmentation. Therefore, padding was used to achieve the necessary size for the pretrained model.

Data Augmentation

Having got to a point where the data was prepared for the modelling part it was important to ensure more robustness to the data in the training set allowing for more information being given to the model. For the augmentation the albumentations library was used to ensure all augmentation steps were done in the same way on the image and the mask patches. These augmentation steps included a horizontal flip, a vertical flip and a random 90 degree rotation, and previously mentioned padding and cropping as can be seen in below example. Due to albumentations, for the augmentation the input and output pair were taken and any of those five augmentation steps were automatically implemented on the output data if implemented on the input data at the same time. The random cropping of 128 was chosen due to the required input size of the pretrained models used. For the validation and test set it was ensured that the image size is compatible with the model, however further augmentation was not done on these two sets ensuring realistic implementation.

Augmented patches

Figure 6: Augmented patches

Modelling

All data being prepared, the modelling phase of the work could commence. After creating below own Unet model on solely the data provided, it was also determined that comparing it to existing models previously trained on larger datasets and conducting fine-tuning could be an effective approach, considering such little of our own data being available. For this pre-trained model part it was decided to use the Unet model as well as DeepLabV3+ model allowing for performance comparison, with both models powerful in image segmentation tasks.

Unet From Scratch

In a first step to create the entire model from scratch, the small Unet model was used, due to its compatibility with multi-band Sentinel-2 imagery. We added two consecutive convolutional layers with Batch Normalization and ReLU activation, both layers with a convolution of 3 by 3 and padding applied preserving spatial size. This way the model convergence could be improved through the activation being normalised, as well as non-linearity being introduced.

The model was built in a way to be able to take in the 12 bands described previously and output a binary segmentation of black or white. The encoder layers extracted features and progressively reduced spatial dimensions, and the bottleneck later allowed for high-level feature capturing. Feature maps were upsampled and reintroduced with the decoder path expansion. The output layer was reduced to binary segmentation with sigmoid converting the logits into probabilities. Finally, the model was trained with early stopping with a patience of 15 and epochs of 100

Unet Hyper-Tuning and Transfer-Learning

For addressing the challenge of limited data availability, a pre-trained Unet segmentation model from pytorch was also utilized. The pre-trained model used is 800+ pretrained convolutional and transformer-based backbones. For the backbone of the Unet model the ResNet34 encoder was chosen, whilst ImageNet weights were selected for the encoder weighting. The model was used in two different ways, firstly the model was used with simple fine-tuning and no changes to the layers implemented from our side, and for the second model only the encoder being frozen and the decoder and segmentation head. As previously mentioned for the pre-trained and fine tuned models three input channels were configured, corresponding to the available RGB bands and as with the small Unet model a binary segmentation output was implemented distinguishing between wetland or non-wetland areas for the wetlands mask in black or white. This pretrained model had total parameters of 22’437’457. This model was once implemented with hyper-tuning and once with transfer-learning. For the transfer-learning model the decoder and segmentation head were adapted in the end giving an with 1’152’785 trainable parameters and the rest being frozen, making up only 5.14% of the original parameter amount.

Deeplabv3+ Hyper-Tuning and Transfer-Learning

For the DeepLabV3 Plus model the same approach as the Unet fine-tuned model and U-Net transfer-learning model was applied, with them also using ResNet34 encoder and ImageNet weights. The total parameters of the U-Net model were 24’436’369 with 3’151’697 of these parameters being trainable for the transfer-learning model, making up 12.9% of the parameters.

Evaluation

The DiceLoss was chosen as the loss function for all models, considering its ability to better account for imbalanced datasets. The mean IoU was used as the main evaluation method due to its focus in measuring overlap. It is constructed by using the background IoU and Wetland IoU, which in turn is formed from the IoU. The IoU is derived from the confusion matrix outcome, for this reason all these evaluation metrics need to be calculated.

To best show the outputs of the model and understand its predictions an app was created. This app allows to input a satellite image with the shapefile and gives an output image. The app will also inform of a cloud detection confidence.

VI. Results

In this section we show the final results from the models used. Firstly, the results from the cloud inference models are shown below. These include the test scores of LightGBM, XGBoost, CatBoost and Random Forest models, as well as SHAP analysis for top feature identification, and finally an example image output together with the statistics of that example. This is then followed by the results of the image segmentation models used including the plotted Loss and IoU curves in figure 9, a confusion matrix example in figure 10, model results table in table 4 and an example prediction output of train and test visualisation.

Cloud Cover Results

For the cloud cover the results in table 2 showed a highest F1-Score on the test set for LightGBM, followed by XGBoost and Catboost of 0.73 each and lowest F1-Score score of 0.67 for the random forest model. The SHAP Analysis shows that band1 has the highest impact on cloud detection. The example image shows a confidence interval of 96.15% that clouds are present in the example patch.

Table 2: Test score Cloudy model

Model	tp	tn	fp	fn	precision	recall	f1_score
LightGBM	5	28	2	1	0.71	0.83	0.77
XGBoost	4	29	1	2	0.80	0.67	0.73
CatBoost	4	29	1	2	0.80	0.67	0.73
Random Forest	4	28	2	2	0.67	0.67	0.67

Top features for cloudy classification model

Figure 7: Top features for cloudy classification model

Example output cloudy image

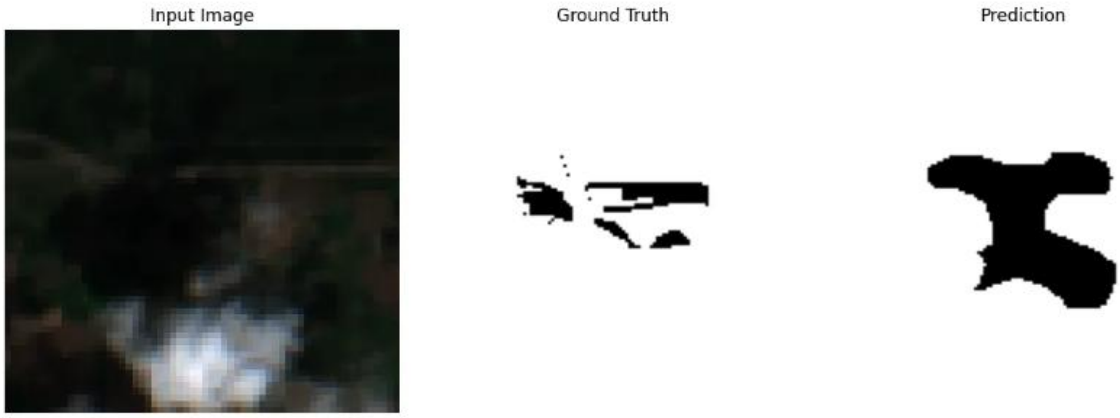

Figure 8: Example output cloudy image

Table 3: Statistics for example output cloudy image

Statistics of Example:	
Wetland Coverage	10.35%
Cloud Detection confidence	96.15%
IoU	0.1259
Precision	0.1440
Recall	0.5010
F1 Score	0.2236

Image segmentation results:

Looking into the image segmentation results it can be seen in the table below, across the training results the Unet fine-tuned model is the highest performing with a mean IoU of 87.85, outperforming the fine-tuned DeepLabV3plus of 70.2 and the two transfer-learning models and small U-Net model all at around 52. Looking into the test mean IoU the DeepLabV3Plus has a highest score with 52.71, followed by the U-Net at 50.8 of which the lower value can be attributed to poor wetland IoU scoring. The Small U-Net performed worst across the board with at 46.85 mean test IoU. An example of the train and test visualisation was also added in figure 11 and figure 12 respectively.

Loss and IoU Curves

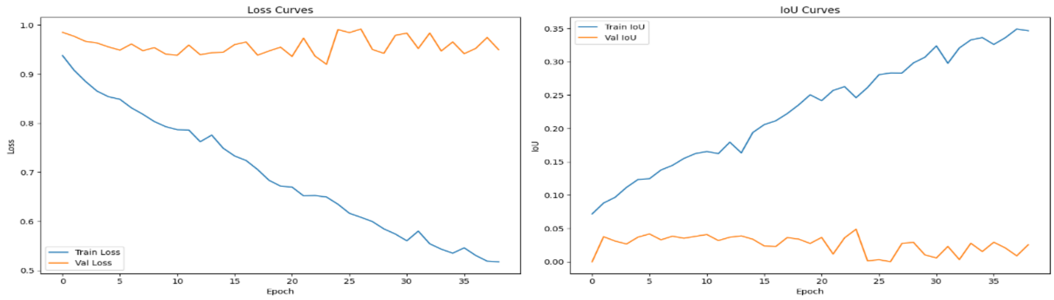

Figure 9: Loss and IoU Curves

Confusion Matrix example

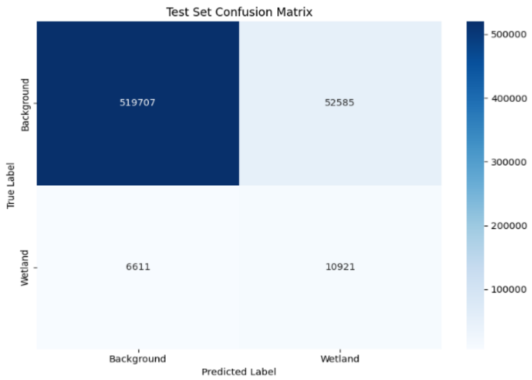

Figure 10: Confusion Matrix example

Table 4: Results Segmentaion Models

Model	total_epochs	train_bg_iou	train_wetland_iou	train_mean_iou	val_bg_iou	val_wetland_iou	val_mean_iou	test_bg_iou	test_wetland_iou	test_mean_iou
DeepLabV3plus_FT	68	97.8	42.6	70.2	97.49	8.29	52.89	95.4	10.02	52.71
Unet_FT	88	99.31	76.4	87.85	97.9	6.71	52.31	96.25	5.34	50.8
DeepLabV3plus_TL	33	89.03	16.59	52.81	87.15	3.23	45.19	86.7	12.04	49.37
Unet_TL	28	87.79	16.3	52.05	85.13	3.03	44.08	84.73	11.53	48.13
SmallUNet	30	86.5	19.02	52.76	93.97	1.71	47.84	81.53	12.17	46.85

Example of train visualisation

Figure 11: Example of train visualization

Example of test visualisation

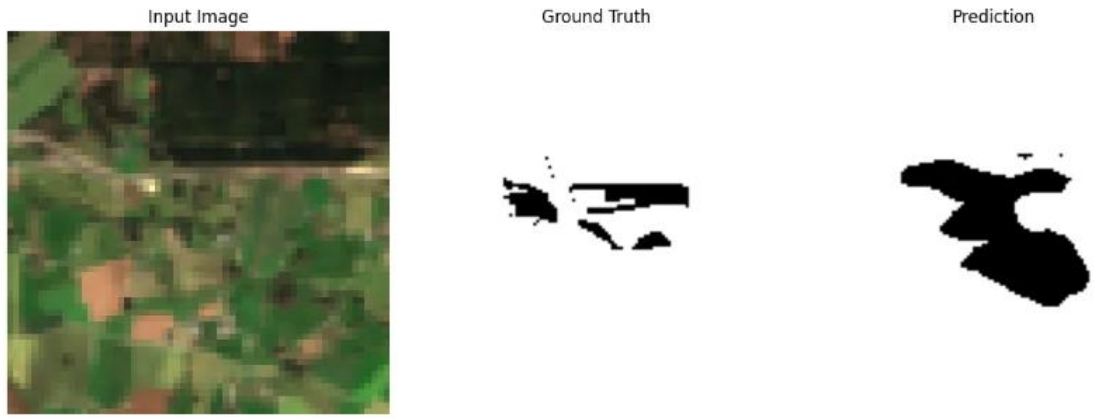

Figure 12: Example of test visualisation

VIII. Discussion

When looking into the performance results of the five models used, it reveals the challenges of image segmentation of wetlands in satellite imagery. Although the fine-tuned U-Net model achieves a strong performance on the training set, achieving a strong mean IoU score, it does not translate to the test set, highlighting strong overfitting. Although slightly less well performing in the training set the DeepLabV3Plus fine-tuned model achieves better generalization with a highest test mean IoU (52.71) out of the five models, of which notably the wetland IoU is significantly higher at 10.04 than the U-Net fine-tuned model at 5.34. The two transfer-learning model both performed worse than their fine-tuned counterparts. These results are fitting when considering the model architectures with DeepLabV3Plus being optimised for better boundary refinement through its ASPP and decoder set-up. Due to the lack of data its is also not a great surprise that the transfer-learning models fail to outperform the simple fine-tuned models. The small U-Net from scratch is performing badly across all phases, highlighting the difficulty of meaningful feature extraction with limited low quality data without any pretrained weights in contrast to the hyper-tuned models.

A major player affecting all models is the strong class imbalance between the background and wetlands in this segmentation tasks, with wetlands making up only 8.93% of the overall image area. This is reflected in the constant high background IoU and low wetland IoU, especially in the test results. This is even more evident with the DiceLoss curve, being more fitting for imbalance datasets and showing strong overfitting. As the DiceLoss shows the generalisation on the test set was not good, showing that data limitations could be the major factor in poor performance rather than just the model architecture.

Rather than simply attributing the weak performance of the models to the models themselves, we can see on the training sets that the pre-trained models are capable of high performance and therefore likely to be able to do well on good data. For this reason it is needed to consider the entire process of this project and identify the crucial points that can be approached to improve the outcome of this task.

Data preparation

First and foremost it is clear that high performance on this task is not possible when the data used is not sufficient. To start with the next steps of the project it is crucial that further data can be prepared, as this task is likely to require thousands of patches to train on before it will be capable of more accurate predictions. Furthermore the quality of the data also needs to be improved, with Sentinel-2 images having a high amount of pixels when the highest level of quality is acquired. This addition would allow the models to identify more detailed information in the patches most likely also leading to significant improvements in the prediction outcomes. The usage of shapefiles to train the model however seems to prove very effective and should be maintained for future model training. It also clearly needs to be remembered that the nature of the provided data having a strong imbalance with the wetland pixel proportion making up only 8.93% of the overall image pixels, and although there are some techniques to improve this, it needs to be noted that this imbalance will make it quite difficult to achieve a mean IoU of over 0.7 on the test set.

Furthermore, the type of data can also be considered. The fact that Sentinel-2 time series data was used for this study seems to be very fitting and a good solution for solving this segmentation task when compared to previous literature on this matter. This especially holds true considering the nature of wetlands and changes they go through in size and colour throughout the year. When further looking into previous studies, in many cases Sentinel-1 images were also used with success. They allowed for neural networks to perform well in image segmentation tasks. However, considering that most often sentinel-2 images were also used with success, it seems that adding sentinel-1 images is not the most urgent implementation for improvement, especially considering the difficulties it can provide in preprocessing of the data. Nonetheless, adding Sentinel-1 imagery can be considered to further improve the performance, allowing the models to pick up different types of information and can be a great solution for the cloud issues encountered, considering its ability to penetrate through cloudy images.

Finally, an important part of working with Sentinel-2 images is taking advantage of the spectral bands available, rather than only using four of the channels. Adding to that the spectral indices of Normalized Difference Vegetation Index (NDVI) and the Normalized Difference Water Index (NDWI) can also add plenty of further information to the model that is not visible to the human eye. This step again can be even more valuable when the data quality is improved. As the better performing models were pre-trained and needed a specific channel input size, this unfortunately could not be taken advantage of there. 12 channels were however used for the small U-Net model, and could have significant impact when the previously mentioned data improvement steps have been implemented.

Processing and Feature engineering

Furthermore, augmenting the training data with techniques like flipping, rotation, and padding improved model robustness but may not have been sufficient. Additional augmentation methods, or synthetic image generation, could be tested. However this is unlikely to have a high impact on performance even if it would slightly increase the robustness of the model

A further problem identified was clouds in images. Apart from simply moving to Sentinel1 images where valuable Sentinel-2 information would get lost, the already prepared separate models on cloud identification in images are capable of identifying clouds in the patches. In future the created cloud identification model can be run prior to running the image segmentation model steps. This step allows for the removal of cloudy images from the dataset. In this case when running the test set it would highlight probability of clouds being in an image allowing to extract these images and not use them. The previous point of requiring a larger dataset will be crucial for this part as well, as this step would result in a reduction of the total patches.

Modelling

The best-performing models of the project were pre-trained neural networks, supporting existing literature and confirming the importance of utilising pre-trained models on small datasets with satellite images. As unfortunately no fine-tuned model on sentinel-2 imagery could be found, the fine-tuned models were not actually run on similar data originally, making it harder to achieve high performance. If a more fitting fine-tuned model already trained on satellite imagery were to be discovered, the performance could already be stronger without even considering more and better quality of data.

Considering the pre-trained models specifically, it is important to note that the best results were found when fine-tuning the pre-trained models. There can be a lot of potential by applying transferred-learning on these pre-tuned models, however in the case of this project this did not prove to be the case despite training on final layers. Transfer-learning can achieve a stronger impact if previously discussed data-steps were to be conducted.

Furthermore, the required input of 3 channels into the pre-tuned models meant that the value added from other spectral bands and indices could not be taken advantage of. In some previous literature bands proved to have an impact on model performance, however it showed mixed results in some past such segmentation tasks and may not lead to much improvement. Therefore, if a pretrained model could be found with a different channel input, likely to be the case on a pretrained model on satellite imagery, or a way could be found to add further channels to the existing pretrained models, an outcome improvement could be possible. This way, at least an understanding could be gained on how much impact these bands can make in this specific use case. Finally, when looking into the small U-Net model it is likely that a lot more data will be needed to make up for the lack of information gained in comparison to the pre-trained models. Nonetheless, due to its set up with bands and augmentation the performance could significantly improve with a large increase in data samples and data quality. Furthermore, as research showed that the Swin U-Net model sometimes performed better on satellite segmentation tasks, exploring this option could be an alternative. Furthermore, on simple satellite image classification tasks machine learning models have performed well in the past, so this could be explored for comparison as well and may perform better with fewer data available.

The evaluation approach used, seems to be most fitting especially considering the class imbalance of the data. It highlighted this issue of imbalance and showed the difficulty of predicting wetland areas. What is really valuable however is seeing the predicted output visually, as what is not captured by these evaluation methods is how close the predicted pixels were to the actual wetland areas. As the closer it is, the more realistic the prediction becomes, however it will always be classed as a false positive.

Overall

Despite limited data and low data quality, this study developed a pipeline for cloud detection, shapefile-based mask overlays, effective patching strategy and model training. Despite the final test performance leaving much to be desired, it shows the difficulty of wetland segmentation when real world constraints such as imbalance, cloud inference and limited low quality data exist. Even considering these challenges the project confirmed the promising foundation of pre-trained models for future deployments. Improvements in data volume, band implementation, and specificity in transfer-learning are likely to strongly impact performance in future iterations of this work. All this allows for a solid foundation and proof of concept allowing for further work to be built upon it. The app created is a good way to quickly see the prediction on satellite images which can be very valuable when the models achieve higher performance. The app also identifies probability of cloud inference which helps users understand not to consider images where this inference is high.

IX. Conclusion

This project set out with a clear and practical goal: to explore whether data science, and specifically deep learning, could be leveraged to detect and monitor wetlands using freely available Sentinel-2 satellite imagery. Through the DataCraft Wetlands Project, we developed a complete pipeline from data preprocessing, exploration of cloud interference, patch extraction, model training, and final evaluation. Despite facing several real-world challenges, our work represents a strong proof of concept and allows for implementation in the future.

Our results revealed both the promise and the complexity of using AI for wetland detection. On the positive side, we successfully demonstrated that pre-trained models, particularly with models like U-Net and DeepLabV3+, provide a viable path for wetland segmentation even with limited data. The use of time series imagery added valuable depth to our dataset, allowing the model to better understand seasonal variation in wetland appearance. However, the final test performance was limited by significant issues, particularly the low quality and quantity of annotated data, which reduced the model's ability to generalize.

Several technical barriers emerged throughout the study. Image sizes were often incompatible with pre-trained models, requiring resizing and transformation strategies that may have compromised spatial detail. The datasets were imbalanced, with wetlands representing only a small portion of the total area and making accurate prediction even harder. Additionally, cloud cover added noise and confusion for the models. Another limitation was the number of spectral bands used in training. While Sentinel-2 provides rich multi-band data, many pre-trained segmentation models are built for 3-channel RGB images, limiting our ability to fully leverage the sensor’s capabilities. Finally, a lack of open-source, wetland-specific pre-trained models presented another obstacle, forcing us to adapt more generic solutions to a very domain-specific task.

Looking ahead, several promising directions are clear. The most impactful step would be to gather a larger, more diverse dataset ideally with over a thousand high-resolution, well-labeled images accompanied by rich metadata. Ensuring that image sizes are compatible with model architectures (divisible by 8, 32, 64, or 256) will reduce unnecessary preprocessing overhead. Addressing class imbalance with advanced augmentation strategies and exploring oversampling methods could improve training effectiveness. The handling of clouds in images can be a crucial step at helping the model perform better. This can be achieved by using Sentinel 1 images that can go through clouds in images, implementing the cloud detection model explored in this project to remove cloudy patches before running the segmentation model. On the modelling side, there's an opportunity to explore more advanced deep learning techniques or band-ratio-based indices to better handle environmental noise like clouds. Incorporating all 12 Sentinel-2 bands and adding NDVI and NDWI indices can enable far richer model understanding. Ultimately, contributing back to the community by developing and open-sourcing a pre-trained wetland segmentation model could have broad utility.

In summary, while the current implementation faced tangible limitations, it also delivered meaningful progress. It confirms that deep learning, especially when combined with open-access satellite data, has significant potential to support environmental monitoring. With targeted improvements in data quality, preprocessing, and model architecture, this method could be a valuable tool for researchers.

XI. Bibliography

Alosius, T. (2023, August 17). Implementing YOLOv8 for Aerial Satellite Image Building Segmentation and Converting to Shape files. *Medium*. https://medium.com/@tony.aloysius.77/implementing-yolov8-for-aerial-satellite-image-building-segmentation-and-converting-to-shape-files-899b2a8affa5

Amani, M., Salehi, B., Mahdavi, S., & Granger, J. (2017). *SPECTRAL ANALYSIS OF WETLANDS IN NEWFOUNDLAND USING SENTINEL 2A AND LANDSAT 8 IMAGERY*.

augmenters.size—Imgaug 0.4.0 documentation. (n.d.). Retrieved March 25, 2025, from https://imgaug.readthedocs.io/en/latest/source/overview/size.html

Baetens, L., Desjardins, C., & Hagolle, O. (2019). Validation of Copernicus Sentinel-2 Cloud Masks Obtained from MAJA, Sen2Cor, and FMask Processors Using Reference Cloud Masks Generated with a Supervised Active Learning Procedure. *Remote Sensing*, *11*. https://doi.org/10.3390/rs11040433

Barrie, S. (2022, December 14). *Into the Unknown—Introduction to Geospatial Raster and Vector data with Python*. https://stephen137.github.io/posts/Geospatial/Working%20with%20Geospatial%20Data%20in%20Python.html#raster-data

Buslaev, A., Parinov, A., Khvedchenya, E., Iglovikov, V. I., & Kalinin, A. A. (2020). Albumentations: Fast and flexible image augmentations. *Information*, *11*(2), 125. https://doi.org/10.3390/info11020125

Cabezas, M., & Diez, Y. (2024). An Analysis of Loss Functions for Heavily Imbalanced Lesion Segmentation. *Sensors*, *24*(6), Article 6. https://doi.org/10.3390/s24061981

Campos-Taberner, M., Haro, F., Martinez, B., Izquierdo-Verdiguier, E., Atzberger, C., Camps-Valls, G., & Gilabert, M. A. (2020). Understanding deep learning in land use classification based on Sentinel-2 time series. *Scientific Reports*, *10*. https://doi.org/10.1038/s41598-020-74215-5

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). *Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation* (No. arXiv:1802.02611). arXiv. https://doi.org/10.48550/arXiv.1802.02611

Clewley, D., Bunting, P., Shepherd, J., Gillingham, S., Flood, N., Dymond, J., Lucas, R., Armston, J., & Moghaddam, M. (2014). A Python-Based Open Source System for Geographic Object-Based Image Analysis (GEOBIA) Utilizing Raster Attribute Tables. *Remote Sensing*, *6*, 6111–6135. https://doi.org/10.3390/rs6076111

Dang, K. B., Nguyen, M. H., Nguyen, D. A., Phan, T. T. H., Giang, T. L., Pham, H. H., Nguyen, T. N., Tran, T. T. V., & Bui, D. T. (2020). Coastal Wetland Classification with Deep U-Net Convolutional Networks and Sentinel-2 Imagery: A Case Study at the Tien Yen Estuary of Vietnam. *Remote Sensing*, *12*(19), Article 19. https://doi.org/10.3390/rs12193270

Demir, I., Koperski, K., Lindenbaum, D., Pang, G., Huang, J., Basu, S., Hughes, F., Tuia, D., & Raskar, R. (2018). DeepGlobe 2018: A Challenge to Parse the Earth through Satellite Images. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 172–17209. https://doi.org/10.1109/CVPRW.2018.00031

Fuentes-Peñailillo, F., Ortega-Farias, S., Rivera, M., Bardeen, M., & Moreno, M. (2018). *Comparison of vegetation indices acquired from RGB and Multispectral sensors placed on UAV*. 1–6. https://doi.org/10.1109/ICA-ACCA.2018.8609861

Gangal, A., Kumar, P., & Kumari, S. (2021). *Complete Scanning Application Using OpenCv*. https://doi.org/10.48550/arXiv.2107.03700

Gao, B.-C. (1996). NDWI?A Normalized Difference Water Index for Remote Sensing of Vegetation Liquid Water from Space. *Remote Sensing of Environment*, *58*, 257–266.

https://doi.org/10.1016/S0034-4257(96)00067-3

Gasteiger, J., Emde, C., Mayer, B., Buras, R., Buehler, S. A., & Lemke, O. (2014). Representative wavelengths absorption parameterization applied to satellite channels and spectral bands. *Journal of Quantitative Spectroscopy and Radiative Transfer*, *148*, 99–115. https://doi.org/10.1016/j.jqsrt.2014.06.024

GISGeography. (2019, April 22). Sentinel 2 Bands and Combinations. *GIS Geography*. https://gisgeography.com/sentinel-2-bands-combinations/

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., & Moore, R. (2017). Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, *202*, 18–27. https://doi.org/10.1016/j.rse.2017.06.031

Hosseiny, B., Mahdianpari, M., Brisco, B., Mohammadimanesh, F., & Salehi, B. (2021). WetNet: A Spatial-Temporal Ensemble Deep Learning Model for Wetland Classification Using Sentinel-1 and Sentinel-2. *IEEE Transactions on Geoscience and Remote Sensing*, *PP*, 1–14. https://doi.org/10.1109/TGRS.2021.3113856

Ji, L., Zhang, L., & Wylie, B. (2009). Analysis of Dynamic Thresholds for the Normalized Difference Water Index. *Photogrammetric Engineering & Remote Sensing*, *75*, 1307–1317. https://doi.org/10.14358/PERS.75.11.1307

Jones, E. G., Wong, S., Milton, A., Sclauzero, J., Whittenbury, H., & McDonnell, M. D. (2020). The Impact of Pan-Sharpening and Spectral Resolution on Vineyard Segmentation through Machine Learning. *Remote Sensing*, *12*(6), Article 6. https://doi.org/10.3390/rs12060934

Kaplan, G., & Avdan, U. (2017). MAPPING AND MONITORING WETLANDS USING SENTINEL-2 SATELLITE IMAGERY. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *IV-4-W4*, 271–277. WG IV/1
4th International GeoAdvances Workshop – GeoAdvances 2017: ISPRS Workshop on Multi-dimensional & Multi-scale Spatial Data Modeling (Volume IV-4/W4) -

14–15 October 2017, Safranbolu, Karabuk, Turkey. https://doi.org/10.5194/isprs-annals-IV-4-W4-271-2017

Le Guillou, A., Niculescu, S., & Schmullius, C. (2023). Machine and deep learning methods for detection and mapping of coastal wetlands of Crozon Peninsula (Brittany, France) used metric and sub-metric spatial resolution. *Proceedings of SPIE, the International Society for Optical Engineering*, *12734*. https://doi.org/10.1117/12.2678006

Li, Z., Shen, H., Cheng, Q., Liu, Y., You, S., & He, Z. (2019). Deep learning based cloud detection for medium and high resolution remote sensing images of different sensors. *ISPRS Journal of Photogrammetry and Remote Sensing*, *150*, 197–212. https://doi.org/10.1016/j.isprsjprs.2019.02.017

Lima, R. P. de, Vahedi, B., & Karimzadeh, M. (2023). Comparison of Cross-Entropy, Dice, and Focal Loss for Sea Ice Type Segmentation. *IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium*, 145–148. https://doi.org/10.1109/IGARSS52108.2023.10282060

Liu, M., Fu, B., Xie, S., He, H., Lan, F., Li, Y., Lou, P., & Fan, D. (2021). Comparison of multi-source satellite images for classifying marsh vegetation using DeepLabV3 Plus deep learning algorithm. *Ecological Indicators*, *125*, 107562. https://doi.org/10.1016/j.ecolind.2021.107562

Long, J., Shelhamer, E., & Darrell, T. (2015). *Fully Convolutional Networks for Semantic Segmentation* (No. arXiv:1411.4038). arXiv. https://doi.org/10.48550/arXiv.1411.4038

Mahdianpari, M., Granger, J. E., Mohammadimanesh, F., Salehi, B., Brisco, B., Homayouni, S., Gill, E., Huberty, B., & Lang, M. (2020). Meta-Analysis of Wetland Classification Using Remote Sensing: A Systematic Review of a 40-Year Trend in North America. *Remote Sensing*, *12*(11), Article 11. https://doi.org/10.3390/rs12111882

(PDF) Sentinel-1 and Sentinel-2 Data Fusion for Mapping and Monitoring Wetlands. (2024, December 9). ResearchGate. https://doi.org/10.20944/preprints201807.0244.v1

Qubvel-org/segmentation_models.pytorch. (2025). [Python]. qubvel-org. https://github.com/qubvel-org/segmentation_models.pytorch (Original work published 2019)

Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation* (No. arXiv:1505.04597). arXiv. https://doi.org/10.48550/arXiv.1505.04597

Rumora, L., Miler, M., & Medak, D. (2020). Impact of Various Atmospheric Corrections on Sentinel-2 Land Cover Classification Accuracy Using Machine Learning Classifiers. *ISPRS International Journal of Geo-Information*, *9*(4), Article 4. https://doi.org/10.3390/ijgi9040277

Segmentation Models—Segmentation Models documentation. (n.d.). Retrieved March 23, 2025, from https://smp.readthedocs.io/en/latest/models.html#unet

Segmenting Satellite Imagery with the Segment Anything Model (SAM) || *Image Segmentation using AI*. (2024, July 26). https://www.topview.ai/blog/detail/segmenting-satellite-imagery-with-the-segment-anything-model-sam-image-segmentation-using-ai

Shukla, G., Tiwari, P., Dugesar, V., & Srivastava, P. K. (2021). Chapter 9—Estimation of evapotranspiration using surface energy balance system and satellite datasets. In P. K. Srivastava, M. Gupta, G. Tsakiris, & N. W. Quinn (Eds.), *Agricultural Water Management* (pp. 157–183). Academic Press. https://doi.org/10.1016/B978-0-12-812362-1.00009-6

Temporal Convolutional Neural Network for the Classification of Satellite Image Time Series. (n.d.). Retrieved March 23, 2025, from https://www.mdpi.com/2072-4292/11/5/523

What is transfer learning? | *IBM*. (2024, February 12). https://www.ibm.com/think/topics/transfer-learning

Wurm, M., Stark, T., Zhu, X. X., Weigand, M., & Taubenböck, H. (2019). Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, *150*, 59–69. https://doi.org/10.1016/j.isprsjprs.2019.02.006

Xie, Q., Dash, J., Huang, W., Peng, D., Qin, Q., Mortimer, H., Casa, R., Pignatti, S., Laneve, G., Pascucci, S., Dong, Y., & Ye, H. (2018). Vegetation Indices Combining the Red and Red-Edge Spectral Information for Leaf Area Index Retrieval. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *11*. https://doi.org/10.1109/JSTARS.2018.2813281

Yao, J., & Jin, S. (2022). Multi-Category Segmentation of Sentinel-2 Images Based on the Swin UNet Method. *Remote Sensing*, *14*(14), Article 14. https://doi.org/10.3390/rs14143382

XII. List of Figures

XIII. List of Tables