Semantic Segmentation of Multispectral Remote Sensing Images Using Transformers

Pablo Canosa García
Master in Artificial Intelligence
Universidade de Santiago de Compostela
Santiago de Compostela, Spain
pablo.canosa.garcia@rai.usc.es

***Abstract*—Semantic segmentation for Earth observation is a process that involves classifying each pixel in an image into predefined categories. For remote sensing images used in land cover and land use analysis, segmentation facilitates detailed analysis and understanding by identifying elements such as water bodies, vegetation, urban areas, and roads. Deep learning, particularly transformers, has been transformative in the field of semantic segmentation of remote sensing images, yielding significative better results than previous techniques due to their high generalization capability and adaptability to different properties of the images. In this work, a semantic segmentation model, Mask2Former, which has been developed for semantic segmentation of RGB images, is adapted to be used over multispectral images. Mask2Former is based on transformers and performs semantic segmentation by assigning masks to the classes of elements presents in the image. The 4-band multispectral images considered are those contained in the Five Billion Pixels dataset and captured by the Gaofen-2 satellite. Models trained with the proposed Mask2Former for multispectral images were compared with models trained with the original Mask2Former using only the RGB bands, as well as with state-of-the-art methods in the literature. The results demonstrated the benefits of using multispectral images in terms of standard metrics such as overall accuracy and IoU.**

***Index Terms*—Semantic segmentation, remote sensing, multispectral imaging, transformer, land cover**

I. Introduction

Multispectral images provide detailed information far beyond the visible spectrum. This information allows for enhanced analysis and interpretation of materials and environments, facilitating advancements in numerous fields. In remote sensing for Earth Observation the multispectral information is typically captured by sensors onboard satellites or aerial vehicles.

Unlike RGB images, which capture data in only three bands (red, green, and blue), multispectral images significantly increase the number of bands, usually ranging from 4 to 15. These additional bands include non-visible wavelengths such as near-infrared (NIR), short-wave infrared (SWIR), thermal infrared (TIR), and mid-wave infrared (MWIR), providing more information on the reflectance properties of the elements and materials and enabling more comprehensive analysis and interpretation of the captured scenes. Applications of these kind of images include monitoring biodiversity in large forest areas [1] or assessing the effects of climate change [2], among others. The essential aspect of these applications is understanding the land use/cover of the areas being studied [3].

Image segmentation has been used in remote sensing image processing since the launch of the Landsat-1 satellite for object identification, understanding and interpreting visual information [4]. It is widely used to identify and detect land covers such as buildings, trees, water bodies, and grasslands. Segmentation consists in partitioning an image into distinct regions or segments, each segment representing different objects or areas of interest.

Three types of segmentation can be performed, as seen in Fig. 1:

1) **Instance segmentation** focuses on identifying and delineating each object instance in an image. This task not only classifies objects but also provides a precise boundary for each instance, ensuring that multiple objects of the same class are distinguishable. Instance segmentation methods typically use region-based approaches and object proposals to achieve high accuracy in detecting and outlining objects.
2) **Semantic segmentation** assigns a class label to every pixel in an image, categorizing all parts of the scene into predefined classes such as “sky”, “road”, or “car”. Unlike instance segmentation, semantic segmentation does not differentiate between individual instances of the same class. This task is essential for understanding the overall context and structure of an image, making it crucial for applications like scene parsing and environmental understanding.
3) **Panoptic segmentation** combines the strengths of both instance and semantic segmentation into a single unified task. Each pixel in the image is assigned both a semantic label and an instance ID, allowing for the identification and delineation of individual object instances while also categorizing amorphous regions.

Recently, the advent of deep learning techniques has significantly advanced research in semantic segmentation. To date, numerous innovative deep learning-based methods have been proposed, each following different technical approaches and targeting various applications. Compared to traditional methods, these deep learning-based techniques have demonstrated significant improvements in effectiveness [6].

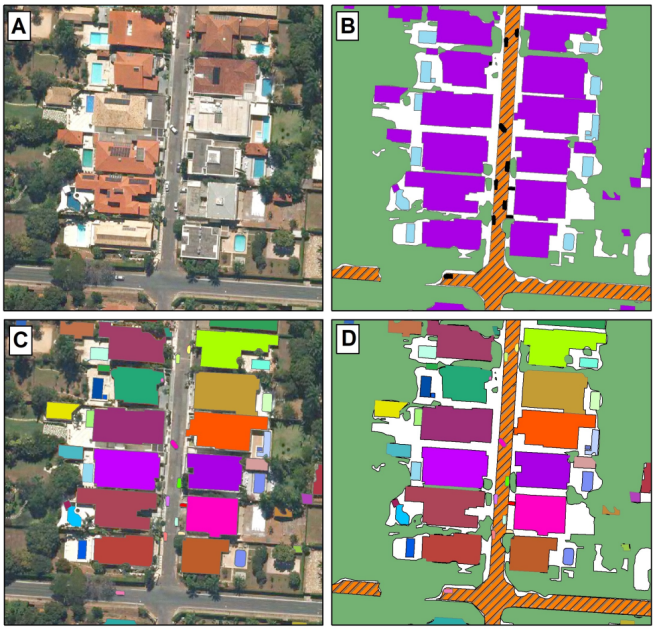

Fig. 1. Example of the different image segmentation tasks. (A) Original image, (B) Semantic segmentation, (C) Instance segmentation, and (D) Panoptic segmentation [5].

Deep learning image segmentation methods can be classified according to the network architecture implemented: [7]: Fully Convolutional Networks (FCNs), Convolutional Neural Networks (CNNs) with graphical models, Encoder-Decoder-based models, Regional CNN (R-CNN) models, Dilated CNN models, Recurrent Neural Networks (RNNs), and Transformers.

FCNs are designed for image segmentation and include only convolutional layers. It allows for efficient pixel-wise predictions and handling variable input sizes through upsampling techniques. Maggiori et al. [8] demonstrate its application for remote sensing image segmentation. CNNs with graphical models aim to improve segmentation accuracy by incorporating scene-level context, achieved with conditional random fields and Markov random fields as help, used for classification of hyperspectral images by Gao et al. [9].

Encoder-Decoder based models are composed of an encoder, in charge of mapping an image to a high dimensional space, and a decoder in charge of generating the segmentation mask out of the high dimensional representation. This type of method is widely used in image segmentation. Models such as EfficientUNet++ [10], SegNet [11], U-Net [12], and M-Net [13] stand out. Multispectral U-Net [14] shows an adaptation of U-Net for landslide detection in remote sensing multispectral images.

Dilated CNN models use dilated (atrous) convolutions to increase the receptive field without extra parameters, the new type of convolution enlarges the receptive field while keeping the same computational cost. Dilated-ResUnet [15] uses these convolutions for multispectral remote sensing image classification. In the case of R-CNN models, they are based on selecting regions of interest in the image, and extracting features from each region independently for classification, implemented in Mask R-CNN [16], PANet [17], MaskLab [18], and modified R-CNNs for remote sensing [19].

Finally, Transformers implement attention mechanisms to improve segmentation by focusing on relevant parts of the image. Some examples of transformer models for image segmentation are DETR [20], CapViT [21], which focuses on land cover classification, or Mask2Former [22].

This work presents a first approach to implementing the Mask2Former transformer architecture for semantic segmentation of remote sensing multispectral images. Mask2Former is designed to be applied to RGB images, so it was adapted to exploit the whole spectral data of multidimensional images. The evaluation was carried out over a multispectral remote sensing image dataset containing several classes such as cities, mountains, villages, bodies of water, or industrial parks. The main contributions of this work are:

1) **Architectural Adaptation**. Modifying the architecture of Mask2Former to handle the unique characteristics and additional bands present in multispectral data, ensuring the model can leverage the full range of information.
2) **Multispectral vs. RGB Analysis**. Analyzing how the semantic segmentation of multispectral images perform in comparison to consider only the three RGB bands, identifying the specific advantages and potential limitations of using multispectral data for remote sensing applications.
3) **Performance Comparison**. Conducting a comprehensive comparison of the performance of the modified Mask2Former model with existing state-of-the-art methods in the field of remote sensing image segmentation.

This master's thesis is organized as follows. Section II explains the fundamentals of MaskFormer and Mask2Former. How Mask2Former was adapted to handle multispectral data is described in Section III; the experiments and results are discussed in Section IV. Finally, conclusions and future work are discussed in Sections V and VI.

II. Mask2Former Architecture

In this section, we present the basics of MaskFormer and how this segmentation model evolved into Mask2Former, its second version.

MaskFormer [23] is a model, part of the Detectron2 framework [24], designed for semantic segmentation tasks, transitioning from per-pixel classification to mask classification [23]. Mask classification is a different approach to segmentation that separates the tasks of dividing the image and classifying it. Instead of assigning a class to each pixel, methods based on mask classification predict several binary masks, with each mask linked to a specific class.

Fig. 2 outlines the MaskFormer architecture. It comprises three key modules: a pixel-level module, a transformer module, and a segmentation module.

- **Pixel-level module.** This module takes an input image and utilizes a backbone network to extract low-resolution

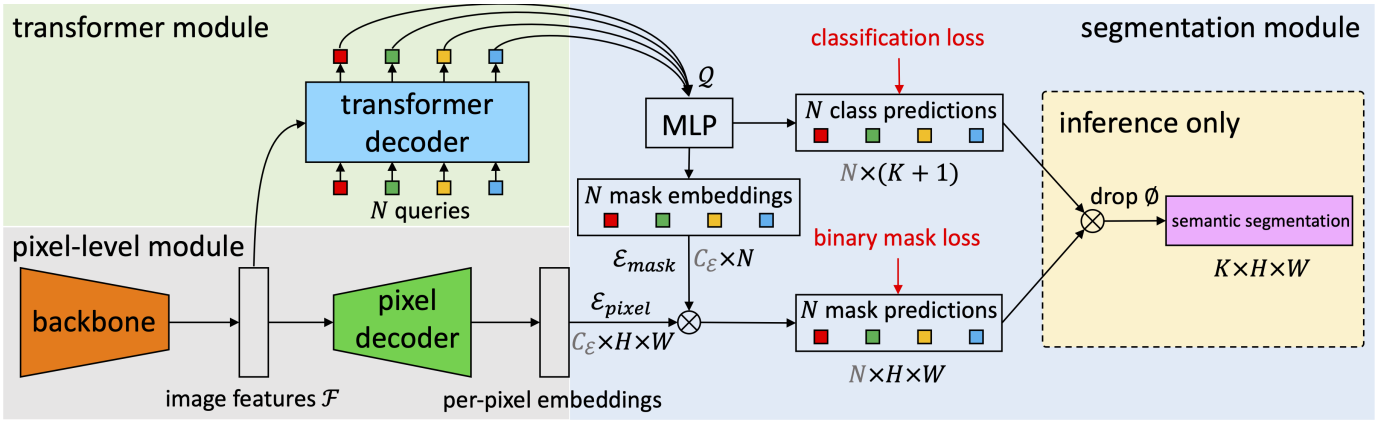

Fig. 2. Schematic of the MaskFormer modules [23].

image features. These features are gradually upsampled to generate per-pixel embeddings, capturing detailed information about each pixel in the image. This module is designed to facilitate the generation of binary mask predictions.

- **Transformer module.** The transformer module employs a stack of transformer decoder layers to compute per-segment embeddings. These embeddings encode global information about each segment and are essential for capturing contextual relationships within the image. The transformer decoder produces all predictions in parallel, enhancing efficiency.
- **Segmentation module.** The segmentation module consists of a linear classifier followed by a softmax activation, generating class probability predictions for each segment. Additionally, a multi-layer perceptron (MLP) is employed to convert per-segment embeddings into mask embeddings. These mask embeddings are then used to generate binary mask predictions for each segment.

During the inference, the output of MaskFormer consists of probability-mask pairs, representing the predicted segmentation for the input image. General inference strategies can be applied to convert these outputs into panoptic or semantic segmentation formats, depending on the evaluation metric. Specifically, pixels are assigned to probability-mask pairs based on the highest class probability and mask prediction probability. For semantic segmentation, segments sharing the same category label are merged, while for instance-level segmentation tasks, the index of the probability-mask pair distinguishes different instances of the same class.

Mask2Former [22] adopts a similar architecture to MaskFormer, with the following modules: a backbone, a pixel decoder, and a transformer decoder, as illustrated in Fig. 3. It introduces an enhanced feature processing stage using multi-scale features in the pixel decoder, modified loss functions that include strategies to address class imbalance and convergence issues, and a new transformer decoder with masked attention *masked attention*. These improvements are explained in more detail below.

The first module of Mask2Former is the backbone, which is a pre-trained model, such as ResNet [25], trained on large datasets. It extracts low-resolution features from the input image and works with multiple layers, where early layers capture fine details while deeper ones capture more general features. The output of the backbone is input to the pixel decoder, as shown in Fig. 3.

The next module is the pixel decoder, which receives the feature maps produced by the backbone and generates high-resolution pixel embeddings. It creates a multiscale feature pyramid with resolutions at 1/32, 1/16, and 1/8 of the original image. Instead of using only the high-resolution feature map, each resolution of the multi-scale feature is fed into a different layer of the Transformer decoder. Finally, the high-resolution feature map is directly compared to the output of the transformer decoder to make the final prediction, as seen in Fig. 3.

The final module of Mask2Former is the transformer decoder, which receives the feature pyramid generated by the pixel decoder. In Mask2Former, the order of self-attention and cross-attention modules is switched, introducing a "masked attention" approach that prioritizes computational effectiveness. Additionally, dropout, which is typically applied to residual connections and attention maps, was found to be unnecessary and has been removed to avoid performance degradation. Inspired by techniques like PointRend [26] and Implicit PointRend [27], the mask loss calculation is modified to operate on a reduced set of randomly sampled points rather than the entire mask. As seen in Fig. 3, the transformer decoder is composed of four different parts [22]:

- **Masked Attention Layer:** This layer performs the masked attention operation where each query attends only to the localized foreground regions of the predicted mask from the previous layer, instead of the entire feature map. This localized attention helps the model to focus on relevant parts of the image.
- **Self Attention Layer:** This layer allows each query to interact with every other query, facilitating the exchange of information among them. It helps in refining the

query features by incorporating global context from other queries.

- **Normalization Layer:** This layer is responsible for adding the residual connections to the output of the attention layers, and then normalizing the result.
- **Feed Forward Network:** This network processes each query feature independently with a series of linear transformations and non-linear activations.

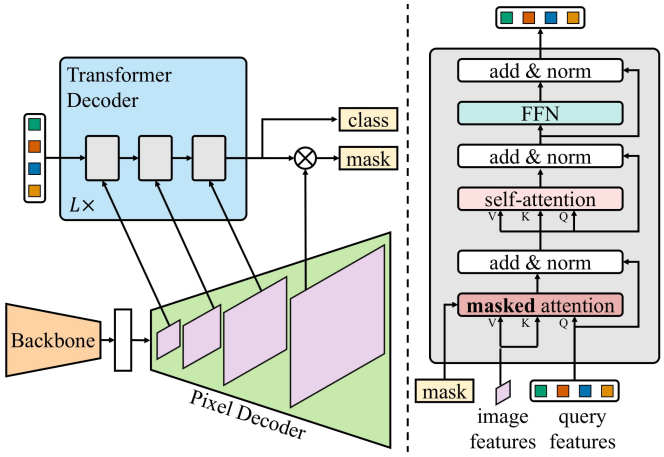

Fig. 3. Mask2Former architecture: The left side displays the different modules, while the right side provides a more detailed schema of the Transformer Decoder [22].

III. Mask2Former for Multispectral Images

This section explains the adaptation of Mask2Former to exploit the spectral information of multispectral images for semantic segmentation.

The Mask2Former model was adapted to handle multispectral image datasets, facilitating image segmentation and analysis across both NIR and RGB bands. Adapting this method for multispectral remote sensing images involves modifying various parts of the approach. First, the training workflow was extended to support multispectral images, which involved adjustments in the dataset handling, model initialization, and overall training management. This ensured the system could process the specialized data formats and metadata associated with multispectral images.

Second, custom mappers had to be created to preprocess the multispectral data and prepare it for the model. These mappers read images in specific formats and remove irrelevant regions, ensuring the data fed into the model is accurate and useful.

Third, the configuration setup was enhanced to define essential parameters such as the number of training iterations, image characteristics, and dataset metadata. These configurations were extended to support different model architectures, specifically adjusting for ResNet50 [25] and ResNet101 [25] backbones.

Finally, the demo script was updated to process multispectral images effectively, incorporating custom data handling methods. This script enables the practical application of the trained model by interpreting input images and producing segmentation results, adjusted for multispectral image specifics.

Changes made to code are detailed in Appendix A. The code is available at the following repository: https://gitlab.citius.gal/hiperespectral/mask2former-for-multispectral-images.

IV. Experiments

In this section, the evaluation focuses on the classification performance of Mask2Former models to assess their overall effectiveness. Different hyperparameter configurations will be tested. Furthermore, comparisons will be made between the multispectral and RGB versions of the models to investigate the influence of multispectral bands on the final results. Finally, a comparison between our proposed models and state-of-the-art models using the same dataset will be presented.

A. Experimental setup

1) Dataset: The dataset Five Billion Pixels [29] is used to train and evaluate the proposed adaptation of Mask2Former for multispectral remote sensing images. This dataset is the last iteration of the Gaofen Image Dataset (GID) [28], a dataset widely used in remote sensing to test and evaluate machine learning algorithms. Its name comes from the Gaofen-2 (GF-2) satellite, which captured the 4-band multispectral images.

Five Billion Pixels, which extends the original GID, is described in Table I. Instead of the 5/15 basic categories of the GID, a more comprehensive category system of 24 land use and land cover categories is used. Fig. 4 shows four images of the dataset with examples of each of the 24 categories.

The Five Billion Pixels category extension fully relied on human annotation, and it was conducted in the following manner [29]:

1) **Category system determination**: Categories are based on Chinese Land Use Classification Criteria (GB/T 21010-2017).
2) **Coarse labeling**: Interpretation experts broadly outline regions corresponding to different categories on each GF-2 image based on the category system. For areas with uncertainty, Google Earth and Google Maps, along with their respective geographic coordinates, are used as reference points.
3) **Fine labeling**: The labeling team uses the lasso tool in Adobe Photoshop to outline ground objects, ensuring that the edges of the segment precisely align with the edges of the ground objects.
4) **Fine and spot checking**: The categories and the edges of each segment are finally checked and corrected by experts.
5) **Final spot checking**: Examination of 10% sample of Five-Billion-Pixels, ensuring that no obvious errors remain.

Five Billion Pixels images are available in two formats: 16-bit non-quantized and 8-bit quantized images, both containing 4 bands: red, green, blue, and near infrared. There are 150 images, 120 for training and 30 for testing. Each image has a size of 6908×7300 pixels and a spatial resolution

TABLE I
GID [28] AND FIVE BILLION PIXELS [29] DATASET INFORMATION.

	Set	Categories	Training	Size	Test
Gaofen Image Dataset	Large-scale Classification	5	120 GF-2 images	6800×7200	30 GF-2 images
	Fine Land-cover Classification	15	30,000 patches	56×56, 112×112, 224×224	10 GF-2 images
Five Billion Pixels	Original	24 + Unlabeled	120 GF-2 images	6908×7300	30 GF-2 images
	Used in this work	24 + Unlabeled	480 GF-2 images	3454×3650	120 GF-2 images

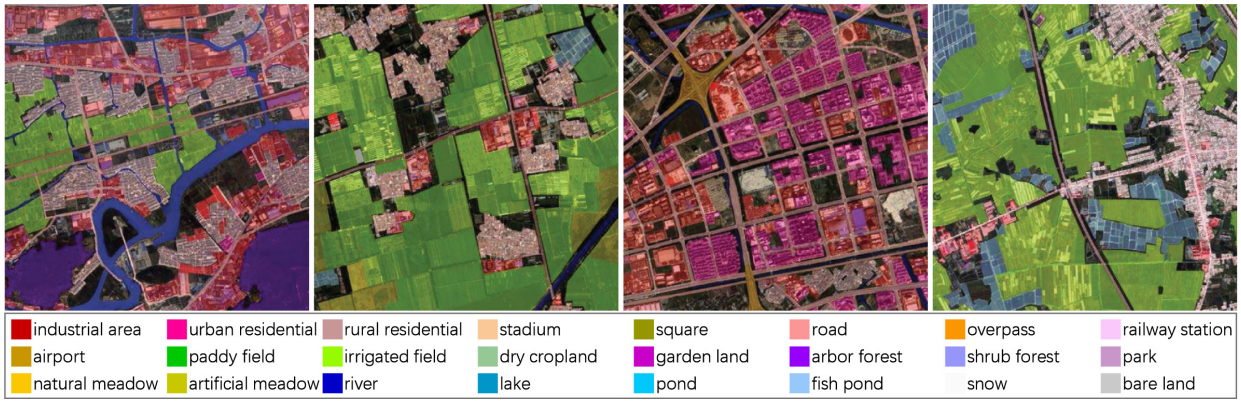

Fig. 4. Five Billion Pixels categories [29].

of 4 m/pixel. The dataset covers areas of 50,000 km^2 in China, from megacities like Beijing and Chengdu to rural and remote inland locations, with variations in climate, geology, and altitude. Color and grayscale masks are provided for each image. Each mask contains all the pixels of the images assigned to a category of the dataset. If a specific area is hard to label, the pixel is then assigned to the "unlabeled" category. These masks are the reference data used for training the algorithm.

To train the network and perform inferences on commodity hardware, all images were divided into quarters, resulting in a size of 3454×3650 pixels. The images were decompressed (TIF format) and saved to disk in raw bitmap format. The original format required reading the spectral bands with specific Python libraries. In addition, the loading of the images to memory is faster. Thus, the modified dataset has a size of 242 GB, of which 193.6 GB correspond to training data and 48.4 GB to test data. This resulted in the modified version of the dataset that can be seen in Table I.

2) Metrics: The metrics implemented are those used in COCO [30] (Common Objects in Context). These metrics are specific to semantic segmentation tasks and widely used in the literature.

To perform the evaluation, Detectron2 provides evaluators, which are classes designed to assess the performance of models on different tasks, such as object detection, instance segmentation, and keypoint detection. These evaluators compute various metrics that provide insights into the effectiveness and accuracy of the models. In this case, we use one for semantic segmentation that calculates the following metrics:

- **Mean Intersection over Union (mIoU)** calculates the intersection over union (IoU) for each category and then takes the average.
- **Frequency Weighted IoU (fwIoU)** takes into account the frequency of each category in the dataset, giving more weight to categories that are more common. It can be useful in datasets with imbalanced category distributions.
- **Mean Accuracy (mACC)** measures the average accuracy per category. It calculates the ratio of correctly predicted pixels to the total pixels in each category and then averages these ratios across all categories.
- **Pixel Accuracy (pACC)**, also known as overall accuracy (OA), calculates the ratio of correctly predicted pixels to the total number of pixels.

3) Execution Environment: All models were trained on the same machine and with the same software versions. The hardware and software characteristics can be seen in Table II.

Models are trained on a desktop computer with an Intel i7-11700K with 128 GB of main memory and an NVIDIA RTX 3080 Ti. As our aim is to use a personal computer setup and not a cluster or a supercomputer, some modifications had to be made to the original Mask2Former: dividing the images in quarters as explained in Section IV-A1 and finding the most efficient hyperparameters combination. This allows us to have a much more accessible network to the general remote-sensing user. The image batch size had a great effect on GPU memory usage, as reference Mask2Former developers trained their models with eight 40GB GPUs.

4) Hyperparameter Optimization: To determine the optimal hyperparameters for our model, we employed a systematic approach of iterative adjustment. This approach involved experimenting with the Learning Rate (LR), number of iterations, and batch size to reach a stable training setup, and then how often to create model checkpoints and to test model perfor-

TABLE II
SYSTEM INFORMATION

Component	Details
Operating System	Ubuntu 22.04.4 LTS
CPU	Intel i7-11700K
RAM	128 GB
GPU	NVIDIA RTX 3080 Ti
GPU memory	12 GB
GPU driver version	550.67
Python	3.8.19
Numpy	1.24.3
Detectron2	0.6
Compiler	GCC 11.4
CUDA compiler	CUDA 12.4
Detectron2 arch flags	8.6
PyTorch	2.2.2
Pillow	10.2.0
Torchvision	0.17.2
OpenCV2	4.9.0

mance, and refining the choices based on these observations. Through this methodical exploration, we were able to identify the configuration that yielded the best results.

The best values we found were a batch size of two images, for GPU memory reasons, and a LR of 0.0001. A lower LR produced a model that did not converge better but took longer to train and a higher LR often did not converge. Regarding the number of iterations to train the model, we found that generally before iteration 30,000 the model that produces the best metrics is obtained, to stagnate in the following iterations with worse results. For this reason, we used 40,000 and 43,000 iterations.

B. Experimental Results

The main objective of the experiments is to evaluate the performance of our Mask2Former for multispectral remote sensing images. Segmentation models using the 4 spectral bands will be compared with the original Mask2Former architecture that only uses the three RGB bands. In addition, we tested different backbones (ResNet50 and ResNet101) to see the difference in the results they provided.

Table III presents the OA, mAcc, mIoU, and fwIoU accuracy metrics of the 4 best models using the four bands or using only the RGB bands of the dataset, as well as using Resnet50 or Resnet101 as backbone.

R50 NirRGB consistently outperforms the other models in all metrics, indicating that the addition of the NIR band significantly improves the performance of the model. Moreover, the best-performing models are all based on the ResNet50, suggesting that it may generalize better than the ResNet101 backbone. From the results drawn from the R101 NirRGB model, we also conclude that simply increasing the complexity of the model, using the NIR band and employing a more complex backbone, ResNet101, does not guarantee better performance.

Seeing that the best results were obtained by models that shared ResNet50 as their backbone, we decided to evaluate them in detail. Fig. 5 shows how the total loss changes for the two types of models (multispectral and RGB) as the number of

TABLE III
ACCURACY METRICS (OA, MACC, MIOU, AND FWIOU) USING THE FOUR BANDS OR ONLY RGB BANDS, WITH RESNET50 OR RESNET101 AS THE BACKBONE

Model	OA (%)	mAcc (%)	mIoU (%)	fwIoU (%)
R50 RGB	79.72	48.87	39.98	66.50
R50 NirRGB	**81.47**	**52.07**	**41.30**	**70.43**
R101 RGB	77.20	49.31	38.37	63.40
R101 NirRGB	76.40	42.62	33.67	63.67

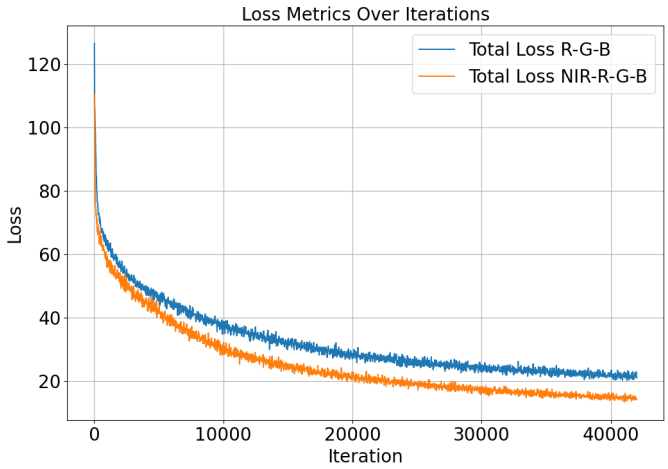

Fig. 5. Total loss comparison using a ResNet50 backbone between multispectral (orange) and RGB (blue) models.

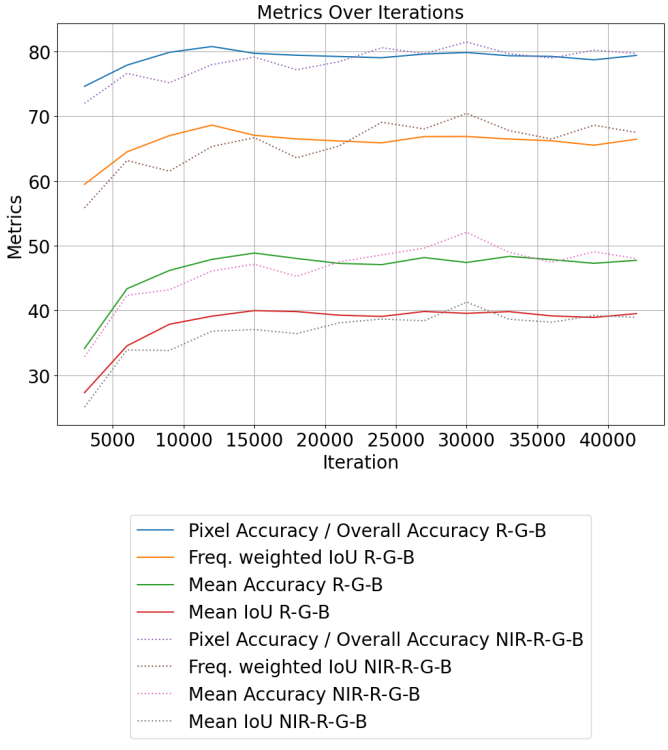

Fig. 6. Metrics comparison using a ResNet50 backbone between multispectral (dotted line) and RGB (continuous line) models.

iterations increases. As we can see, the loss of the multispectral network drops faster. It may indicate that the inclusion of the NIR band enhances the model's learning process. This implies faster convergence, better data utilization, and the possibility of improved accuracy and stability during training.

In addition, Fig. 6 shows how the metrics for the two types of models progress during training. It can be seen how the RGB model obtains better results up to iteration 23,000 approximately. From then on, the model with the NIR band starts to obtain better results. This improvement may be attributed to the additional information provided by the NIR band, which the model adapts to over time.

Table IV presents the IoU, accuracy and frequency of categories for all categories of each of the two models at a category level. At first glance, we can see that the dataset is highly unbalanced with snow representing 0.02% out of the 24 categories and irrigated field being 37.26%. The use of the NIR band reinforced the correct segmentation of certain categories, such as airports, dry croplands, or irrigated fields.

Even seeing the overall improvement that occurs with the addition of the NIR band, in some categories, such as urban residential, overpass or road, the RGB model performs considerably better. This indicates that the improvement by adding the extra band depends on what type of categories we want to segment, indicating that the NIR band does not add information in these cases.

Other interesting results we can observe are the behavior of each network with the three least common categories of the dataset, snow, square, and stadium. The NIR band greatly improves the segmentation of snow and square but not the stadium category. Having an irregular behavior for minority classes is usual in multispectral remote sensing images that are usually unbalanced.

C. Comparison with previous works

To assess the effectiveness of the proposed models, the results are compared with recent studies on semantic segmentation of the Five Billion Pixels dataset, as shown in Table VI. Some results are taken from [31] as this paper addresses the challenges of category imbalance in large-scale land cover mapping using semi-supervised learning. They propose a unified Class-Aware Semi-Supervised Semantic Segmentation framework and also present results of others methods like ResUnet [32], Advent [33], and CADR [34], among others, as shown in the table. Results provided by a multilayer perceptron (MLP) with fusion and a random forest (RF) with fusion [29] are also included, as well as, results provided by U-Net [12], DeepLabV3+ [35], ResNet101 [25], and GoogLeNet [36].

As shown in Table VI, the R50 NirRGB model proposed in this Master Thesis achieves competitive results. In particular, it obtains the best overall accuracy, the second-best mean accuracy and the third-best mean IoU.

TABLE IV
CATEGORY METRICS COMPARISON BETWEEN THE BEST MULTISPECTRAL AND THE BEST RGB MODELS

		R50 NirRGB		R50 RGB	
Category	**Freq. (%)**	**Acc.**	**IoU**	**Acc.**	**IoU**
mean	-	52.07	41.30	48.87	39.98
airport	0.09	52.95	44.33	30.61	27.93
arbor forest	8.05	92.88	88.55	93.11	89.13
artificial meadow	0.36	19.09	11.90	24.13	14.48
bareland	4.16	55.12	33.79	55.44	47.95
dry cropland	6.65	87.92	72.11	66.08	49.97
fish pond	1.12	55.16	47.23	54.82	43.50
garden land	0.91	17.32	13.49	23.51	16.93
industrial area	3.57	68.87	54.15	69.12	58.03
irrigated field	37.26	92.38	83.73	89.94	77.46
lake	9.87	94.19	84.39	96.57	78.88
natural meadow	1.65	80.53	59.63	82.01	59.74
overpass	0.23	25.88	22.92	49.00	41.19
paddy field	2.40	73.14	58.32	58.51	48.22
park	0.05	13.00	10.67	14.06	12.72
pond	1.03	20.30	16.47	15.10	12.66
railway station	0.08	26.43	21.47	22.74	19.36
river	5.08	84.01	75.75	66.79	62.78
road	3.57	55.01	35.89	61.55	42.51
rural residential	4.39	55.62	43.34	68.74	56.43
shrub forest	3.80	14.97	11.28	24.58	16.17
snow	0.03	61.45	22.28	2.89	1.73
square	0.02	8.09	5.23	0.00	0.00
stadium	0.02	14.25	11.22	17.09	15.38
urban residential	5.60	81.11	63.15	86.52	66.43

TABLE V
METRICS COMPARISON WITH PREVIOUS WORK

Method	**OA (%)**	**mAcc (%)**	**mIoU (%)**
ResUNet [31]	73.24	45.77	35.61
SimCLR + finetune [31]	74.07	45.81	35.17
Pseudo-labeling [31]	74.61	48.83	38.03
Advent [31]	74.81	49.25	38.98
CADR [31]	75.49	47.83	36.63
DPA [31]	75.36	48.57	37.77
CASSSS [31]	75.97	53.53	40.68
MLP+Fusion [29]	23.89	-	9.78
RF+Fusion [29]	27.40	-	10.23
GoogLeNet [29]	69.19	-	28.99
ResNet101 [29]	69.55	-	33.59
DeepLabv3+ [29]	79.87	-	42.12
U-Net [29]	80.35	-	44.51
Proposed R50 RGB	79.72	48.87	39.98
Proposed R50 NirRGB	81.47	52.07	41.30
Proposed R101 RGB	77.20	49.31	38.37
Proposed R101 NirRGB	76.40	42.62	33.67

V. CONCLUSIONS

This paper presents the adaptation of a semantic segmentation architecture, Mask2Former, for use in multispectral remote sensing images. This adaptation involves modifying the architecture to accommodate the distinct features and extra bands found in the multispectral data, ensuring that the model can fully utilize the available information to improve the segmentation accuracy.

The proposed application is trained and evaluated using the Five Billion Pixels dataset with 24 land user and land cover categories and 150 high spatial resolution images of 4 bands.

The experiments compare the performance of models trained on multispectral images versus standard RGB images for semantic segmentation tasks. It has been found that the addition of the NIR band improves the model learning process,

TABLE VI
METRICS COMPARISON WITH PREVIOUS WORK

Method	OA (%)	mAcc (%)	mIoU (%)
ResUNet [1]	73.24	45.77	35.61
SimCLR + finetune [1]	74.07	45.81	35.17
Pseudo-labeling [1]	74.61	48.83	38.03
Advent [1]	74.81	49.25	38.98
CADR [1]	75.49	47.83	36.63
DPA [1]	75.36	48.57	37.77
CASSSS [1]	75.97	53.53	40.68
MLP+Fusion [2]	23.89	-	9.78
RF+Fusion [2]	27.40	-	10.23
GoogLeNet [2]	69.19	-	28.99
ResNet101 [2]	69.55	-	33.59
DeepLabv3+ [2]	79.87	-	42.12
U-Net [2]	80.35	-	44.51
Proposed R50 RGB	79.72	48.87	39.98
Proposed R50 NirRGB	81.47	52.07	41.30
Proposed R101 RGB	77.20	49.31	38.37
Proposed R101 NirRGB	76.40	42.62	33.67

leading to faster convergence, and better utilization of data compared to models trained on RGB images.

It has been shown that using a more complex backbone does not imply better results. Both RGB and multispectral versions get worse results when using a ResNet101 as a backbone.

A detailed class-level performance analysis revealed that the addition of the NIR band significantly improved segmentation accuracy for certain categories. This indicated that the benefit of the NIR band is dependent on the specific class being segmented.

Our R50 NirRGB model achieved competitive results compared to recent studies on semantic segmentation with the Five Billion Pixels dataset. In particular, this model obtains an overall accuracy of 81.47%, while the state-of-the-art U-Net reaches 80.35%.

VI. FUTURE WORK

As this architecture has shown competitive results, we plan further research to improve these results. Future lines of work are detailed below:

1) **Backbone fine tuning**: The backbones used are trained on non-multispectral images, we believe that further training them on multispectral remote sensing land cover images could considerably improve the results.
2) **Pixel decoder modification**: The pixel decoder can be modified to use higher-resolution pixel embeddings. We plan also to test different resolutions for the multi-scale inputs of the transformer decoder.
3) **High-performance computing (HPC)**: It could improve results by allowing larger batch sizes, faster training times, and more extensive hyperparameter tuning. HPC enables the handling of more complex models and larger datasets efficiently, so the full images can be used.
4) **Using raw images**: Five Billion Pixels provides two types of images, 16 bit raw images and quantized 8-bit images. As we use the 8-bit images, it would be necessary in a further stage to analyze how the the results are affected by the quantization process.
5) **Datasets with higher resolution**: It would be interesting to evaluate the model with datasets with higher spatial resolution captured by sensors with higher spectral resolution, such as those obtained by the Micasense RedEdge sensor.

REFERENCES

[1] D. Rocchini, C. Ricotta, and A. Chiarucci. Using satellite imagery to assess plant species richness: The role of multispectral systems. *Applied Vegetation Science*, 10(3):325–331, 2007.

[2] Sajjad Hussain, Linlin Lu, Muhammad Mubeen, Wajid Nasim, Shankar Karuppannan, Shah Fahad, Aqil Tariq, B. G. Mousa, Faisal Mumtaz, and Muhammad Aslam. Spatiotemporal variation in land use land cover in the response to local climate change using multispectral remote sensing data. *Land*, 11(5), 2022.

[3] Jon Christopherson, Shankar N Ramaseri Chandra, and Joel Q Quanbeck. 2019 joint agency commercial imagery evaluation—land remote sensing satellite compendium. Technical report, US Geological Survey, 2019.

[4] Mohammad D Hossain and Dongmei Chen. Segmentation for object-based image analysis (obia): A review of algorithms and challenges from remote sensing perspective. *ISPRS Journal of Photogrammetry and Remote Sensing*, 150:115–134, 2019.

[5] Osmar Luiz Ferreira de Carvalho, Osmar Abílio de Carvalho Júnior, Cristiano Rosa e Silva, Anesmar Olino de Albuquerque, Nickolas Castro Santana, Dibio Leandro Borges, Roberto Arnaldo Trancoso Gomes, and Renato Fontes Guimarães. Panoptic segmentation meets remote sensing. *Remote Sensing*, 14(4), 2022.

[6] Shijie Hao, Yuan Zhou, and Yanrong Guo. A brief survey on semantic segmentation with deep learning. *Neurocomputing*, 406:302–321, 2020.

[7] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3523–3542, 2022.

[8] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. Fully convolutional neural networks for remote sensing image classification. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 5071–5074, 2016.

[9] Qishuo Gao, Samsung Lim, and Xiuping Jia. Hyperspectral image classification using convolutional neural networks and multiple feature learning. *Remote Sensing*, 10(2), 2018.

[10] João Lourenço Silva, Miguel Nobre Menezes, Tiago Rodrigues, Beatriz Silva, Fausto J. Pinto, and Arlindo L. Oliveira. Encoder-decoder architectures for clinically relevant coronary artery segmentation, 2021.

[11] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation, 2016.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[13] Sreelakshmi S, Malu G, Elizabeth Sherly, and Robert Mathew. M-net: An encoder-decoder architecture for medical image analysis using ensemble learning. *Results in Engineering*, 17:100927, 2023.

[14] Lin Bai, Wei le Li, Qiangfa Xu, Weihang Peng, Kai Chen, Zhenzhen Duan, and Huiyan Lu. Multispectral u-net: A semantic segmentation model using multispectral bands fusion mechanism for landslide detection. In *CDCEO@IJCAI*, 2022.

[15] Mayank Dixit, Kuldeep Chaurasia, and Vipul Kumar Mishra. Dilated-resunet: A novel deep learning architecture for building extraction from medium resolution multi-spectral satellite imagery. *Expert Systems with Applications*, 184:115530, 2021.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018.

[17] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation, 2018.

[18] Liang-Chieh Chen, Alexander Hermans, George Papandreou, Florian Schroff, Peng Wang, and Hartwig Adam. Masklab: Instance segmentation by refining object detection with semantic and direction features, 2017.

[19] Peng Li, Peng Ren, Xiaoyu Zhang, Qian Wang, Xiaobin Zhu, and Lei Wang. Region-wise deep feature representation for remote sensing images. *Remote Sensing*, 10(6), 2018.

[20] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020.
[21] Yongtao Yu, Tao Jiang, Junyong Gao, Haiyan Guan, Dilong Li, Shangbing Gao, E Tang, Wenhao Wang, Peng Tang, and Jonathan Li. Capvit: Cross-context capsule vision transformers for land cover classification with airborne multispectral lidar data. *International Journal of Applied Earth Observation and Geoinformation*, 111:102837, 2022.
[22] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. 2022.
[23] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 17864–17875. Curran Associates, Inc., 2021.
[24] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.
[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
[26] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering, 2020.
[27] Bowen Cheng, Omkar Parkhi, and Alexander Kirillov. Pointly-supervised instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2617–2626, 2022.
[28] Xin-Yi Tong, Gui-Song Xia, Qikai Lu, Huanfeng Shen, Shengyang Li, Shucheng You, and Liangpei Zhang. Land-cover classification with high-resolution remote sensing images using transferable deep models. *Remote Sensing of Environment*, 237:111322, 2020.
[29] Xin-Yi Tong, Gui-Song Xia, and Xiao Xiang Zhu. Enabling country-scale land cover mapping with meter-resolution satellite imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 196:178–196, 2023.
[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.
[31] Runmin Dong, Lichao Mou, Mengxuan Chen, Weijia Li, Xin-Yi Tong, Shuai Yuan, Lixian Zhang, Juepeng Zheng, Xiaoxiang Zhu, and Haohuan Fu. Large-scale land cover mapping with fine-grained classes via class-aware semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16783–16793, October 2023.
[32] Foivos I. Diakogiannis, François Waldner, Peter Caccetta, and Chen Wu. Resunet-a: A deep learning framework for semantic segmentation of remotely sensed data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 162:94–114, April 2020.
[33] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation, 2019.
[34] Xinting Hu, Yulei Niu, Chunyan Miao, Xian-Sheng Hua, and Hanwang Zhang. On non-random missing labels in semi-supervised learning, 2022.
[35] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation, 2017.
[36] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014.

APPENDIX A
CODE CHANGES TO MASK2FORMER FOR MULTISPECTRAL SEMANTIC SEGMENTATION

In this appendix, the modifications needed in Mask2Former[1] and Detectron2[2] to train multispectral models are detailed [3].

Mask2Former code provides a Python script, called "train_net.py". It encapsulates all handling necessary from configuration setup, dataset registration, and model initialization to data loading, training loop management, evaluation, logging, checkpointing, and hyperparameter tuning.

The first and most notable contribution was to modify this script to load the Five Billion Pixels dataset annotations in the format supported by Mask2Former. This format can be found in the Detectron2 documentation[4]. The annotations format is composed of a list of dictionaries, with each image in the dataset having a unique associated dictionary. Each dictionary contains a key-value mapping with information such as the full path to the image, height, width, ground truth, or bounding boxes. An additional value has been added to manage how the images should be loaded, whether as multispectral images or as standard RGB images. In addition to the annotations, the dataset metadata is also added, which is the information shared by all the images in the dataset. Examples of this metadata are the number of classes, their names, whether they are stuff or things, which class is used to mark the pixels to be ignored or the type of evaluation to apply to the test set. All of this loading of the dataset data is encapsulated in the new method added to the original script and is performed for both the training split and the test split.

In Mask2Former and Detectron2, dataset mappers are in charge of handling data preprocessing, augmentation, and any necessary conversions required to prepare the data for the model. As a summary, they take each dictionary and return a data format ready to be consumed by the model. The changes applied were:

- **Mask2Former mapper**: Using a modified read function capable of reading dataset images saved in custom raw-band format and the correct removal of unlabeled regions in the image. This mapper is used in the training phase.
- **Detectron mapper**: This mapper is used in the evaluation phase by Detectron2, it was modified to use the new customized reading functions.

Even though the mappers take care of obtaining a format that can be fed to the network, they do not remove dimensions, this results in an image: dimensions $\times$ height $\times$ width. This is a problem since the network expects three dimensions red, green and blue, but now four arrive with the NIR band as a layer. To solve this problem, the network makes use of a stem. The stem is the first layer of the network that is in charge of downsampling the network input, so that it can later be correctly processed. This block, for multispectral images, receives an input with a shape of $4 \times 800 \times 864$ and downsamples it to produce a format with which the network can train.

The Detectron2 "detection_utils.py" contains common data processing utilities that are used in a typical object detection data pipeline. The file was modified to standardize the image

[1] Mask2Former code: https://github.com/facebookresearch/Mask2Former
[2] Detectron2 code: https://github.com/facebookresearch/detectron2
[3] All the code modified and weights can be accessed via this repository: https://gitlab.citius.gal/hiperespectral/mask2former-for-multispectral-images
[4] Detectron2 annotations specification https://detectron2.readthedocs.io/en/latest/tutorials/datasets.html#custom-dataset-dicts-for-new-tasks

reading functions used by our network, now the images are loaded in the same way in all parts of the code.

All the configurations used by the model had to be changed. Mask2Former configurations are made of two parts, the base configuration and the decorator configuration, which add extra hyperparameters. We make use of three different configurations:

- **Base configuration**: This configuration is not called directly but is used as the base configuration for the rest, it is responsible for setting the number of iterations, image characteristics such as the mean value and standard deviation of each band, the name of the train and test dataset in the metadata, which dataset mapper to use, how many iterations to save the models or when to perform testing.
- **ResNet50 configuration [25]**: This configuration decorates the base configuration adding different hyperparameters.
- **ResNet101 configuration [25]**: It adds more info to both of the configurations explained before but changes the weights of the backbone so that it uses a ResNet101.

Finally, the demo script was modified. It is in charge of providing, from a configuration file, an input image and a set of weights, the result of the trained image segmenter. In order to put it into operation, it is necessary to again provide information about the dataset, the annotations and the metadata. This is because, during execution, this file is not in the same memory space as the one on which it was trained. In addition, we also have to modify the reading method to use the new custom one. These images are sent to a Detectron2 method that uses the OpenCV library, due to how it is implemented the order of the channels must be inverted, from NIR-R-G-B to B-G-R-NIR. It also supports working on models that use only 3 bands.