Semantic Segmentation of Satellite Imagery: an Empirical Study on Deep Architectures

Matteo Lugli
Unimore
283122@studenti.unimore.it

Nicola Morelli
Unimore
284023@studenti.unimore.it

Pietro Moriello
Unimore
284427@studenti.unimore.it

Abstract

In recent years, the importance of remote sensing, and in particular satellite imagery, has significantly grown. With satellites such as NASA's LANDSAT, China's Gaofen from the National Space Administration, and the European Space Agency's (ESA) Sentinel-1 and Sentinel-2, we have access to a continual stream of extensive satellite imagery, providing valuable data for research on a daily basis. Leveraging advanced semantic segmentation techniques and expert annotations, we can generate segmentation masks of considerable importance for political decision-making, environmental management, and agricultural applications. This study delves into optimal practices for training deep learning models on remote sensing semantic segmentation. We introduce a straightforward yet effective method for extracting pertinent training data from large-scale datasets, addressing the challenge of class imbalance. The proposed approach allowed us to train state-of-the-art models using approximately 11% of our training set, which is a subset of the GID15 dataset, while achieving performance comparable to that of models trained on the full dataset. Additionally, we show a method for retrieving similar images using segmentation masks produced by our models, comparing it with a method based on deep learning features extracted from DINO-ViT.

1. Introduction

Remote sensing is the comprehensive study of satellite imagery acquired through orbiting satellites, drones, or aircraft. It proves highly valuable in agricultural and environmental contexts, as well as in urban planning or risk assessment. The popularity of competitions such as DeepGlobe [11] serves as evidence of this, featuring challenges related to road land cover classification, building detection and road extraction. Currently, the European Union is making substantial investments in satellite data acquisition. Through the Copernicus [16] program, it has deployed the Sentinel satellites into orbit, significantly enhancing its ability to monitor and analyze environmental changes. In recent years, many annotated datasets have been released that leverage ESA satellite images, which are valuable resources to train machine learning and deep learning models. [33] can be used for flood monitoring, and it provides useful information that can be exploited to improve disaster prevention, response and management. EuroSAT [21] contains Sentinel-2 imagery labeled for land cover classification, covering different classes such as forest, agricultural fields or urban areas. ESA itself has released WorldCover [17], a vast dataset that provides a segmented map of the entire Earth's surface, categorized into 11 distinct classes. By enabling the training of models for comprehensive monitoring and in-depth analysis, these datasets make it possible to evaluate the health and environmental conditions of our planet. However, they also present several challenges. First of all, it's harder to fine-tune [49] pre-existing models on remote sensing related tasks. Many state-of-the-art models are trained on generic datasets (think of ImageNet [12], COCO-Stuff [2] or ADE20K [51]), which differ significantly from those used in remote sensing. Another significant issue is class imbalance, particularly present in semantic segmentation annotations. These problems paired with the enormous amount of images present in most of these datasets make training deep models a challenging task with limited computational resources. In this work, we performed extensive experiments to converge towards optimal practices to train state of the art semantic segmentation models on remote sensing imagery. We present and evaluate the impact of a simple yet effective strategy that we used to reduce the amount of data and solve the class imbalance problem in our dataset. We call this tecnique 'EBB' (Entropy Based Balancing). We also release weights of 4 modified versions of state of the art semantic segmentation models (U-Net, DeepLabV3-Resnet101, DeepLabv3-MobileNet and Segformer) specifically fine-tuned on remote sensing imagery. Moreover, we performed an in-depth analysis of the models' behavior by comparing their outputs and conducting retrieval. For similarity measurement, we

primarily relied on the segmentation mask class histogram generated by above mentioned models and deep features extracted from DINO-ViT [5]. This approach provided valuable insights into how the models encode and distinguish different input patterns. In section 2 we review some of the most recent works related to remote sensing semantic segmentation. In section 3 we present the main features of the GID15 dataset and we illustrate the EBB algorithm. In section 4 we discuss the choices we made concerning deep learning architectures and report the used training strategies and hyperparameters. We also provide a detailed formulation of the metrics used for performance evaluation. In section 5.1 we analyze the obtained results. Section 5.2 is entirely dedicated to retrieval experiments and analysis. Here we show that employing segmentation masks for retrieval is useful, especially the class distribution of the masks, and we propose a method based on the Earth Mover's Distance and class-ordering to be used as a distance measure. Then we compare this method with vector distances between embeddings extracted from Vision Transformers trained with self-supervision (DINO-ViT) and from the backbone of one of our models. In chapter 5.3 we conduct ablation experiments to measure the impact of the proposed methods.

2. Related Work

2.1. ConvNets

Since the advent of AlexNet [29] convolutional neural networks have seen a dramatic rise in popularity for tasks related to computer vision, including image classification, object detection [18], and semantic segmentation [30]. Following AlexNet, several highly influential works extended the deep convolutional architecture to enhance learning capabilities, improve training stability, and optimize resource utilization. Among the most significant and noteworthy architectures to highlight are GoogleNet [43], VGG [42], and ResNet [20]. The ability of these models to extract useful features has proven to be highly beneficial even in really complex scenarios, and set the cornerstone of many of the impactful works that would follow in the subsequent years. Semantic segmentation has always been a challenging task, as highlighted by Long et al. [31], who thoroughly discuss the complexity of accurately segmenting objects in diverse and intricate scenes. They also introduce FCN (Fully Convolutional Networks) which is one of the first attempts to exploit deep features to extract a dense segmentation mask at pixel level. A notable advancement in this field was also made by Ronneberger et al. who introduced the U-Net architecture [38], an encoder-decoder network initially designed for medical imaging purposes. The encoder part consists of the repeated application of two 3x3 convolutions each followed by a *ReLU* (Rectified Linear Unit) activation function and a maxpool operator. The goal of the decoder is to obtain a segmentation mask by expanding the hidden embedding: the authors employ up-convolutions (upsampling of the feature map followed by 2x2 convolutions) to progressively upscale the features, ultimately achieving the output in the desired shape. More modern implementations of this architecture also make use of batch norm [26] to stabilize training and transpose convolution [31] [1] to achieve better performance. Moreover, they exploit skip connections between encoder and decoder hidden features to improve the ability of the network to preserve spatial information. This design became the baseline for many other architectures like U-Net++ [52] or U-Net3+ [24], in which the authors explore the idea of dense skip connections. While these advancements focused on carefully connecting the encoding and decoding path, other approaches, such as PSPNet [50] and the DeepLab series [8], shifted their focus towards capturing multi-scale context and improving the network's ability to handle variations in object size and shape. PSPNet, for instance, employs a pyramid pooling module to aggregate contextual information at different scales, which is crucial for precise segmentation of objects with varying sizes. Similarly, DeepLab incorporates atrous convolution and fully connected conditional random fields (CRFs) to achieve high-resolution segmentation outputs without significantly increasing the computational load.

2.2. Attentive architectures

In more recent years we saw the exponential increase in popularity of the transformer architecture [47] also in computer vision. The groundbreaking work of [15] et al. demonstrated how to effectively use transformers for image classification tasks, by proposing a new innovative non-CNN based architecture called ViT (Vision Transformer). The latter essentially consists of 3 main components:

1. Preprocessor: the original image is partitioned into fixed-sized patches, which are then projected into a hidden space and augmented with positional encoding. The embeddings along with an additional artificial token to identify the learned class are then forwarded through the encoder.

2. Encoder: the encoder adopts a combination of self-attention, layer normalization and multi-layer perceptron (MLP) to complement the embeddings with contextual information, similarly to the original transformer.

3. Decoder: the features are passed to a simple MLP head that finally classifies the image.

This paradigm shift soon had a significant impact on semantic segmentation as in many other fields of research [13] [7]. Xie et al. [48] obtained impressive results on datasets like Cityscapes [10], ADE20K and COCO-Stuff

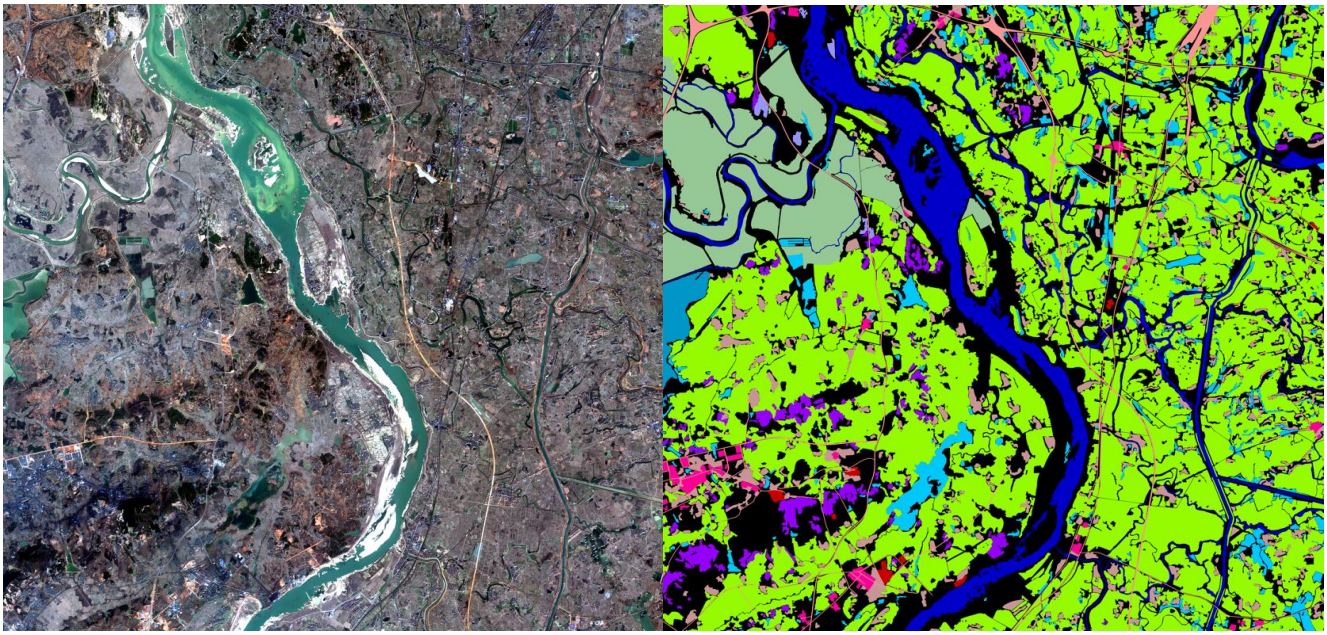


Figure 1. GID15 example RGB image (left panel) and corresponding segmentation mask (right panel).

with the novel SegFormer, which employed a cutting-edge architecture based on the Transformer framework. Many scholars worked on the integration between CNN-based and transformer-based architectures, releasing hybrid models like TransU-Net [6] or SwinU-Net [3].

2.3. Popular approaches in remote sensing

Most of these architectures find application in Remote Sensing, which involves acquiring information (images in this case) about objects or areas from a distance, often using satellite or aerial sensors. The training of deep architectures is particularly challenging for this task, due mainly to difficulties in obtaining quality and abundant annotations, and to heavy class-imbalance. To the best of our knowledge, the two main approaches to tackle remote sensing image segmentation are: *(i)* classic deep learning strategies based on supervised learning and *(ii)* semi-supervised learning to exploit the availability of large amount of unlabeled data. Concerning classic strategies, [27] presents a dual-path U-Net that makes use of both RGB images and NiR (Near infra Red) images taken from a manually balanced subset of GID15 (3). In [4] the authors introduce GFFnet, a network that adopts both an attention based branch to extract global context information and a convolutional network as the local feature extraction branch. Talhaa et al. [44] worked on the decoder part of a U-shaped network using both an attention module and dense skip connections, underlining the importance of decoding with respect to encoding. We also reviewed works such as [45] [14] [46] where the goal of the authors is to use semi-supervised learning in order to alleviate class-imbalance. [14] proposes the automatic creation of a semi-supervised learning dataset more balanced towards minority classes, coupled with a class-balanced cross entropy loss and contrastive learning to regularize feature embeddings. [46] employs a siamese network to perform unsupervised domain adaptation between densely labeled and unlabeled images: one branch outputs pixel pseudo-labels and the other a segmentation mask; a joint loss functions is then constructed with these outputs. Unlike the mentioned studies, our work primarily focuses on an empirical investigation of existing architectures, aiming to identify the most effective techniques for training them. In addition, we conduct an in-depth analysis of the diverse behaviors exhibited by the architectures, including the observation of activation maps and the use of generated masks for retrieval tasks. This approach provides a solid foundation for developing new models.

3. Data

3.1. Introduction to GID15

We decided to use GID15 [45] as our main dataset. It contains 150 pixel-level annotated images, which are labeled in 16 categories: unlabeled (background), industrial land, urban residential, rural residential, traffic land, paddy field, irrigated land, dry cropland, garden plot, arbor woodland, shrub land, natural grassland, artificial, grassland, river, lake, and pond. The images were acquired by Gaofen-2, the second high-resolution optical Earth observation satellite developed by China National Space Admin-

istration (CNSA) as part of CHEOS (China High Resolution Earth Observation System). Launched in 2014, Gaofen-2 is designed to capture optical imagery with a resolution of up to 0.8 meters in panchromatic mode and up to 3.2 meters in multispectral mode. The RGB images contained in the dataset have a resolution of 7200×6800 at 4m per pixel and have been manually annotated by experts, resulting in a highly diverse and valuable dataset to work with. An example image with the corresponding mask is reported in Figure 1. In this work, we randomly picked 90 images for training, 10 for validation and 5 for testing. Since the images are

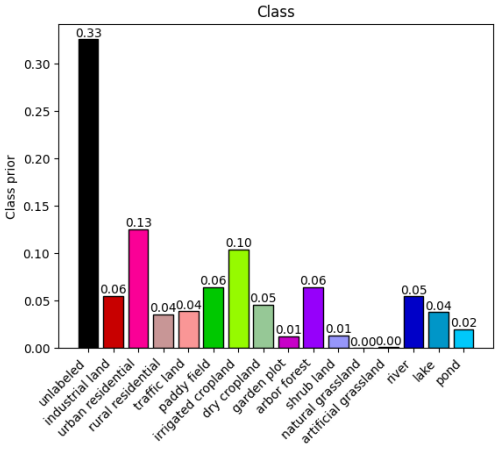


Figure 2. Class priors of GID15. Bar colors correspond to colors used for segmentation masks.

too large to be directly forwarded through a neural network, it is common practice in remote sensing segmentation to divide them into smaller non-overlapping patches. Specifically, we decided to use 224×224 as our patch size. Unlike other works [27] [23], we choose the train-validation-test split a priori (before cropping the patches). This procedure avoids inducing bias in performance evaluation, as the network won't have to classify a patch related to a geographically close region (belonging to the same full-size image) to one it has already seen during the training phase.

3.2. Entropy-Based Balancing

After plotting class priors shown in Figure 2 we noticed a significant imbalance in the distribution of classes. Most of the pixel are unlabeled, and many classes such as shrub land, garden plot, artificial grassland or natural grassland are under-represented. For this reason, we decided to adopt an automatic balancing strategy that we found extremely beneficial when working on datasets of this kind. We call this technique *Entropy-Based Balancing* (EBB), outlined in Algorithm 1. The core idea behind this methodology is to reduce the size of the dataset while keeping the class balanced. For reference, we use the following notation:

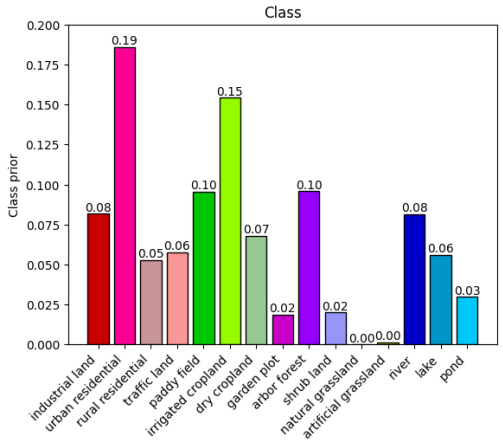


Figure 3. Class priors of GID15 dataset excluding pixels labeled as background.

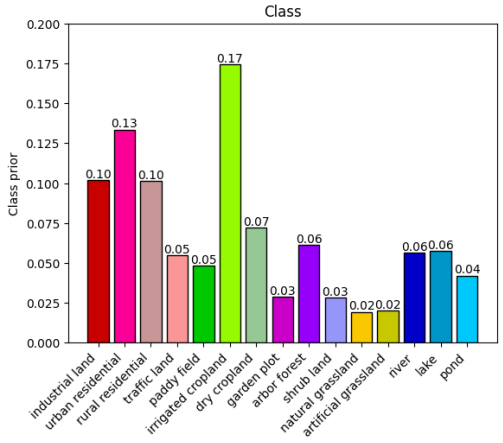


Figure 4. Class priors of GID15 after applying EBB using $\mathcal{S} = 1680$ and granularity of 512.

- $\mathcal{N}$: new balanced dataset;
- S: desired size of $\mathcal{N}$;
- $\mathcal{D}$: original dataset cropped in non-overlapping tiles;
- $f(t)$: function that counts the number of pixels belonging to each class of given a labeled tile t;
- $H(x)$ is the entropy by Shannon, defined as the following:

$$H(X) = -\sum_{x \in X} p(x) \log p(x)$$

In this work we used a granularity of 512, meaning that $\mathcal{D}$ has been filled with tiles of size 512×512. We used desired size S of 1680. We made this choice because of computational constraints, but reducing the granularity can improve the balancing even further. Indeed, the primary draw-

back of this method is its computational cost, which scales quadratically with the number of tiles. Figure 3 shows the probability distribution of classes in $\mathcal{D}$ (unlabeled pixels are excluded) before the balancing procedure, while Figure 4 shows the balanced distribution of the resulting reduced dataset $\mathcal{N}$. The former is heavily skewed towards urban residential, with artificial and natural grassland being statistically negligible. EBB solves the mentioned problems, while also balancing river,lake and pond. It's interesting to see how the new distribution is skewed towards irrigated cropland, showing that the class is probably present in patches that are taken by the algorithm to balance the rest of the distribution. After EBB we use the obtained tiles to get the 224×224 training samples (*patches*) by further cropping them, obtaining 4 patches for each tile. To prevent overfitting and improve generalization capabilities of our model, we apply several data augmentation techniques, such as random rotation, gaussian blur and rescaling. We present some

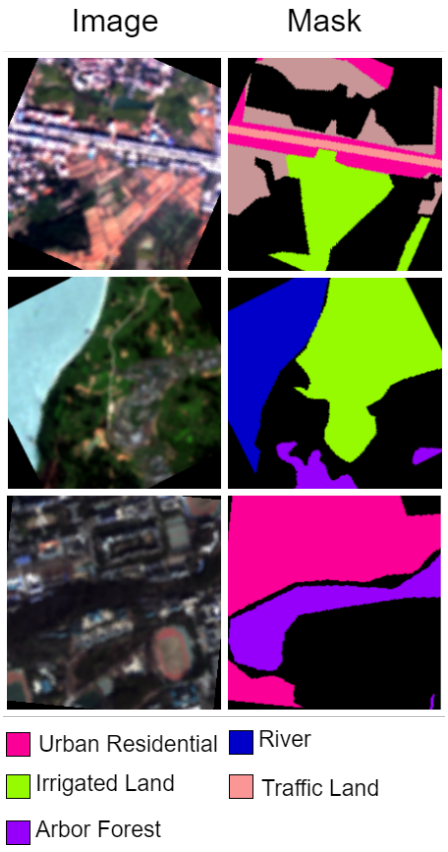


Figure 5. Training samples: augmented images (left) and corresponding masks (right).

training samples in Figure 5. We also apply what we call *Random Shift*: instead of cropping training samples as exactly non-overlapping patches, we randomly shift the selected area of a small amount of pixels (20, about 9% of patch size) vertically and horizontally. This way the model utilizes most of the available information during each epoch while effectively avoiding overfitting.

Algorithm 1 EBB

initialize empty cumulative histogram C
$\mathcal{N} = \varnothing$
for i in range S **do**
  $E = \varnothing$
  for each tile t in $\mathcal{D}$ **do**
    $h = f(t)$
    append $[H(C + h), t, h]$ to E
  end for
  m = elment of E with min $H(C + h)$
  append $m[1]$ to $\mathcal{N}$
  remove $m[1]$ from $\mathcal{D}$
  $C = C + m[2]$
end for

4. Methods

In this section we present the architectures selected for training (4.1), our evaluation metrics (4.2), and a simple yet effective technique employed to achieve visually clean results during inference (4.3).

4.1. Models

We selected 4 popular models to train:

- **U-Net**: we have chosen this model as it is a well known standard for semantic segmentation. Its architectural simplicity allowed us to easily make modifications to the original structure [38] according to our needs. We considered two different variants of this model: the first implements the "up-conv" block using bilinear upsampling and a standard 3×3 convolutional layer, while the second employs a learnable 2×2 transpose convolution with padding set to 2. We found that the latter performed better, showing improved learning capabilities. Accordingly, this was adopted as the 'baseline' for our analysis. We added a batchnorm layer after each encoding block to further improve training stability. Moreover, we trained these models from scratch, with randomly initialized weights. Differently from the original model, we upsample the features to obtain a segmentation mask with the same shape as the input image. We change the last 1×1 convolutional layer to output 16 channels as it is the number of classes for our dataset.

- **Resnet101**: we trained this architecture using DeepLabV3 [9] as it has been proven to achieve impressive results in semantic segmentation. The Atrous

Spatial Pyramid Pooling (ASPP) exploits dilated convolutions applied in parallel on the feature map extracted by the backbone to grasp multi scale information. In the end, CRF(Conditional Random Field) is used to refine the output and obtain the segmentation mask.

We used Pytorch implementation [36] with pre-trained weights on COCO and finetuned it on our dataset. Since the original model returned logits as a 20-dimensional tensor, we modified the output channels of the final layer to be consistent with our dataset by setting them to 16.

- **MobileNet**: we also used this architecture combined with DeepLabV3 to provide a lightweight alternative compared to the other models. This allowed us to evaluate its efficiency while maintaining a smaller model size. We used PyTorch implementation of MobileNet [34] and employed the same set of modifications previously applied to ResNet101.

- **Segformer**: this model allowed us to benchmark the previously mentioned models against a state-of-the-art attention-based architecture specifically designed for semantic segmentation. Segformer extracts coarse context maps and fine local features to further extend the idea of using self-attention in the encoding part first introduced by ViT, which is only capable of extracting low-resolution features. It also implies a simple yet effective MLP decoder which fuses the multi-scale feature maps obtained by the encoder without using complex or computationally demanding modules. We used the Hugging Face [25] implementation written by Niels Rogge [37] with pretrained weights from ADE-512-512 as our baseline. Since the output shape of the mentioned model is (150, $\frac{H}{4}$, $\frac{W}{4}$), we added a custom head to further refine the architecture and obtain the desired shape. Firstly the last classifier head was replaced with a conv-layer with kernel size of 1 in order to have 16 channels as output, followed by two layers of transpose convolution to upsample the output to the original shape.

4.2. Evaluation Metrics

We used several standard metrics to evaluate the performance of our models. We compute our metrics by means of True Positives (TP), False Positives (FP), and False Negatives (FN). An example confusion matrix is shown in figure 6. We use N as the total number of classes, and i to specify a single class.

- **IoU** per class:

$$IoU_i = \frac{TP_i}{TP_i + FN_i + FP_i} \tag{1}$$

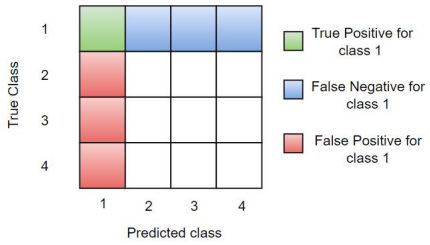


Figure 6. Example confusion matrix.

- **mIoU** :

$$mIoU = \frac{1}{N} \sum_i IoU_i \tag{2}$$

- **mPrecision:**

$$mP = \frac{1}{N} \sum_i \frac{TP_i}{TP_i + FP_i} \tag{3}$$

- **mRecall:**

$$mR = \frac{1}{N} \sum_i \frac{TP_i}{TP_i + FN_i} \tag{4}$$

- **mF1** score:

$$mF1 = \frac{1}{N} \sum_i 2 \times \frac{P_i * R_i}{P_i + R_i} \tag{5}$$

where P_i and R_i denote respectively precision and recall for class i.

- **OA**: letting M denote the confusion matrix, we compute overall accuracy as the sum of the values on the main diagonal divided by the sum of all the values. In the following formula, i and j are used to indicate rows and columns respectively.

$$OA = \frac{\sum_i M_{ii}}{\sum_i \sum_j M_{ij}} \tag{6}$$

4.3. Inference

We perform inference on 224×224 patches and obtain the corresponding segmentation masks. We also create full sized 7200×6800 images (same size as the GID15 images) by assembling the patches together. By doing so we observed a considerable loss of information on the patch borders, resulting in visible lines between patches. In 5.1 we analyze how this behaviour can be related to unwanted border activations in the hidden feature maps. We call this problem "tiling" and we managed to solve it inspired by

the mirroring method presented in [38]. Instead of performing the forward pass on a 224×224 patch, we apply what we call *Border Correction* and make inference on a slightly bigger patch (256×256) then crop the output to only keep the central 224×224 portion of the segmentation map. A demonstrative example of border correction is presented in Figure 7.

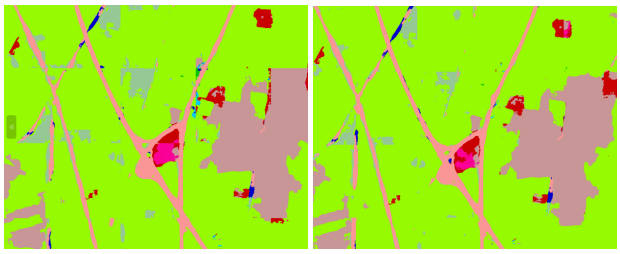


Figure 7. On the left, patches are aligned without border correction. On the right, we avoid tiling by applying border correction.

5. Experiments

To enable the efficient conduction of numerous experiments by varying hyperparameters, we developed our own open source framework [32] based on PyTorch. It is a virtual lab thought for semantic segmentation of remote sensing images that allows users to launch training sessions, inference and evaluation using a simple YAML configuration file.

We trained four different architectures on GID15 reduced dataset (using our EBB balancing strategy on the training set) and compare their performance on the validation set (Table 1) and training set (Table 2) using a patch size of 224×224, which is the same patch size used during training. Every model has been trained for at least 50 epochs which we found to be more then enough to achieve convergence. We ignored the background pixels during training, as we proved to be the optimal choice (5.3). For this reason, we don't consider background in the metrics calculations. We tested two different combinations of optimizer and scheduler, which proved to be the best during our experiments: *(i)* SGD1 employs the SGD [40] optimizer (lr=0.006, momentum=0.9, weight decay=0.00001) with the PolynomialLR [35] scheduling strategy with 20 as the number of steps, while *(ii)* ADAM1 uses ADAM [28](lr=0.005) and the PolynomialLR scheduler with power=2 which makes the lr drop slightly faster after each epoch.

Figure 8 shows full size ground truth images taken from the validation set and the corresponding output for each of our models. They are obtained by assembling 960 224×224 patches using Border Correction as explained in section 4.3. Note that since we ignored unlabeled pixels during training, the output contains no black pixels as the models never predicts background. Moreover, we segment Sentinel-2 images (RGB, 10m per pixel) to test our models on a different domain then the training one. An example is shown in figure 10.

5.1. Discussion

As Table 1 shows, the models all demonstrate comparable results, with no model performing significantly worse than the others and U-Net showing slightly inferior performance. Interestingly, MobileNet achieves great results (mIoU of 0.49, mR of 0.67, mF1 of 0.62 and OA of 0.76) despite the low number of parameters. This could be because the model, with its lower learning capacity compared to the others, is forced to focus more on abstraction rather than on fine details. This trait allows it to perform better on a satellite image dataset, where understanding the broader context of an area of interest is more important than capturing minor details. We support this hypothesis with a significant example: Figure 10 shows the segmentation of two Sentinel-2 image using MobileNet and U-Net. In image 10 **A**, it is evident that U-Net attempts to accurately delineate the boundaries of the urban area at the bottom, whereas MobileNet demonstrates a less precise approach. However, the latter generates a more satisfactory segmentation mask by abstracting minor details and producing a cleaner result, particularly in the upper section. This difference in the two models is also reflected in the activation maps shown in Figure 9. U-Net exhibits features that are considerably less refined compared to those of MobileNet, which are far less "noisy".

Indeed, U-Net was originally designed for medical imaging, which requires a high degree of precision and localization, achieved thanks to multiple residual connections between the encoder and decoder paths. In contrast, MobileNet, when trained with DeepLab, leverages atrous convolutions to enhance its ability to analyze the image across multiple scales. Weird border activations are clearly visible in U-Net feature maps, and are also present in MobileNet's, though to a lesser degree. This is what causes tiling when adjacent patches are stitched together to produce full-size maps. Figure 8 displays several full-size images taken from the validation set, along with full-size segmentation masks, each obtained by aligning 960 segmented patches produced by our models using Border Correction. As Figure 8 **B** shows, MobileNet grasps more valuable information around the river, which is a tough area to classify using small patches. Indeed, all the models fail to segment the river's internal patches correctly, instead predicting the class 'pond'. The confusion matrix reported in Figure 11 suggests that not being able to distinguish between river, pond and lake is a common problem among all architectures. This is due to the fact that it's difficult for the model

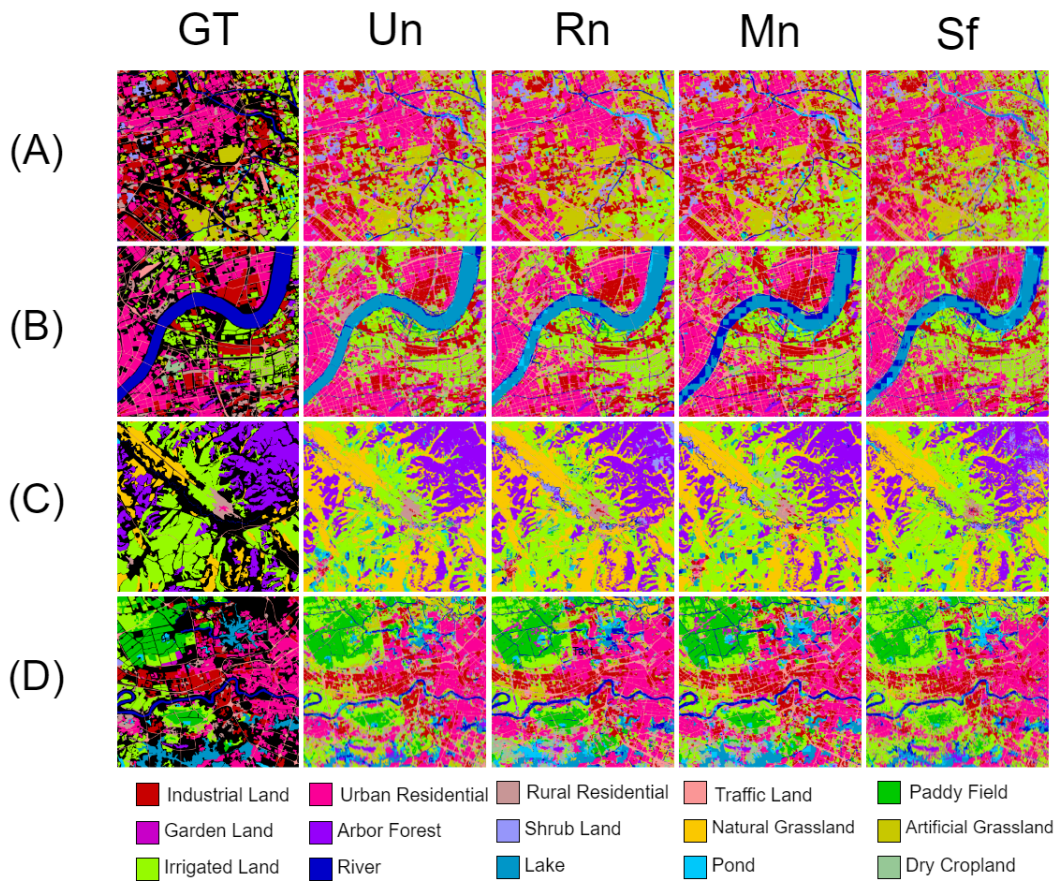


Figure 8. Full size inference (7200 × 6800) for 4 sample images taken from the validation set. From left to right we present the output of U-Net(Un), Rn(Resnet101), MobileNet(Mn) and Segformer(Sf). GT stands for Ground Truth.

Table 1. Performance comparison of tested architectures on validation set.

Model	Parameters	Optimizer	Evaluation Metrics				
			mIoU	mP	mR	mF1	OA
U-Net [38]	33M	SGD1	0.43	0.56	0.62	0.57	0.72
DeepLabV3-Resnet101 [36]	61M	ADAM1	0.47	0.60	0.62	0.60	0.76
DeepLabV3-MobileNet [34]	11M	ADAM1	**0.49**	0.61	**0.67**	**0.62**	0.76
Segformer [37]	47M	ADAM1	0.46	0.60	0.62	0.60	0.76

to correctly segment a patch completely covered by water and distinguish if it's part of a river, lake or pond without seeing the full picture. This particular problem does not occur in Figure 8 **D**, where the river is narrower. Almost all models except for U-Net are able to segment it correctly for the most part. All models misclassify ponds by labelling them as paddy field, as they are hard to distinguish for the human eye as well. Figures 8 **A**,**B** and **D** highlight the models' excellent ability to segment urban areas, particularly in identifying roads and residential areas. Figure 8 **C** shows that the models are also capable of segmenting regions covered by less frequent classes, such as 'arbor forest' or 'natural grassland'. It is challenging to directly compare the performance of our model with that of other researchers due to variations in the train-validation-test splits used in different studies. These splits may have been predetermined or randomly generated after image cropping, further complicating direct comparison. Despite this, our results are consistent with those reported on the test set in [27], although their approach relies on a smaller amount of data compared to ours. Our results are also in line with the ones reported in [41], which also use a different subset of GID15

Table 2. Performance comparison of tested architectures on test set.

Model	Parameters	Optimizer	Evaluation Metrics				
			mIoU	mP	mR	mF1	OA
U-Net [38]	33M	SGD1	0.34	0.42	0.49	0.43	0.71
DeepLabV3-Resnet101 [36]	61M	ADAM1	0.39	0.48	0.55	0.48	0.80
DeepLabV3-MobileNet [34]	11M	ADAM1	0.38	0.46	0.58	0.48	0.79
Segformer [37]	47M	ADAM1	0.31	0.40	0.46	0.40	0.73

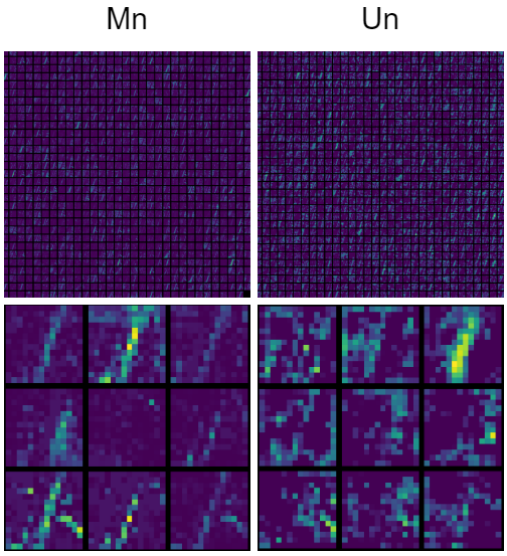


Figure 9. Top left: last layer activations of MobileNet(Mn). Top right: last layer activations for U-Net(Un). The lower panels display zoomed-in areas of the images above.

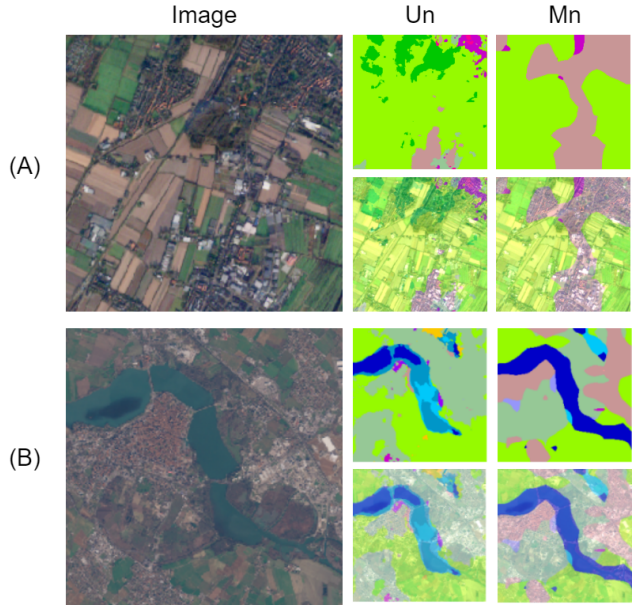


Figure 10. Sentinel-2 images (10m per pixel resolution) and corresponding segmentation masks, with linear blends showing the overlap between mask and image.

and patch size. In [46] the authors present an updated version of GID15 called 5BP (*Five Billion Pixels*), retaining the same images but updating the masks, which can now include up to 25 classes. While 5BP is substantially different from GID15, they obtain similar performance to ours when using the same architectures, even though they experimented with a variety of other models.

5.2. Retrieval

Segmentation masks convey high-level semantic information on satellite images that can be used to compare them. Such a retrieval system is robust against the visual variance of each class; this means that the impact of differences between instances of the same class is mitigated. Initially, we explored different scores to evaluate similarity: mean IoU, weighted IoU, pixel precision and EMD. Apart from the latter, all other scores are mainly influenced by spatial correspondence of pixels; Earths Mover's Distance [39] is computed on signatures (histograms generalizations) instead, in this case on the normalized class distribution of output masks. We argue that scores that rely on spatial information can be misleading, particularly if the main focus is on retrieving images based on semantic content.

EMD requires defining a ground distance matrix $\mathbf{D}$ that best represents inter-class relationships. To achieve this, an ordering of the discrete indexes associated to labels needs to be defined. Since the association present in our reference dataset already yields a semantically sound ranking (similar classes are close together in the ordering), we decided to stick with it. We do not further explore different orderings. The resulting distance between ordered classes $\mathbf{c}_i$ and $\mathbf{c}_j$ is $\mathbf{D}_{ij} = |j - i|$.

Now, any two distributions considered will be sorted, will have equal masses and the distance matrix results in a one dimensional embedding, which brings the EMD to the special case of the Mallow distance, as explained in [22]; here we employ the L_1 distance, as already stated.

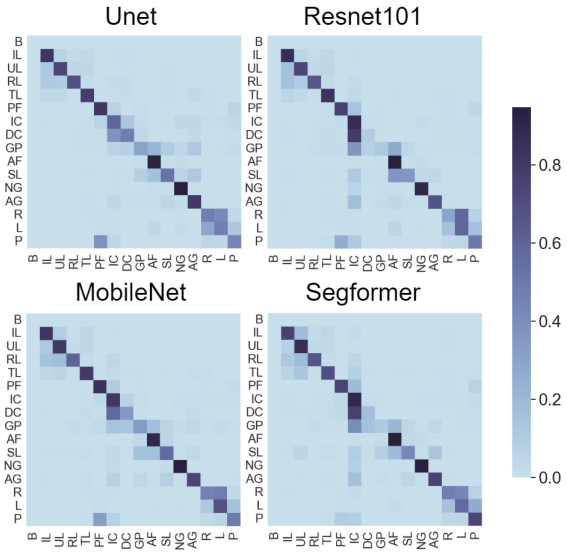


Figure 11. Confusion matrix for the evaluated models on the validation set, illustrating the distribution of true positives, false positives, true negatives and false negatives for every class.

Going back to the advantages of EMD, Figure 12 presents an example of the possible pitfalls of scores with a strong positional bias. As the reported results show, "spatial" scores are influenced by overlaps of common classes, which may overweight other dissimilarities. This leads to situations in which masks with matching semantic contents are not identified as such due to segmentation regions not being found in corresponding locations across two masks.

On the other hand, EMD produces results aligned with the semantic content of the images, leading to a more robust estimation of differences between two segmentation results, and ultimately between the corresponding source images. Since the proposed approach does not make particular assumptions on raw images, it can be easily employed with different data sources, like ESA's Sentinel-2 or Gaofen-2 products. The caveat is that the model must show acceptable performance on the whole range of possible inputs.

5.2.1 Deep approach with pre-trained DINO-ViT

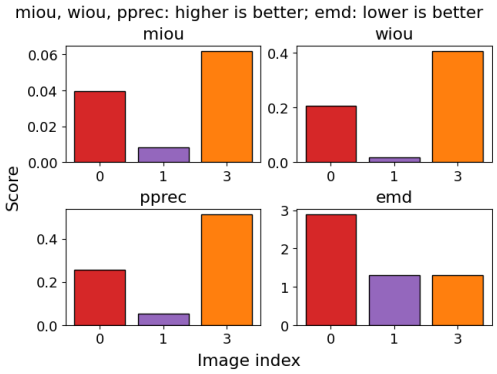
DINO (Self-**di**stillation with **no** labels) [5] is a self-supervised training methodology that builds on BYOL [19]. It produces features that can be employed effectively for image retrieval and copy detection, and has been successfully employed with both Vision Transformers and ConvNets. In this work, we evaluated the capabilities of retrieval of four different flavours of ViTs pre-trained with DINO on ImageNet: ViT-S/8, ViT-S16, ViT-B/8 and ViT-B/16, where "S" and "B" stand for "small" and "base" respectively (21M, 85M parameters), while /N represents the tokenization size of images ($N \times N$). Our baseline are the retrieval result of the EMD approach described earlier. We selected the L2 distance as a similarity measure between embeddings. Two retrieval databases composed of 300 224×224 crops and 10 query images (of the same shape) were created for this test; one is sourced from Gaofen-2 images of our test set, the other from Sentinel-2 images. Query images were selected to include instances of cities, farmland, suburbs, water bodies and rural areas. The database images are instead randomly cropped from larger images.

Table 3 shows how these transformers rank images with respect to EMD. TopN scores measure if the first pick from EMD is present in the first N picks of the transformer; TopN@M scores measure the accuracy with which the first N picks from EMD are found in the top M ranked from the transformer (by counting the total number of matches). Regarding GID images, the various configurations exhibit comparable results: S/16 and B/8 seem to achieve the best results, with a slight advantage for the latter; on the other hand, performance on the ESA task is poor across all configurations.

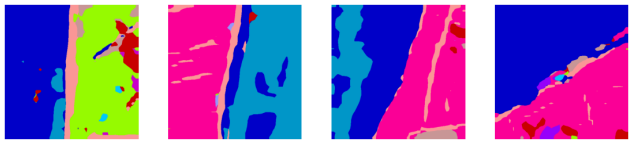
We argue that one of the main reasons behind these results is the fact that these models were trained on dataset (ImageNet) which does not contain typical remote sensing examples, thus the learned representations are only partially transferable to this task. When visual cues are sufficient to bring similar images close to each other in feature space, retrieved results are acceptable[1] (see 19a and 19c); when images are noisy, do not present clear distinctive characteristics, or what should be perceived as semantically close presents relevant visual differences between images, we observe failures in retrieving meaningful images (like in 19b and 19d). Lower resolution appears to be another factor that strongly impacts performance, as is clearly highlighted by results on Sentinel-2 images in Table 3.

DINO relies on data augmentations, in particular the "multi-crop" approach, to train models able to produce features that are invariant to changes in view and that are somewhat able to draw connections between different parts of the same image. While said invariance might be desirable when images have a single class of interest (for example most training images used for classification), it may not be beneficial when images contain multiple areas of interest (as is the case for a "dense" task like segmentation), since in this case happen that different crops might have little in common. Another observation is that ground truth information permits to draw connections between instances that are visually different; by not considering this training signal, models trained with self-supervision alone might not be able to incorporate this kind of invariance, thus leading to unsatisfactory results when we expect results that need to be robust to this type of noise. Semi-supervised approaches

[1] This statement finds some backing if we consider the fact that by rotating the query image in 19a by 180 deg, we obtain the exact same results.



(a) Computed scores



(b) Segmentation masks

(c) Source images

Figure 12. Possible pitfalls of spatial scores, visualized on four Gaofen-2 patches. Images are indexed from left to right, starting at 0. All scores are computed taking image "2" as reference.

may be better suited for such a task.

In conclusion, an emerging property of ViTs trained with DINO is the presence of semantic segmentation information in the heads of the last layer self-attention maps, when the `[CLS]` token is used as the attention's query. We visualize these maps in Figure 20 for query images and the best match of the ViT row found in Figure 19.

5.2.2 MobileNet backbone features for retrieval

We decided to evaluate the characteristics of our best model in terms of retrieval using feature planes extracted from its backbone. We obtained a vector representation by applying a max operation on the full $\frac{H}{32} \times \frac{W}{32}$ planes, which results in a 960-dimensional tensor. Similarity is computed using the L_1-distance. We also tried with the L_2-distance, but obtained slightly worse results. Table 3 shows the obtained results: this method achieves results closer to EMD than those of DINO-ViTs, but we can still observe significant discrepancies with our baseline method. In Figure 19 we visualize results obtained with this approach.

We tried different methods of obtaining a feature vector, namely performing the max operation across channels (which results in a $\frac{H}{32} \times \frac{W}{32}$ activation map), and performing a spatial max-pooling operation to obtain a 2×2 map followed by a flattening. We used the L_2 distance with both approaches, and did not achieve better results. We note that since the MobileNet backbone outputs strided feature maps, these directly depend on spatial characteristics of the input images, and thus may potentially suffer the same pitfalls presented earlier in this section.

5.3. Ablation study

In this chapter we evaluate the contribution of the various training procedures that we employed. In particular, we investigate EBB in section 5.3.1 and the custom loss function (where unlabeled pixels are excluded) in section 5.3.2. In section 5.3.3 we explain how to reduce per-class IoU variance to achieve better scores for minority classes by using a weighted loss function.

5.3.1 Reducing the dataset

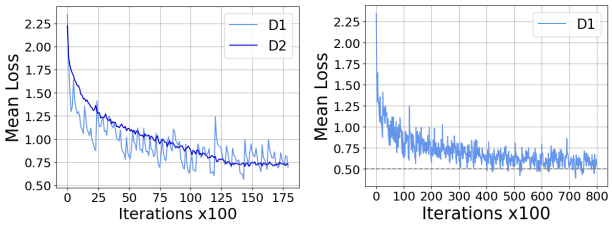


Figure 13. The left panel displays the mean training loss every 100 iterations for both D1 and D2. The right panel presents the training loss of D1 measured throughout the entire training process.

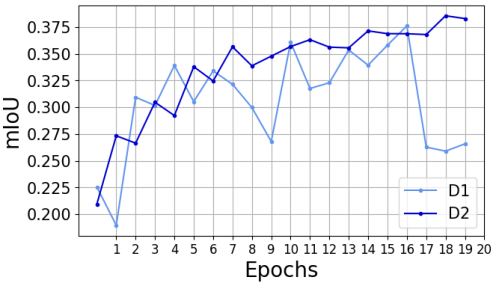


Figure 14. mIoU of both D1 and D2 for each epoch of training.

To evaluate the impact of the EBB strategy we train U-Net under the same conditions varying only the training

Table 3. Performance on retrieval task by deep networks with respect to our handcrafted score based on EMD

Model	Gaofen-2						
	Top1	Top5	Top10	Top5@5	Top5@10	Top5@20	Top5@50
DINO-ViT-S/8	0.1	0.1	0.2	0.1	0.16	0.26	0.48
DINO-ViT-S/16	0.1	0.1	0.1	0.08	0.179	0.239	0.52
DINO-ViT-B/8	0.1	0.1	0.2	0.08	0.14	0.24	0.54
DINO-ViT-B/16	0.1	0.1	0.1	0.06	0.08	0.159	0.459
MobileNet (L1)	0.2	0.5	0.7	0.16	0.36	0.46	0.58
	Sentinel-2						
DINO-ViT-S/8	0.0	0.0	0.0	0.04	0.04	0.04	0.26
DINO-ViT-S/16	0.0	0.0	0.0	0.02	0.04	0.08	0.26
DINO-ViT-B/8	0.0	0.0	0.0	0.02	0.04	0.08	0.26
DINO-ViT-B/16	0.0	0.0	0.0	0.02	0.06	0.08	0.26
MobileNet (L1)	0.1	0.1	0.2	0.06	0.1	0.22	0.44

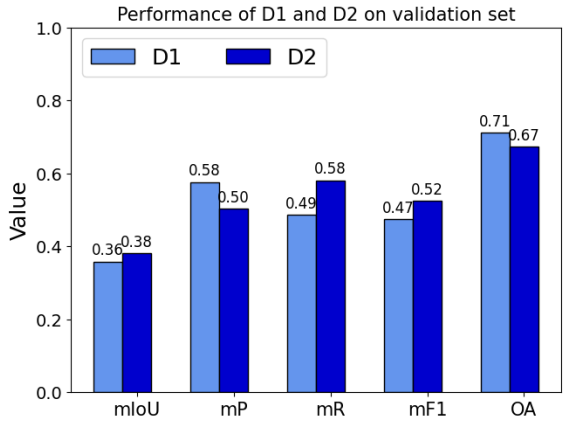


Figure 15. mIoU, mPrecision, mRecall, mF1 score and OA for both D1 and D2 on validation set.

dataset. We call D1 the model trained on the full dataset, while D2 is the model trained on the reduced dataset. Both models have been trained for 18K iterations using a batch size of 10, cross entropy as the loss function and SGD1 optimizer (lr=0.006, momentum=0.9, weight decay=0.00001, PolynomialLR scheduler). Figure 13 shows that the training loss relative to D2 is much more stable, while the model likely struggles to handle the large amount of unbalanced data in the original dataset. To further prove our point, we trained D1 for 62K more iterations to see if the training loss could reach a plateau. As shown in Figure 13 the curve does not consistently drop below 0.5. Figure 14 shows the trend of mIoU on the validation set over the epochs for both models. The curve corresponding to D1 is much more unstable, and its peak is lower compared to the peak of the curve for D2. Figure 15 shows the performance of D1 and D2 on the validation set using all evaluation metrics (we loaded the model checkpoint after epoch 16 for D2, as it holds the best performance). D1 is considerably more precise then D2, while D2 shows a better recall. In general the scores obtained are comparable, demonstrating that the model achieved excellent performance by training on approximately 11% of the data.

5.3.2 Excluding unlabeled pixels

We also assessed the impact of excluding the background by training two models: the first one (B1) employs a standard cross entropy loss, while for the second one (B2) the background is ignored and does not contribute to the input gradient. We chose the U-Net baseline for both models. For B1 he mIoU score inherently excludes the index 0 since the model never predicts it. This would result in a IoU of zero for this class, which we excluded from the summation (4.2). For B2 we calculated the mIoU by zeroing out the first row of the confusion matrix, which corresponds to the True Positives and False Negatives for the background class. This way the two models will be evaluated on the same amount of classified pixels. We trained both models for 50 epochs on the reduced dataset, using SGD1. Figure 16 illustrates the trend of loss and precision over the course of training. It can be observed that B2 is able to learn more valuable information without considering unlabeled regions, and it performs better on the validation set. We also report all of the metrics measured using the last checkpoint of both models in Figure 17. B2 outperforms B1 across all evaluation metrics, demonstrating an improvement of up to 8% in overall accuracy (OA).

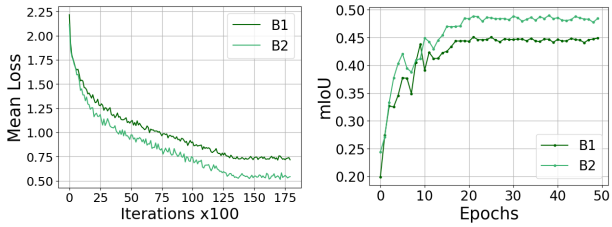


Figure 16. Comparison of mean training loss and mIoU over time: the left panel displays the mean training loss every 100 iterations for models B1 and B2, while the right panel presents the mIoU achieved by both models after each epoch.

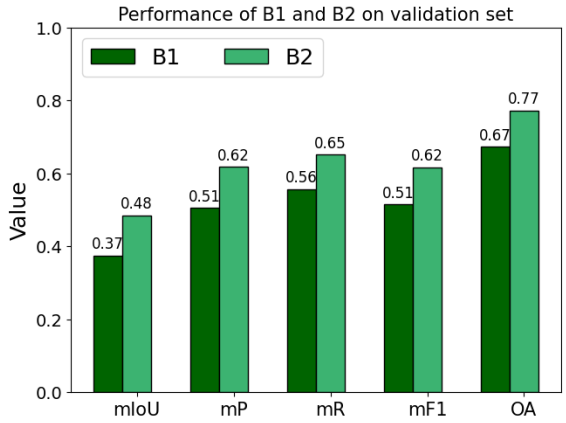


Figure 17. mIoU, mPrecision, mRecall, mF1 score and OA for both B1 and B2 on validation set.

5.3.3 Using a weighted loss function

To further mitigate class imbalance, we experimented with using a weighted loss function. We use our standard Cross Entropy Loss, with each weight being inversely proportional to the frequency of its class in the training set. So for each class i, w_i is computed as follows:

$$w_i = \frac{\text{Total Pixels}}{2 \times \text{Pixels of class } i}$$

We trained 2 models based on our baseline U-Net with the same hyperparameters and dataset presented in 5: the first employed the weighted loss while the second the standard cross entropy loss. We refer to the first as *wUn* and to the latter as *Un*. As figure 18 shows, the spread of IoUs values is slightly narrower for *wUn*, showing the effectivness of the weighted loss, even if it's a small improvement. Moreover, MobileNet achieved the best IoUs with a boxplot that exhibits a more compact distribution, with the quartiles being relatively symmetrical. *wUn* achieves similar performance with respect to *Un* on the validation set, with a mIoU of 0.44 and a OA of 0.69. This is probably due to the fact that our validation set has a class distribution that is skewed

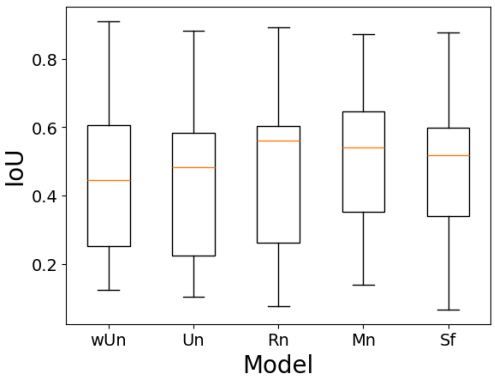


Figure 18. Boxplot showing the spread of IoU values for each of our main models: Un(U-Net with standard Cross Entropy loss), wUn(U-Net with weighted cross entropy loss), Resnet(Rn), MobileNet(Mn) and Segformer(Sf).

towards some classes (like irrigated cropland or urban residential). A further investigation can be done by selecting another (more balanced) validation set. One might consider applying EBB to the validation set as well and comparing the performance of the weighted model against the standard model.

6. Conclusion

In this work we studied the impact of various training practices on the performance of semantic segmentation architectures, in particular when dealing with an heavily imbalanced datasets. Our experiments showed that it is possible to achieve better results only a small balanced subset of the total available pixels in the dataset. In particular, we built this subset by iteratively selecting constant-size tiles taken from the dataset's full size images that maximize the cumulative distribution of classes in the set, creating what we call *Entropy Based Balancing*. We then focused on excluding the background from the loss function and weighting each class based on its frequency in the training set, which led to yet some small improvements. On top of these, we added data augmentations such as blurring (only on source images) and rotations (both on images and ground truth masks).

We then proposed a simple retrieval mechanism that uses class distributions of segmented images, based on the EMD and with class-ordering, to compute a similarity score. We showed that this method is able to retrieve plausible results, while being applicable to multiple sources of satellite imagery. By comparison, we also tried using distances between embeddings of DINO-ViTs and of the MobileNet backbone. Even though both approaches where not at the same level as our handcrafted method, we believe that it would be still worth investigating models trained with

DINO self-supervision as components of segmentation architectures, since this method can leverage the huge amount of unlabeled remote sensing datasets publicly available.

7. Contributions

P.M: retrieval, visualization, model analysis. **M.L**: development, evaluation, data analysis. **N.M**: experimental work, model training, data analysis. We thank Davide Caffagni for his precious insights regarding retrieval and good training practices.

References

[1] Mateusz Buda, Ashirbani Saha, and Maciej A Mazurowski. Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm. *Computers in Biology and Medicine*, 109, 2019. 2

[2] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context, 2018. 1

[3] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. In *European conference on computer vision*, pages 205–218. Springer, 2022. 3

[4] Yong Cao, Chunlei Huo, Shiming Xiang, and Chunhong Pan. Gffnet: Global feature fusion network for semantic segmentation of large-scale remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2024. 3

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021. 2, 10

[6] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L. Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation, 2021. 3

[7] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling, 2021. 2

[8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, 2017. 2

[9] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 5

[10] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 2

[11] Ilke Demir, Krzysztof Koperski, David Lindenbaum, Guan Pang, Jing Huang, Saikat Basu, Forest Hughes, Devis Tuia, and Ramesh Raskar. Deepglobe 2018: A challenge to parse the earth through satellite images. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 172–181, 2018. 1

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 1

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. 2

[14] 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. 3

[15] Alexey DOSOVITSKIY. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[16] ESA. Copernicus project. https://www.esa.int/Applications/Observing_the_Earth/Copernicus, 2024. Accessed: 2024-08-31. 1

[17] ESA. World cover. https://esa-worldcover.org/en, 2024. Accessed: 2024-08-31. 1

[18] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[19] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020. 10

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 2

[21] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 204–207. IEEE, 2018. 1

[22] Le Hou, Chen-Ping Yu, and Dimitris Samaras. Squared earth mover's distance-based loss for training deep neural networks, 2017. 9

[23] Zhehao Hu, Yurong Qian, ZhengQing Xiao, Guangqi Yang, Hao Jiang, and Xiao Sun. Sabnet: Self-attention bilateral network for land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2024. 4

[24] Huimin Huang, Lanfen Lin, Ruofeng Tong, Hongjie Hu, Qiaowei Zhang, Yutaro Iwamoto, Xianhua Han, Yen-Wei

Chen, and Jian Wu. Unet 3+: A full-scale connected unet for medical image segmentation, 2020. 2

[25] HuggingFace. Huggingface webpage. https://huggingface.co/, 2024. Accessed: 2024-08-31. 6

[26] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. 2

[27] Jionghui Jiang, Xi'an Feng, and Hui Huang. Semantic segmentation of remote sensing images based on dual-channel attention mechanism. *IET Image Processing*, 2024. 3, 4, 8

[28] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 7

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 2

[30] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 2

[31] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation, 2015. 2

[32] Pietro Moriello Matteo Lugli, Nicola Morelli. Remote sensing semantic segmentation framework. https://github.com/theElandor/CVCS, 2024. Accessed: 2024-08-31. 7

[33] Philip Popien. Sen1floods11. https://github.com/cloudtostreet/Sen1Floods11, 2024. Accessed: 2024-08-31. 1

[34] Pytorch. Mobilenet-deeplabv3. https://pytorch.org/vision/stable/models/generated/torchvision.models.segmentation.deeplabv3_mobilenet_v3_large.html#torchvision.models.segmentation.deeplabv3_mobilenet_v3_large, 2024. Accessed: 2024-08-31. 6, 8, 9

[35] Pytorch. Polynomiallr. https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.PolynomialLR.html, 2024. Accessed: 2024-08-31. 7

[36] Pytorch. Resnet101-deeplabv3. https://pytorch.org/vision/stable/models/generated/torchvision.models.segmentation.deeplabv3_resnet101.html#torchvision.models.segmentation.deeplabv3_resnet101, 2024. Accessed: 2024-08-31. 6, 8, 9

[37] Niels Rogge. Segformer. https://github.com/NVlabs/SegFormer, 2024. Accessed: 2024-08-31. 6, 8, 9

[38] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 2, 5, 7, 8, 9

[39] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vision*, 40(2):99–121, nov 2000. 9

[40] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016. 7

[41] Bo Si, Zhennan Wang, Zhoulu Yu, and Ke Wang. Abnet: An aggregated backbone network architecture for fine landcover classification. *Remote Sensing*, 16(10):1725, 2024. 8

[42] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. 2

[43] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014. 2

[44] Muhammad Talha, Farrukh A Bhatti, Sajid Ghuffar, and Hamza Zafar. Adu-net: semantic segmentation of satellite imagery for land cover classification. *Advances in Space Research*, 72(5):1780–1788, 2023. 3

[45] Xin-Yi Tong, Gui-Song Xia, Qikai Lu, Huangfeng 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, doi: 10.1016/j.rse.2019.111322*, 2020. 3

[46] 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. 3, 9

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 2

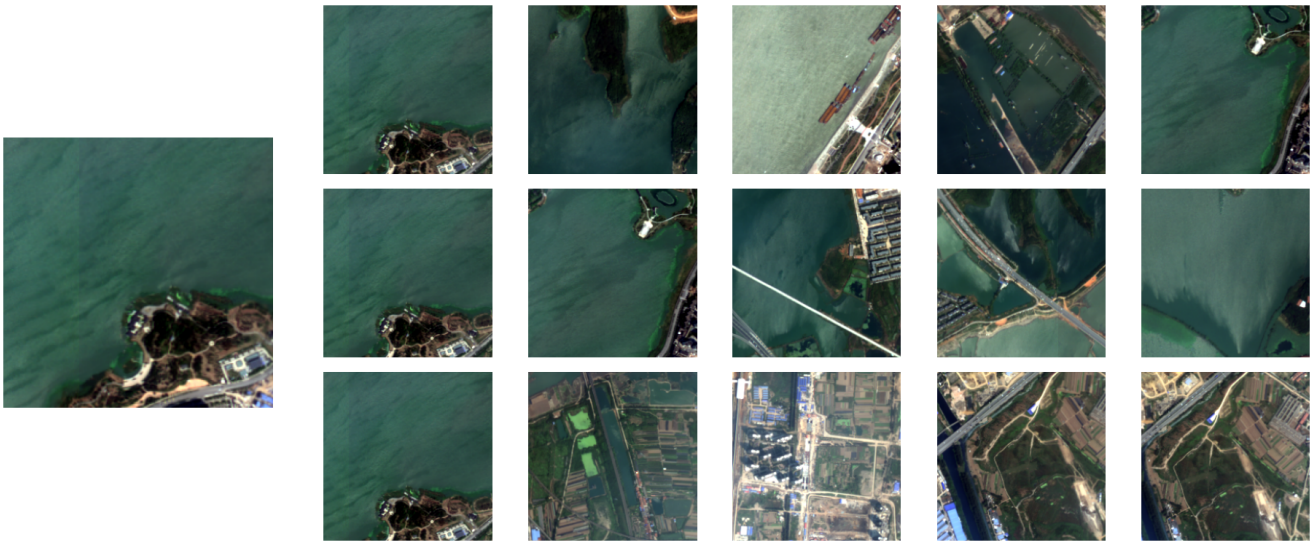
[48] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers, 2021. 2

[49] Haodong Yang, Xinyue Kang, Long Liu, Yujiang Liu, and Zhongling Huang. Sar-hub: Pre-training, fine-tuning, and explaining. *Remote Sensing*, 15(23):5534, 2023. 1

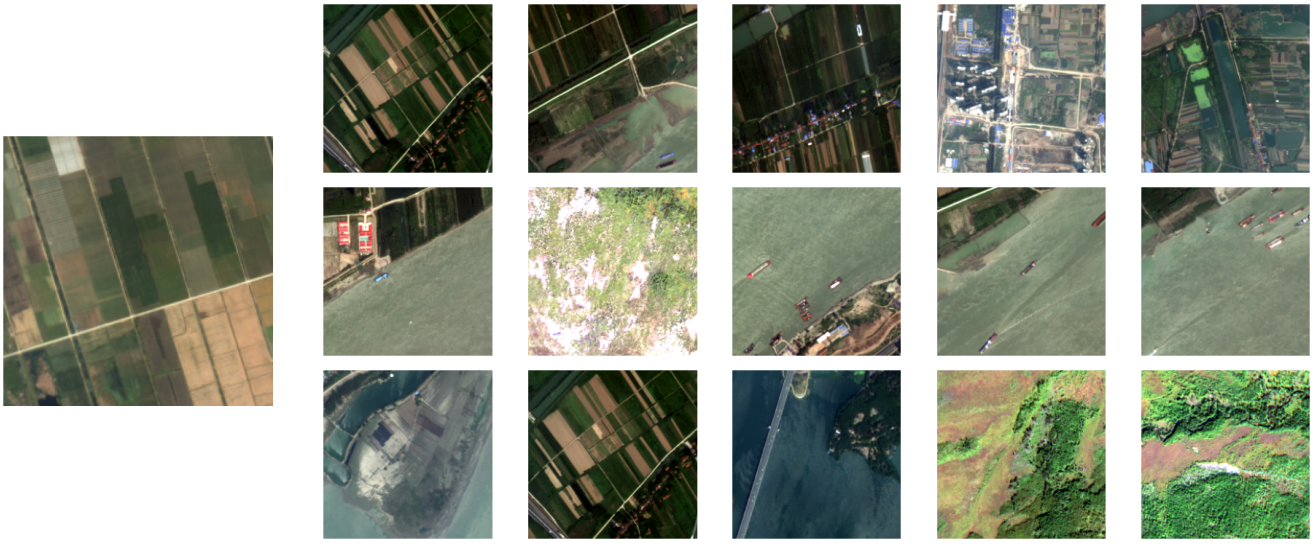
[50] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network, 2017. 2

[51] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset, 2018. 1

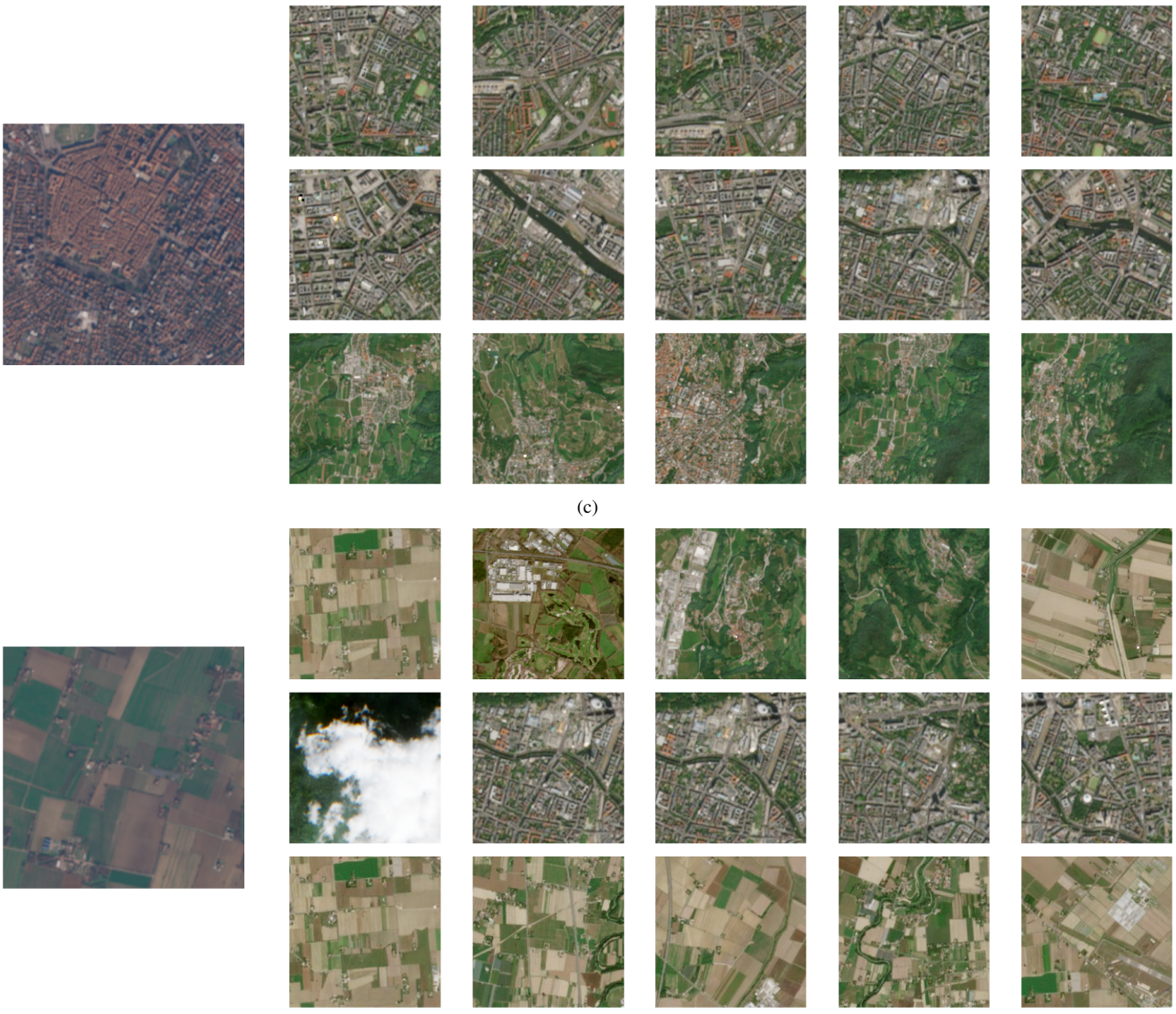
[52] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation, 2018. 2



(a)



(b)



(c)

(d)

Figure 19. Examples of retrieval, featuring both Gaofen-2 ((a), (b)) and Sentinel-2 ((c), (d)) queries (on the left of each composition). Each row presents the best 5 results (sorted from left to right) obtained using three different methods: top row for EMD score; middle row L_2 distance between DINO-ViTS/16 embeddings; bottom row L_2 distance between embeddings derived from the MobileNet backbone features.

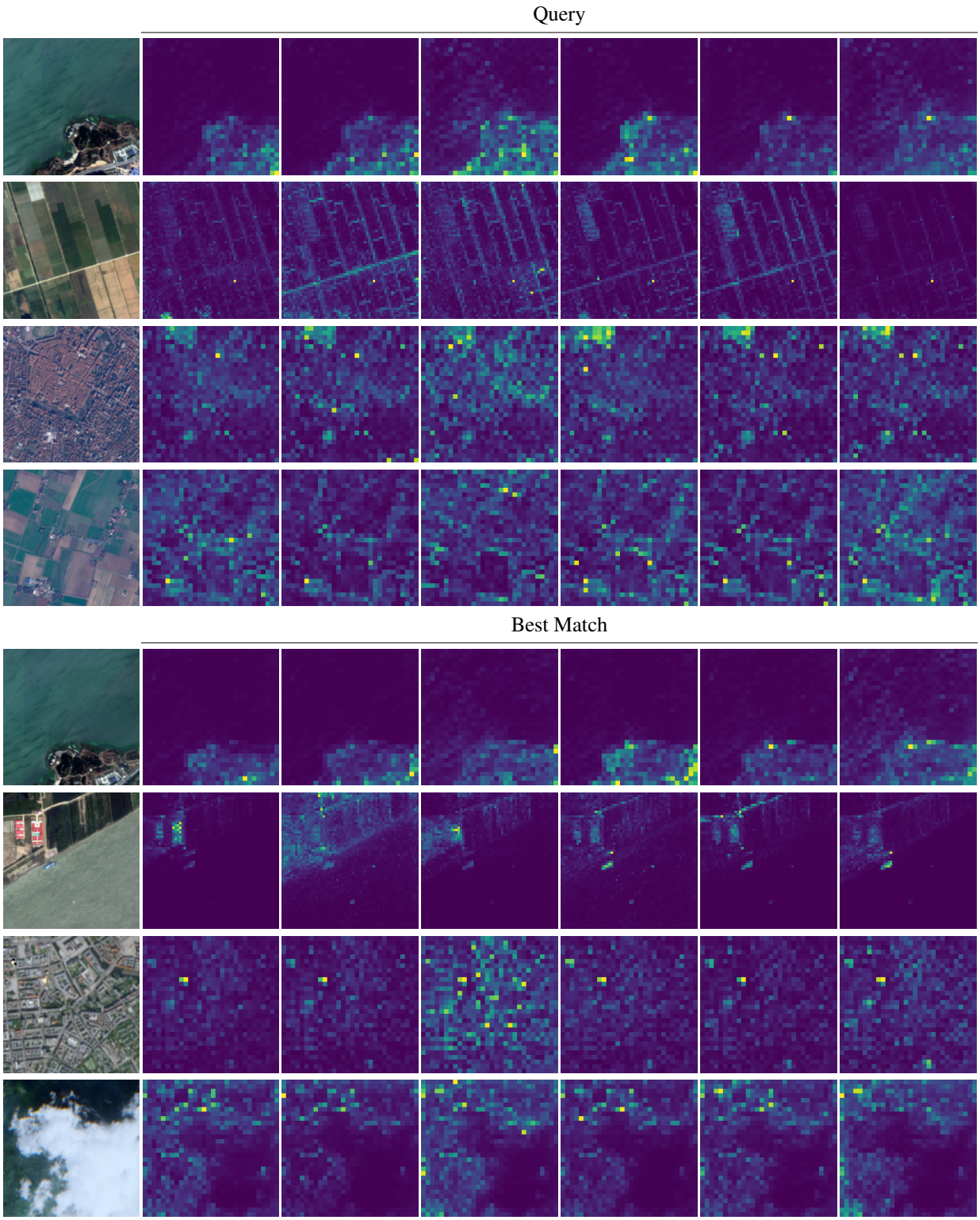


Figure 20. Self-attention heads of the last layer of a ViT-S/16. Each map shows the activation of the `[CLS]` token when used as a attention query for the transformer's heads. We compare activations between query images and the best match returned.