Supplementary Material
OpenEarthMap:
A Benchmark Dataset for Global High-Resolution Land Cover Mapping

Junshi Xia[1,*], Naoto Yokoya[2,1,*,†], Bruno Adriano[1,*], and Clifford Broni-Bediako[1]

[1]RIKEN AIP, Japan {junshi.xia,bruno.adriano,clifford.broni-bediako}@riken.jp

[2]The University of Tokyo, Japan yokoya@k.u-tokyo.ac.jp

In this supplementary, we provide a detailed description of the land cover classes of the OpenEarthMap dataset. We also present a brief overview of the baseline methods that were experimented on the OpenEarthMap dataset for the semantic segmentation and unsupervised domain adaptation tasks. The training settings and more experimental results on the semantic segmentation and the unsupervised domain adaptation tasks are presented. Furthermore, we provide more land cover mapping results that were created from out-of-sample images (i.e., images not included in OpenEarthMap) to further demonstrate the generalization of the OpenEarthMap feature space. The attribution of all source data is summarized at the end.

1. Land Cover Classes

Using the Anderson classification [1] as a starting point, we subdivided the *urban* class into three classes: *building*, *road*, and *developed space*, which are visually interpretable in high-resolution images at a sub-meter level of ground sampling distance (GSD). The definitions of 8 classes in OpenEarthMap are summarized as follows.

- **Bareland** includes natural areas covered by sand or rocks without vegetation, and other accumulations of earthen materials.
- **Rangeland** includes areas dominated by herbaceous vegetation or bushes that are not cultivated or grazed, as well as grass and shrubs in gardens, parks, and golf courses.
- **Developed space** includes areas such as sidewalks, pavements, footpaths, parking lots, and construction sites as well as artificial grass areas search as tennis courts, baseball and football fields, etc. A lane in-between parking lots is considered as a road. The materials include asphalt, concrete, stones, bricks, and tiles. Compacted soil is also labeled as developed space.

*Equal contribution. †Corresponding author.

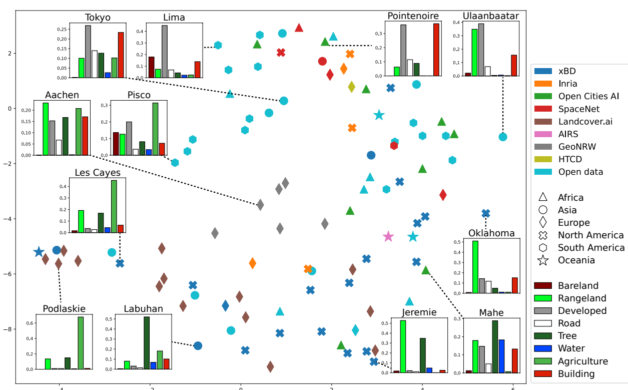

Figure 1: t-SNE 2D visualization of the 97 regions based on the class proportions in the OpenEarthMap dataset.

- **Road** includes lanes, streets, railways, airport runways, and highway/motorway for vehicles (e.g., trucks, cars, motorbikes, trains, and airplanes) excluding bicycles. The materials of roads include asphalt, concrete, and soil.
- **Tree** includes individual trees and a group of trees that are identified from their shapes (shadow) and height.
- **Water** includes water bodies (e.g., rivers, streams, lakes, sea, ponds, dams) and swimming pools.
- **Agriculture land** includes areas used for producing crops (e.g., rice, wheat corn, soybeans, vegetables, tobacco, and cotton), perennial woody crops (e.g., orchards and vineyards), and non-native vegetation for grazing.
- **Building** includes residential, commercial, and industrial buildings.

Figure 1 shows a t-SNE 2D plot of the 97 regions using class proportions in the OpenEarthMap dataset. As can be seen in the bar graphs of 12 representative regions, the class proportions in the different regions are diverse.

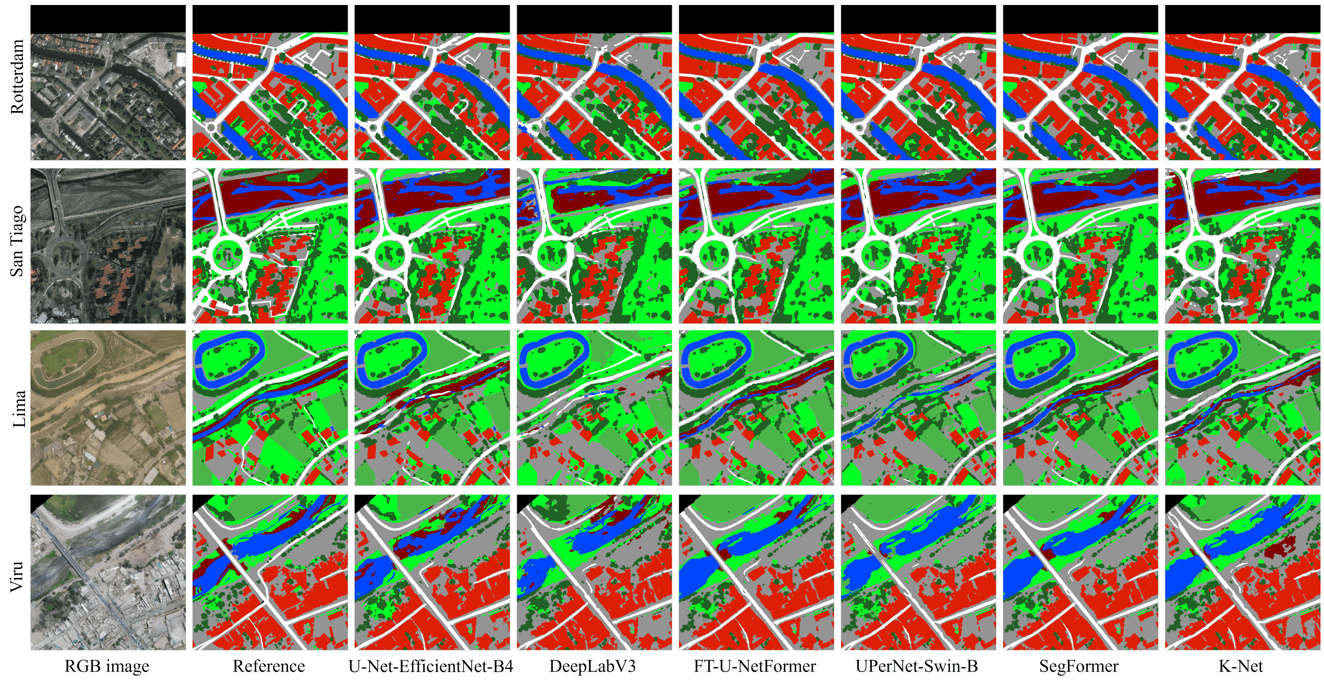

Figure 2: Visual comparison of land cover mapping results of some of the baseline models.

Table 1: Performance of ImageNet pre-trained model with different optimizers.

Model	Pre-trained on	Optimizer	mIoU
UPerNet-Swin-B	ImageNet	SGD	62.15
	ImageNet	AadmW	66.09
	None	SGD	62.32
	None	AadmW	**66.13**

Table 2: The results of using different input patch sizes and different loss functions with UPerNet-Swin-B.

(a) Input patch size

Patch size	mIoU
512×512	66.09
620×620	66.14
768×768	66.17
1024×1024	**67.02**

(b) Loss functions

Function	mIoU
CE	66.09
CE+Lovasz	66.21
CE+Focal	65.89
CE+Lovasz+Focal	**66.38**

2. Land Cover Semantic Segmentation

2.1. Brief Overview of the Baselines

The land cover semantic segmentation baseline networks that were experimented on the OpenEarthMap dataset are CNN-based (U-Net [16], DeepLabV3 [3], HRNet [17], K-Net [28], and ConvNeXt [12]) and Transformer-based (U-NetFormer [22], FT-U-NetFormer [22], SETR [29], SegFormer [26], UPerNet [25] with the backbones of ViT [7], Twins [5], and Swin Transformer [11]) architectures. U-Net uses an encoder-decoder structure to extract objects and image context at different scales. U-Net models with VGG-11, ResNet-34, and EfficientNet-B4 as backbones were adopted. DeepLabV3 [3] uses a dilation hyperparameter of convolutional layers to develop atrous spatial pyramid pooling for robust object segmentation through many scales. HRNet builds high-resolution representations by continually executing multi-scale fusions across parallel convolutions. We employed an HRNet with W48 as a backbone. K-Net [28] separates instances and semantic categories uniformly with a number of learnable kernels. The kernels conduct convolution on the image features to provide segmentation predictions. ConvNeXt [12] is a pure ConvNet model constructed entirely from standard ConvNet modules. UPerNet with a backbone of ConvNeXt-B was used. The U-NetFormer [22] selects the advanced ResNext101 as the encoder and develops an efficient global–local attention mechanism to model both global and local information in the decoder. FT-U-NetFormer [22] replaces the CNN encoder with the Swin Transformer (Swin-B). SETR [29] interprets an input image as a sequence of patches represented by a learned patch embedding, then, modifies the sequence using a global self-attention module for discriminative feature representation learning. A SETR PUP with a back-

Table 3: Compact segmentation models discovered on OpenEarthMap training set with SparseMask and FasterSeg. The class IoUs and the mIoU are calculated on the test set of OpenEarthMap with TTA applied.

Method	Trial	IoU (%)								mIoU	Params	FLOPs	FPS
		Bareland	Rangeland	Developed	Road	Tree	Water	Agriculture	Building	(%)	(M)	(G)	(ms)
SparseMask	1st	**47.00**	**53.42**	46.40	46.94	66.98	**79.02**	72.64	69.25	**60.21**	2.96	**10.28**	25.9
	2nd	45.96	53.01	**46.71**	47.26	67.12	78.81	**71.88**	69.27	60.00	3.10	10.39	26.4
FasterSeg	1st	34.04	51.40	44.97	55.82	66.58	74.50	70.33	69.14	58.35	**2.23**	14.58	74.8
	2nd	35.78	52.03	46.32	**56.97**	**67.20**	75.76	70.70	**70.55**	59.41	3.47	15.37	**89.5**

Table 4: Compact segmentation models discovered on OpenEarthMap training set with SparseMask and FasterSeg. The class IoUs and the mIoU are calculated on the test set of OpenEarthMap without TTA applied.

Method	Trial	IoU (%)								mIoU	Params	FLOPs	FPS
		Bareland	Rangeland	Developed	Road	Tree	Water	Agriculture	Building	(%)	(M)	(G)	(ms)
SparseMask	1st	**46.15**	**51.88**	44.01	43.64	65.20	**77.41**	**71.47**	66.10	58.23	2.96	**10.28**	51.2
	2nd	44.78	51.56	44.34	43.95	65.44	77.34	70.98	66.09	58.06	3.10	10.39	57.2
FasterSeg	1st	33.52	50.60	43.93	54.74	65.98	73.55	69.73	68.36	57.55	**2.23**	14.58	143.2
	2nd	34.50	51.27	**45.27**	**55.94**	**66.61**	74.71	70.05	**69.73**	**58.51**	3.47	15.37	**171.3**

bone of ViT-L was used. SegFormer [26] unifies transformers with lightweight multilayer perceptron decoders without considering positional encoding. A MiT-B5 encoder was used as a backbone for SegFormer. Swin Transformer [11] creates hierarchical feature maps by merging image patches into deeper layers. Twins [5] uses a spatially separable attention mechanism comprising of locally-grouped self-attention and global sub-sampled attention.

2.2. Experimental Details

All the baselines we used for the experiments are PyTorch-based. The U-Net-based architectures are adopted from Yakubovskiy [27] and Wang *et al.* [22], and the other architectures are from MMsegmentation [6]. The networks were trained on a single NVIDIA GPU DGX-1/DGX-2 with 16/32GB of RAM. The number of epochs was set to 200, and a batch size of 8 with an image input size of 512×512 randomly cropped was employed. The cross-entropy (CE) loss was used in training all the networks. For the U-Net-based architectures, we used AdamW optimizer [13] with a learning rate of 1×10^{-4} and weight decay of 1×10^{-6}. For the MMsegmentation-based architectures, we used the default settings of each method. We adopted stochastic gradient descent (SGD) optimizer with a learning rate of 1×10^{-3}, weight decay of 5×10^{-4}, and momentum of 0.9 for the DeeplabV3 and HRNet networks. The rest of the networks used AdamW optimizer with a learning rate set as 6×10^{-5}, weight decay as 0.01, and betas parameters as 0.9 and 0.999. A polynomial learning rate decay with a factor of 1.0 and an initial linear warm-up of 1500 iterations was used. The backbones in all the networks were pre-trained on the ImageNet dataset. No data augmentation was applied during training for all networks. Following previous works, we used mIoU to assess the performance of all models. All results are based on test-time augmentation (TTA) with flipping operations.

2.3. Results

Visualization: More visual examples of segmentation results obtained from some selected baseline methods are presented in Figure 2. In the first row (Rotterdam), DeepLabV3 failed to identify the entire stretch of *road*. In the second row (San Tiago), the entire *bareland* and the small *developed space* on top of the *bareland* were identified by FT-U-NetFormer and SegFormer. U-Net-EfficientNet-B4, FT-U-NetFormer and SegFormer were able to identify the *bareland* that stretches from along the river (third row), but DeepLabV3, UPerNet-Swin-B, and K-Net failed to identify it. Compared to DeepLabV3 and K-Net, U-Net-EfficientNet-B4, DeepLabV3, and FT-U-NetFormer did recognize most parts of the *water* body in Viru (fourth row). UPerNet-Swin-B and DeeplabV3 classified the *water* body in Viru as *agricultural land*, *rangeland*, and *developed space*.

Ablation study: We conducted an ablation study to examine the effects of ImageNet pre-training, optimizers, image size, and loss functions on the OpenEarthMap dataset using a UPerNet-Swin-B network. We employed two different optimizers (AadmW and SGD), three loss functions (CE, Lovasz, and Focal), and four different patch sizes. As shown in Table 1, the model that was not pre-trained on ImageNet performed better than the one pre-trained on ImageNet. In both cases, AdamW optimizer attains better results. Table 2 shows the results of using different patch sizes and different loss functions for training UPerNet-Swin-B. A larger patch size tends to achieve better performance. The combination of all three loss functions (CE, Lovasz, and Focal) can improve the performance of UPerNet-Swin-B.

2.4. Neural Architecture Search Methods

The two automated neural architecture search methods, SparseMask [24] and FasterSeg [4], which we adopted were particularly proposed for compact architecture search for semantic segmentation tasks in computer vision. Both methods employed a gradient-based search strategy similar to the one used in DARTS [10]. Whereas SparseMask used a pruning technique to compress the searched architectures, teacher-student co-searching (knowledge distillation technique) was used in FasterSeg. Following the architecture search protocol in both methods, we conducted four experiments, two with each method, by searching for lightweight architectures on the OpenEarthMap dataset. All the experiments we performed on a single NVIDIA Tesla P100 with 16GB memory. It took about 0.8 GPU days and 2 GPU days to perform the architecture search with SparseMask and FasterSeg, respectively. The architectures discovered with both methods were trained from scratch using the same training protocol adopted in [24] and [4] with the exception of setting the number of epochs to 450. We measured the class-specific IoUs and mIoU using a single-scale input with TTA (see Table 3) and without TTA (see Table 4) of flipping operations applied to the test set of the OpenEarthMap dataset. The FPS and the FLOPs were calculated with a single image of 1024×1024 pixels as an input. The inference speed was calculated using the settings in FasterSeg. Table 3 and Table 4 show the detail class-specific IoUs along with the mIoUs that are presented in the paper.

3. Unsupevised Domain Adaptation

3.1. Brief Overview of the Baselines

We adopted a metric-based method (MCD [19]), adversarial training methods (AdaptSeg [18], category-level adversarial network (CLAN) [14], TransNorm [23], and fine-grained adversarial learning framework for domain adaptive (FADA) [20]), and self-training methods (pyramid curriculum DA (PyCDA) [9], class-balanced self-training (CBST) [30], instance adaptive self-training (IAST) [15], and DAFormer [8]) for the unsupervised domain adaptation task. DAFormer is a transformer-based model, and the others are based on DeepLabV2. The adversarial training methods seek to match the distributions of the source and target domains from input-feature-output or a patch level in a generative adversarial network. AdaptSeg [18] uses adversarial learning in the output space and a multi-level adversarial network to effectively perform output space domain adaptation at different feature levels. CLAN [14] aligns the classes with an adaptive adversarial loss. TransNorm [23] uses an end-to-end trainable layer to make networks more transferable across domains. FADA [20] uses a fine-grained adversarial learning framework by aligning the class-level features. Self-learning methods generate the target domain's pseudo labels, retrain the model and repeat the procedure. PyCDA [9] observes the target properties and fuses multi-scale features. CBST [30] and IAST [15] aim at selecting balanced samples to improve the quality of pseudo labels. DAFormer [8] constructs a transformer encoder and a multilevel context-aware feature fusion decoder by adopting three effective training strategies: rare class sampling, ImageNet Feature Distance, and a learning rate warmup.

Table 5: Comparing source-only and oracle training for different networks on the 24 regions test set of OpenEarthMap.

Model	mIoU (%)	
	Source-only	Oracle
U-Net-EfficientNet-B4	**63.17**	64.09
DeepLabV2	50.01	54.65
DeepLabV3	55.27	59.83
HRNet	56.25	60.02
SegFormer	58.25	**64.76**
UPerNet-Swin-B	52.35	61.82
K-Net	57.21	64.12

3.2. Experimental Details

For the DeepLabV2-based methods, we adopted the architectures in Wang *et al.* [21] and kept the default setting. We used the following training settings (batch size of 8 with image input size of 512×512 randomly cropped) in the semantic segmentation task for all the DeepLabV2-based methods and the DAFormer. DeepLabV2 with ResNet50 was used as an extractor, and a discriminator was constructed using fully convolutional layers. The classification and the discriminator learning rates for the adversarial training methods were set to 5×10^{-3} and 10^{-4}, respectively. Adam optimizer was used in the discriminator with momentum of 0.9 and 0.99. We adopted the default pseudo-generation hyper-parameters in CBST and IAST, and set the classification learning rate to 10^{-2}. All the networks were trained for 40K steps in two stages. In the first stage, the models were trained only on the source images for 8K steps for initialization. In the second stage, the pseudo-labels were then updated every 2K steps for the remaining training process. For the DAFormer, an MiT-B5 encoder was adopted with AdamW. Other hyper-parameters remained the same as in the original literature [8].

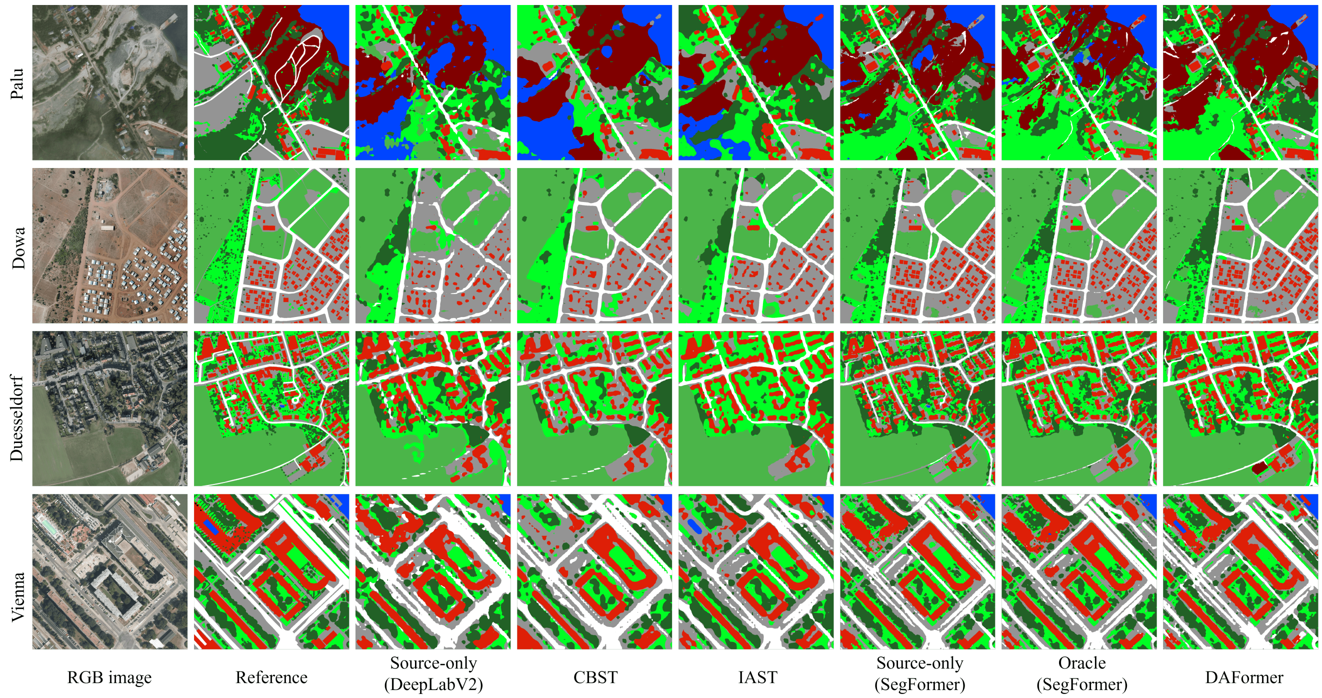

Figure 3: Visual comparison of unsupervised domain adaption results of some of the baseline models.

3.3. Results

Visualization: More visual examples of the UDA results are presented in Figure 3. In the first row (Palu), the source-only DeepLabV2 can barely identify the *tree* (bottom-left) and *developed space* (center-left). These areas were classified as *bareland* and *water*. CBST and IAST also did not perform well in these areas. Source-only SegFormer and DAFormer slightly improved their performance in these areas. In the second row (Dowa), source-only DeepLabV2 could not recognize the tiny *road* and small *buildings*. IAST and CBST performed better on the *road*, but they cloud not recognize the small *buildings*. DAFormer performs exceptionally well in those two areas. In the third row (Dusseldorf), DAFormer could identify the long and tiny *road* (bottom), and the other methods only recognized some parts of the *road*. The IAST did identify the small *water* area (top-left) in Vienna (fourth row), and the other methods classified it as *building*.

Comparison of network architectures: To further evaluate the suitability of the networks for the UDA task, we performed several experiments with source-only and oracle training for different models and provided their mIoU results in Table. 5. The classical DeepLabV2 yielded the worst results in both source-only and oracle. UPerNet-Swin-B was slightly better than DeepLabV2 in the source-only setting. SegFormer obtained the best oracle performance, and U-Net-EfficientNet-B4 outperformed SegFormer on the source-only setting. Generally, U-Net-EfficientNet-B4, SegFormer, and K-Net shared the top positions in both source-only and oracle settings. These networks are recommended to be further investigated for the development of UDA methods on the OpenEarthMap dataset.

Continent-wise UDA: Visual comparison of continent-wise UDA results of source-only SegFormer, Oracle and DAFormer are shown in Figure 4. Eight combinations of source and target domains are provided. Here, we denote Africa, Asia, Europe, North America, South America, and Oceania as AF, AS, EU, NA, SA, and OC, respectively. For AF→AS, AS→EU, and AF→EU, DAFormer significantly achieved better results on *agriculture land* when compared to source-only SegFormer. DAFormer also identified the *water* in EU→NA, and the small *buildings* and the tiny *roads* in SA→AF and AF→NA. The class-specific IoUs and mIoUs obtained from source-only U-Net-EfficientNet-B4, source-only SegFormer, and DAFormer are presented in Figure 5, Figure 6, and Figure 7, respectively. For most classes, when OC is considered as the source domain, the transferred result is worst due to the limited number of images in OC. However, when OC is treated as the target domain the performance is better than other settings. With the exception of OC, *building* is the easiest transferred class, which U-Net-EfficientNet-B4 achieved IoUs range of 63.4 to 76.8 and SegFormer achieved a range of 64.3 to 76.3. The most challenging transferred class is *bareland*. The U-Net-EfficientNet-B4 and the SegFormer attained IoUs range

Figure 4: Visual comparison of continent-wise unsupervised domain adaption results of source-only SegFormer, Oracle and DAFormer. Asia: AS, Europe: EU, Africa: AF, North America: NA, South America: SA and Oceania: OC.

of 4.3 to 32.5 and 4.1 to 32.7, respectively. In the order of easiest to challenging class is *building*, *tree*, *road*, *rangeland*, *developed space*,*water*, *agriculture land*, and *bareland*. The performance change from SegFormer to DAFormer is shown in Figure 8. Red indicates an improvement whereas blue depicts a decrease in results. In Figure 8, one can clearly see that DAFormer improved the results in the challenging classes (e.g., *bareland* and *water*).

4. Mapping for Out-of-Sample Images

Figure 9 shows visual comparisons of the Chesapeake Bay land cover map with those generated by U-Net-EfficientNet-B4 models trained on OpenEarthMap, LoveDA, DeepGlobe, and DynamicEarthNet with the same implementation details. The results obtained by the OpenEarthMap model demonstrate fine spatial details and semantically consistent mapping with the Chesapeake Bay land cover map. Note that the maps of the DynamicEarthNet model were obtained with the original 1m GSD because they yielded better results than those processed with 3m GSD. See Figure 10 for a comparison of mapping results with the DynamicEarthNet model using images at different GSDs (i.e., 0.5cm, 1m, and 3m) for inference.

Furthermore, we demonstrate visual results of land cover mapping for out-of-sample images from France (MiniFrance [2]), China (LoveDA [21]), Ecuador (SIGTIERRAS[1]), and Tanzania (Zanzibar Mapping Initiative (ZMI)[2]) in Figures 11, 12, 13, and 14, respectively. These land cover maps were obtained by a U-Net-EfficientNet-B4 trained on the OpenEarthMap dataset. One can observe that the results are at a reasonably high resolution. Considering the fact that these images are not included in the OpenEarthMap dataset, the results further support the generalization capability of the OpenEarthMap model.

[1] http://www.sigtierras.gob.ec/
[2] http://www.zanzibarmapping.org/

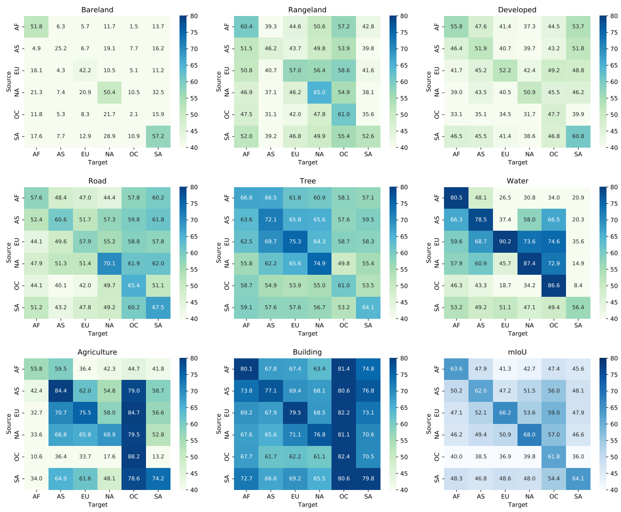

Figure 5: Continent-wise UDA results from source-only U-Net-EfficientNet-B4. Class-specific IoUs and mIoU are shown in the subfigures with oracle values in the diagonal.

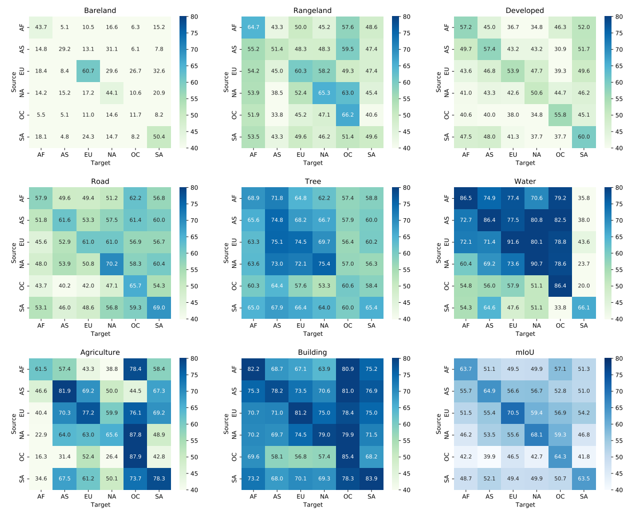

Figure 6: Continent-wise UDA results from source-only SegFormer. Class-specific IoUs and mIoU are shown in the subfigures with oracle values in the diagonal.

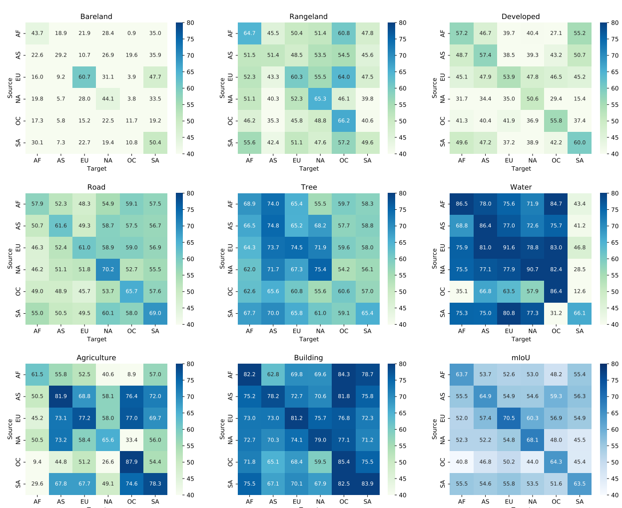

Figure 7: Performance of DAFormer in Continent-wise UDA. Class-specific IoUs and mIoU are shown in the subfigures with oracle values in the diagonal.

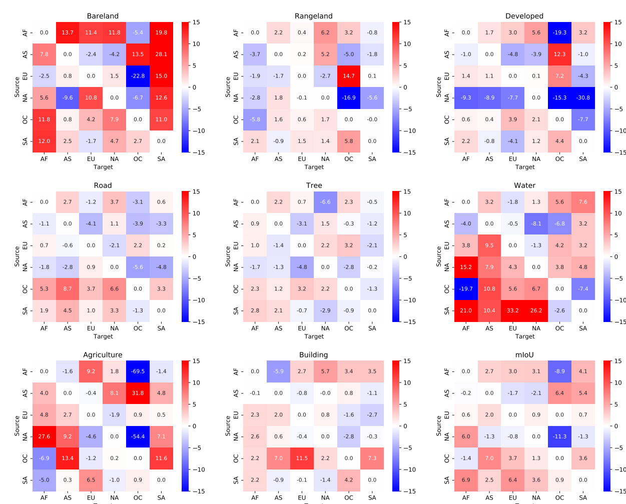

Figure 8: Performance change from SegFormer to DAFormer in Continent-wise UDA. Class-specific IoUs and mIoU differences are shown in the subfigures.

5. Attribution of Source Data

Table 6 summarizes attribution of source data for 97 regions in OpenEarthMap. Our label data are provided under the same license as the original RGB images, which varies with each source dataset. Label data for regions where the original RGB images are in the public domain or where the license is not explicitly stated are licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

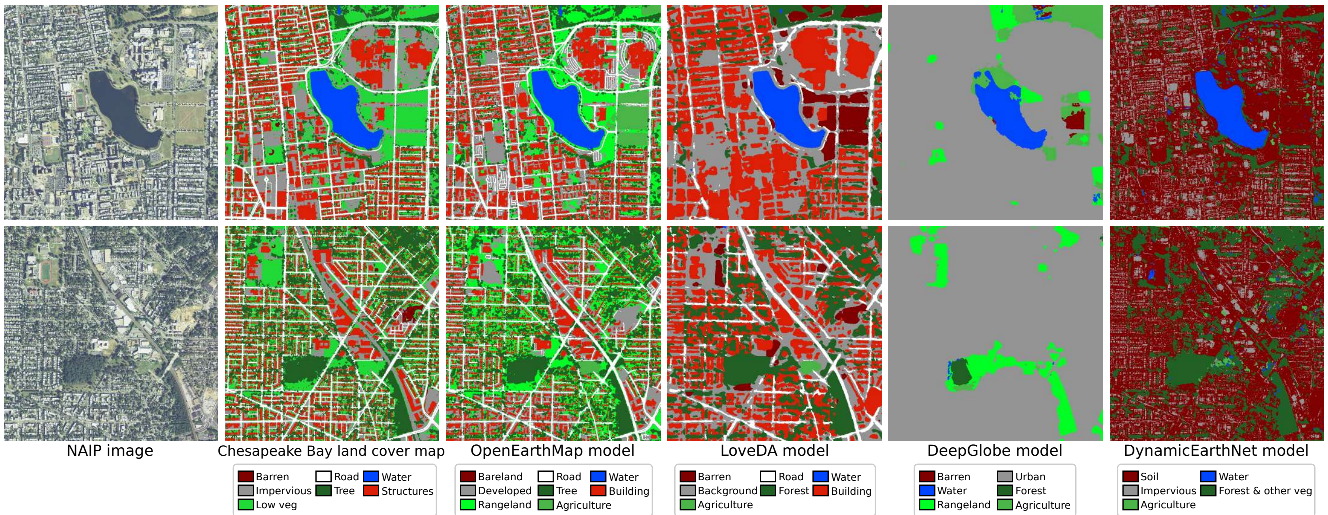

Figure 9: Visual comparison of Chesapeake Bay land cover map with land cover maps generated by U-Net models trained on OpenEarthMap, LoveDA, DeepGlobe, and DynamicEarthNet. The NAIP images are the source data.

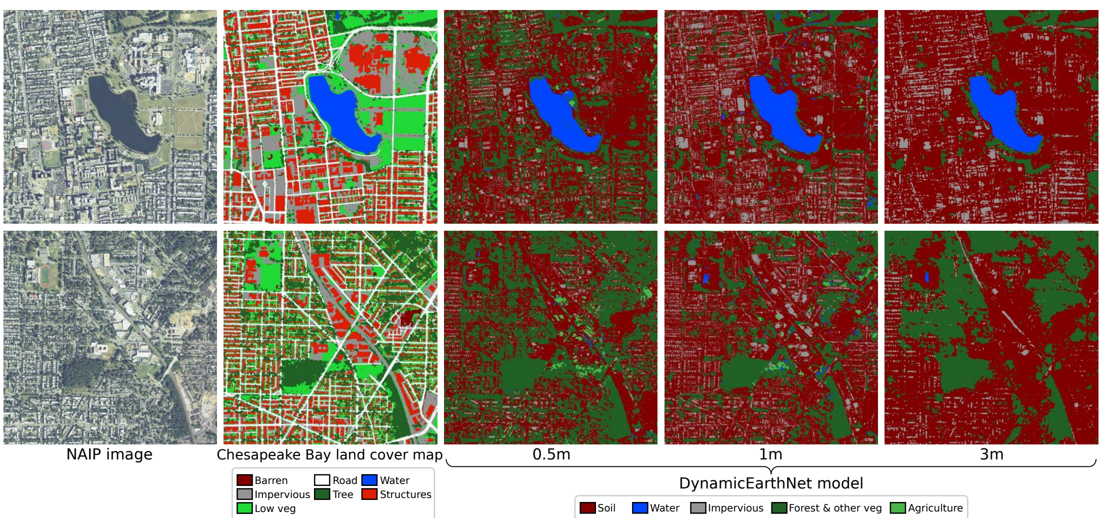

Figure 10: Visual comparison of land cover maps generated by U-Net trained on DynamicEarthNet using NAIP images at different GSDs (i.e., 0.5cm, 1m, and 3m) for inference.

Figure 11: Out-of-sample mapping examples of MiniFrance from France.

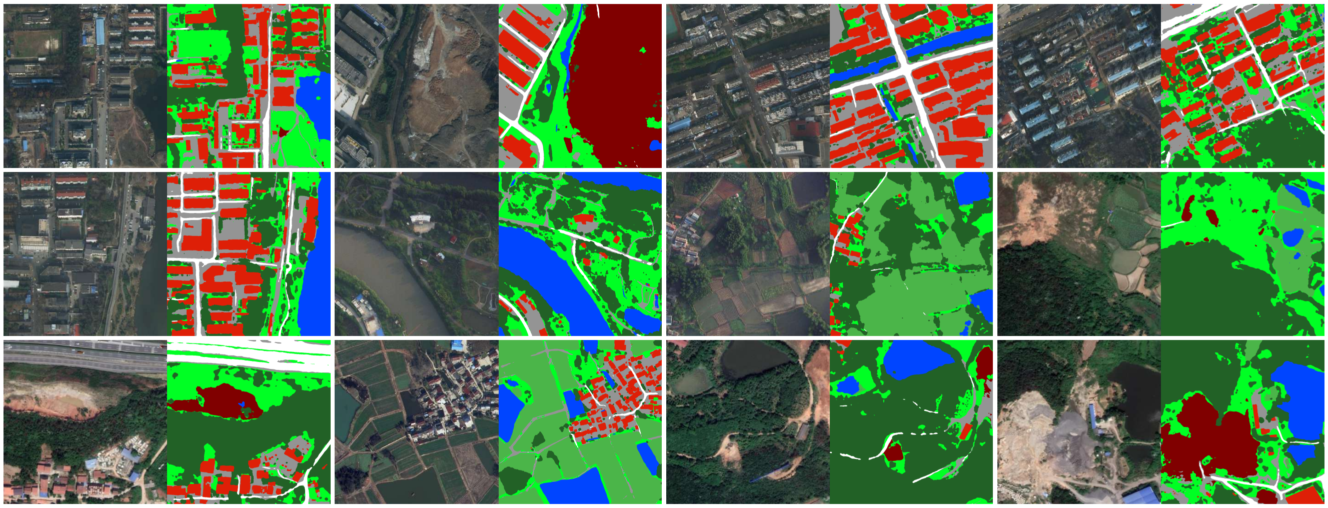

Figure 12: Out-of-sample mapping examples of LoveDA from China.

Figure 13: Out-of-sample mapping example of SIGTIERRAS from Ecuador.

Figure 14: Out-of-sample mapping example of ZMI from Tanzania.

Table 6: Attribution of source data of the 97 regions in OpenEarthMap.

Region	Country	Source data	URL	Provider	License
Christchurch	New Zealand	AIRS	https://www.airs-dataset.com/	Land Information of New Zealand	CC BY 4.0
Aachen	Germany	GeoNRW	https://www.opengeodata.nrw.de/produkte/	German state North Rhine-Westphalia	DL-DE-BY-2.0
Bielefeld	Germany	GeoNRW	https://www.opengeodata.nrw.de/produkte/	German state North Rhine-Westphalia	DL-DE-BY-2.0
Dortmunt	Germany	GeoNRW	https://www.opengeodata.nrw.de/produkte/	German state North Rhine-Westphalia	DL-DE-BY-2.0
Dusseldorf	Germany	GeoNRW	https://www.opengeodata.nrw.de/produkte/	German state North Rhine-Westphalia	DL-DE-BY-2.0
Koeln	Germany	GeoNRW	https://www.opengeodata.nrw.de/produkte/	German state North Rhine-Westphalia	DL-DE-BY-2.0
Muenster	Germany	GeoNRW	https://www.opengeodata.nrw.de/produkte/	German state North Rhine-Westphalia	DL-DE-BY-2.0
Chisinau	Moldova	HTCD	https://map.openaerialmap.org	Lightcyphers	CC BY 4.0
Tyrol	Austria	Inria	https://project.inria.fr/aerialimagelabeling/	Tyrol	Public domain
Vienna	Austria	Inria	https://project.inria.fr/aerialimagelabeling/	Vienna	Public domain
Austin	USA	Inria	https://project.inria.fr/aerialimagelabeling/	USGS	Public domain
Chicago	USA	Inria	https://project.inria.fr/aerialimagelabeling/	USGS	Public domain
Kitsap	USA	Inria	https://project.inria.fr/aerialimagelabeling/	USGS	Public domain
Dolnoslaskie	Poland	Landcover.ai	https://landcover.ai.linuxpolska.com/	Head Office of Geodesy and Cartography	CC BY-NC-SA 4.0
Kujawsko-pomorskie	Poland	Landcover.ai	https://landcover.ai.linuxpolska.com/	Head Office of Geodesy and Cartography	CC BY-NC-SA 4.0
Lodzkie	Poland	Landcover.ai	https://landcover.ai.linuxpolska.com/	Head Office of Geodesy and Cartography	CC BY-NC-SA 4.0
Lubuskie	Poland	Landcover.ai	https://landcover.ai.linuxpolska.com/	Head Office of Geodesy and Cartography	CC BY-NC-SA 4.0
Malopolskie	Poland	Landcover.ai	https://landcover.ai.linuxpolska.com/	Head Office of Geodesy and Cartography	CC BY-NC-SA 4.0
Mazowieckie	Poland	Landcover.ai	https://landcover.ai.linuxpolska.com/	Head Office of Geodesy and Cartography	CC BY-NC-SA 4.0
Podkarpackie	Poland	Landcover.ai	https://landcover.ai.linuxpolska.com/	Head Office of Geodesy and Cartography	CC BY-NC-SA 4.0
Podlaskie	Poland	Landcover.ai	https://landcover.ai.linuxpolska.com/	Head Office of Geodesy and Cartography	CC BY-NC-SA 4.0
Pomorskie	Poland	Landcover.ai	https://landcover.ai.linuxpolska.com/	Head Office of Geodesy and Cartography	CC BY-NC-SA 4.0
Slaskie	Poland	Landcover.ai	https://landcover.ai.linuxpolska.com/	Head Office of Geodesy and Cartography	CC BY-NC-SA 4.0
Swietokrzyskie	Poland	Landcover.ai	https://landcover.ai.linuxpolska.com/	Head Office of Geodesy and Cartography	CC BY-NC-SA 4.0
Warminsko-mazurskie	Poland	Landcover.ai	https://landcover.ai.linuxpolska.com/	Head Office of Geodesy and Cartography	CC BY-NC-SA 4.0
Wielkopolskie	Poland	Landcover.ai	https://landcover.ai.linuxpolska.com/	Head Office of Geodesy and Cartography	CC BY-NC-SA 4.0
Zachodniopomorskie	Poland	Landcover.ai	https://landcover.ai.linuxpolska.com/	Head Office of Geodesy and Cartography	CC BY-NC-SA 4.0
Ngaoundere	Cameroon	Open Cities AI	https://www.drivendata.org/competitions/60/building-segmentation-disaster-resilience/	Global Facility for Disaster Reduction and Recovery	CC BY 4.0
Kinshasa	Congo	Open Cities AI	https://www.drivendata.org/competitions/60/building-segmentation-disaster-resilience/	Global Facility for Disaster Reduction and Recovery	CC BY 4.0
Pointenoire	Congo	Open Cities AI	https://www.drivendata.org/competitions/60/building-segmentation-disaster-resilience/	Global Facility for Disaster Reduction and Recovery	CC BY 4.0
Accra	Ghana	Open Cities AI	https://www.drivendata.org/competitions/60/building-segmentation-disaster-resilience/	Global Facility for Disaster Reduction and Recovery	CC BY 4.0
Monrovia	Liberia	Open Cities AI	https://www.drivendata.org/competitions/60/building-segmentation-disaster-resilience/	Global Facility for Disaster Reduction and Recovery	CC BY 4.0
Niamey	Niger	Open Cities AI	https://www.drivendata.org/competitions/60/building-segmentation-disaster-resilience/	Global Facility for Disaster Reduction and Recovery	CC BY 4.0
Mahe	Seychelles	Open Cities AI	https://www.drivendata.org/competitions/60/building-segmentation-disaster-resilience/	Global Facility for Disaster Reduction and Recovery	CC BY 4.0
Dar es salaam	Tanzania	Open Cities AI	https://www.drivendata.org/competitions/60/building-segmentation-disaster-resilience/	Global Facility for Disaster Reduction and Recovery	CC BY 4.0
Dar es salaam	Tanzania	Open Cities AI	https://www.drivendata.org/competitions/60/building-segmentation-disaster-resilience/	Global Facility for Disaster Reduction and Recovery	CC BY 4.0
Zanzibar	Tanzania	Open Cities AI	https://www.drivendata.org/competitions/60/building-segmentation-disaster-resilience/	Global Facility for Disaster Reduction and Recovery	CC BY 4.0
Kampala	Uganda	Open Cities AI	https://www.drivendata.org/competitions/60/building-segmentation-disaster-resilience/	Global Facility for Disaster Reduction and Recovery	CC BY 4.0
Buenos aires	Argentina	Open data	https://map.openaerialmap.org	Municipalidad de Pergamino	CC BY 4.0
Rosario	Argentina	Open data	https://map.openaerialmap.org	Julia Faraudello	CC BY 4.0
Melbourne	Australia	Open data	https://map.openaerialmap.org	City of Melbourne	CC BY 4.0
Cox's bazar	Bangladesh	Open data	https://map.openaerialmap.org	IOM Bangladesh - Needs and Population Monitoring (NPM) Drone	CC BY 4.0
Cox's bazar	Bangladesh	Open data	https://map.openaerialmap.org	IOM Bangladesh - Needs and Population Monitoring (NPM) Drone	CC BY 4.0
Dhaka	Bangladesh	Open data	https://map.openaerialmap.org	AIGEO Center	CC BY 4.0
Santiago	Chile	Open data	https://map.openaerialmap.org	SECTRA	CC BY 4.0
Bogota	Colombia	Open data	https://map.openaerialmap.org	Maptime Bogota	CC BY 4.0
Svaneti	Georgia	Open data	https://map.openaerialmap.org	Transcaucasian Trail Association	CC BY 4.0
Accra	Ghana	Open data	https://map.openaerialmap.org	Environmental Protection Agency Ghana	CC BY 4.0
Western	Ghana	Open data	https://map.openaerialmap.org	UMaT YouthMappers	CC BY 4.0
Al qurnah	Iraq	Open data	https://map.openaerialmap.org	Bilal Koç	CC BY 4.0
Kyoto	Japan	Open data	https://maps.gsi.go.jp/development/ichiran.html	Geospatial Information Authority of Japan	
Tokyo	Japan	Open data	https://maps.gsi.go.jp/development/ichiran.html	Geospatial Information Authority of Japan	
Monrovia	Liberia	Open data	https://map.openaerialmap.org	Uhurulabs	CC BY 4.0
Dowa	Malawi	Open data	https://map.openaerialmap.org	MapMalawi	CC BY 4.0
Ulaanbaatar	Mongolia	Open data	https://map.openaerialmap.org	City of Ulaanbaatar and Asia Foundation	CC BY 4.0
Maputo	Mozambique	Open data	https://map.openaerialmap.org	MapeandoMeuBairro	CC BY 4.0
Abancay	Peru	Open data	https://sigrid.cenepred.gob.pe	National Center for Disaster Risk Estimation, Prevention and Reduction	
Chiclayo	Peru	Open data	https://sigrid.cenepred.gob.pe	National Center for Disaster Risk Estimation, Prevention and Reduction	
Chincha	Peru	Open data	https://sigrid.cenepred.gob.pe	National Center for Disaster Risk Estimation, Prevention and Reduction	
Ica	Peru	Open data	https://sigrid.cenepred.gob.pe	National Center for Disaster Risk Estimation, Prevention and Reduction	
Lambayeque	Peru	Open data	https://sigrid.cenepred.gob.pe	National Center for Disaster Risk Estimation, Prevention and Reduction	
Lima	Peru	Open data	https://sigrid.cenepred.gob.pe	National Center for Disaster Risk Estimation, Prevention and Reduction	
Pisco	Peru	Open data	https://sigrid.cenepred.gob.pe	National Center for Disaster Risk Estimation, Prevention and Reduction	
Piura	Peru	Open data	https://sigrid.cenepred.gob.pe	National Center for Disaster Risk Estimation, Prevention and Reduction	
Sechura	Peru	Open data	https://sigrid.cenepred.gob.pe	National Center for Disaster Risk Estimation, Prevention and Reduction	
Viru	Peru	Open data	https://sigrid.cenepred.gob.pe	National Center for Disaster Risk Estimation, Prevention and Reduction	
Baybay	Philippines	Open data	https://map.openaerialmap.org	SkyEye	CC BY 4.0
San tome	Sao Tome and Principe	Open data	https://map.openaerialmap.org	Drones Adventures	CC BY 4.0
Chiangmai	Tailand	Open data	https://map.openaerialmap.org	UR Field Lab Chiang Mai	CC BY 4.0
Lohur	Tajikistan	Open data	https://map.openaerialmap.org	FAZO Institute, Dushanbe, Tajikistan	CC BY 4.0
Kagera	Tanzania	Open data	https://map.openaerialmap.org	WeRobotics	CC BY 4.0
Zanzibar	Tanzania	Open data	https://map.openaerialmap.org	Commission for Lands and Revolutionary Government of Zanzibar	CC BY 4.0
Tonga	Tonga	Open data	https://map.openaerialmap.org	World Bank, V-TOL Aerospace	CC BY 4.0
Soriano	Uruguay	Open data	https://map.openaerialmap.org	IntelDrones SRL	CC BY 4.0
Rio	Brazil	SpaceNet	https://spacenet.ai/spacenet-buildings-dataset-v1/	Maxar	CC BY-SA 4.0
Shanghai	China	SpaceNet	https://spacenet.ai/spacenet-buildings-dataset-v2/	Maxar	CC BY-SA 4.0
Paris	France	SpaceNet	https://spacenet.ai/spacenet-buildings-dataset-v2/	Maxar	CC BY-SA 4.0
Rotterdam	Netherlands	SpaceNet	https://spacenet.ai/sn6-challenge/	Maxar	CC BY-SA 4.0
Khartoum	Sudan	SpaceNet	https://spacenet.ai/spacenet-buildings-dataset-v2/	Maxar	CC BY-SA 4.0
Vegas	USA	SpaceNet	https://spacenet.ai/spacenet-buildings-dataset-v2/	Maxar	CC BY-SA 4.0
Adelaide	Australia	xBD	https://xview2.org/	Maxar	CC BY-NC-SA 4.0
El rodeo	Guatemala	xBD	https://xview2.org/	Maxar	CC BY-NC-SA 4.0
Jeremie	Haiti	xBD	https://xview2.org/	Maxar	CC BY-NC-SA 4.0
Les-cayes	Haiti	xBD	https://xview2.org/	Maxar	CC BY-NC-SA 4.0
Port-a-piment	Haiti	xBD	https://xview2.org/	Maxar	CC BY-NC-SA 4.0
Saint-louis-du-sud	Haiti	xBD	https://xview2.org/	Maxar	CC BY-NC-SA 4.0
Palu	Indonesia	xBD	https://xview2.org/	Maxar	CC BY-NC-SA 4.0
Labuhan	Malaysia	xBD	https://xview2.org/	Maxar	CC BY-NC-SA 4.0
Mexico city	Mexico	xBD	https://xview2.org/	Maxar	CC BY-NC-SA 4.0
Gorakhpur	Nepal	xBD	https://xview2.org/	Maxar	CC BY-NC-SA 4.0
Pedrogao grande	Portugal	xBD	https://xview2.org/	Maxar	CC BY-NC-SA 4.0
Houston	USA	xBD	https://xview2.org/	Maxar	CC BY-NC-SA 4.0
Joplin	USA	xBD	https://xview2.org/	Maxar	CC BY-NC-SA 4.0
Leilane estates	USA	xBD	https://xview2.org/	Maxar	CC BY-NC-SA 4.0
Little rock	USA	xBD	https://xview2.org/	Maxar	CC BY-NC-SA 4.0
Oklahoma	USA	xBD	https://xview2.org/	Maxar	CC BY-NC-SA 4.0
Panama city	USA	xBD	https://xview2.org/	Maxar	CC BY-NC-SA 4.0
Santa rosa	USA	xBD	https://xview2.org/	Maxar	CC BY-NC-SA 4.0
Thousand oaks	USA	xBD	https://xview2.org/	Maxar	CC BY-NC-SA 4.0
Tulsa	USA	xBD	https://xview2.org/	Maxar	CC BY-NC-SA 4.0
Tuscaloosa	USA	xBD	https://xview2.org/	Maxar	CC BY-NC-SA 4.0
Wallace	USA	xBD	https://xview2.org/	Maxar	CC BY-NC-SA 4.0

References

[1] James Richard Anderson. *A land use and land cover classification system for use with remote sensor data*, volume 964. US Government Printing Office, 1976.

[2] Javiera Castillo-Navarro, Bertrand Le Saux, Alexandre Boulch, Nicolas Audebert, and Sébastien Lefèvre. Semi-supervised semantic segmentation in earth observation: The minifrance suite, dataset analysis and multi-task network study. *Machine Learning*, pages 1–36, 2021.

[3] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[4] Wuyang Chen, Xinyu Gong, Xianming Liu, Qian Zhang, Yuan Li, and Zhangyang Wang. Fasterseg: Searching for faster real-time semantic segmentation. In *International Conference on Learning Representations*, 2020.

[5] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting spatial attention design in vision transformers. In *NeurIPS*, 2021.

[6] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation`, 2020.

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

[8] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *CVPR*, 2022.

[9] Qing Lian, Fengmao Lv, Lixin Duan, and Boqing Gong. Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach. In *ICCV*, 2019.

[10] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *International Conference on Learning Representations*, 2019.

[11] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[12] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022.

[13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[14] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *CVPR*, 2019.

[15] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. In *ECCV*, 2020.

[16] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[17] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

[18] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018.

[19] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474, 2014.

[20] Haoran Wang, Tong Shen, Wei Zhang, Lingyu Duan, and Tao Mei. Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation. In *ECCV*, 2020.

[21] Junjue Wang, Zhuo Zheng, Ailong Ma, Xiaoyan Lu, and Yanfei Zhong. Loveda: A remote sensing land-cover dataset for domain adaptive semantic segmentation. *arXiv preprint arXiv:2110.08733*, 2021.

[22] Libo Wang, Rui Li, Ce Zhang, Shenghui Fang, Chenxi Duan, Xiaoliang Meng, and Peter M. Atkinson. Unetformer: A unet-like transformer for efficient semantic segmentation of remote sensing urban scene imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 190:196–214, 2022.

[23] Ximei Wang, Ying Jin, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Transferable normalization: Towards improving transferability of deep neural networks. In *NeurIPS*, 2019.

[24] Huikai Wu, Junge Zhang, and Kaiqi Huang. Sparsemask: Differentiable connectivity learning for dense image prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[25] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018.

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

[27] Pavel Yakubovskiy. Segmentation models pytorch. `https://github.com/qubvel/segmentation_models.pytorch`, 2020.

[28] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-Net: Towards unified image segmentation. In *NeurIPS*, 2021.

[29] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H.S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, 2021.

[30] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, pages 289–305, 2018.