

Unsupervised Domain Adaptation Architecture Search with Self-Training for Land Cover Mapping

Clifford Broni-Bediako*

Junshi Xia*

Naoto Yokoya^{†,*}

*RIKEN Center for Advanced Intelligence Project (AIP), RIKEN, Tokyo, Japan

[†]Graduate School of Frontier Sciences, The University of Tokyo, Chiba, Japan

{clifford.broni-bediako, junshi.xia}@riken.jp, yokoya@k.u-tokyo.ac.jp

Abstract

Unsupervised domain adaptation (UDA) is a challenging open problem in land cover mapping. Previous studies show encouraging progress in addressing cross-domain distribution shifts on remote sensing benchmarks for land cover mapping. The existing works are mainly built on large neural network architectures, which makes them resource-hungry systems, limiting their practical impact for many real-world applications in resource-constrained environments. Thus, we proposed a simple yet effective framework to search for lightweight neural networks automatically for land cover mapping tasks under domain shifts. This is achieved by integrating Markov random field neural architecture search (MRF-NAS) into a self-training UDA framework to search for efficient and effective networks under a limited computation budget. This is the first attempt to combine NAS with self-training UDA as a single framework for land cover mapping. We also investigate two different pseudo-labelling approaches (confidence-based and energy-based) in self-training scheme. Experimental results on two recent datasets (OpenEarthMap & FLAIR #1) for remote sensing UDA demonstrate a satisfactory performance. With only less than 2M parameters and 30.16 G FLOPs, the best-discovered lightweight network reaches state-of-the-art performance on the regional target domain of OpenEarthMap (59.38% mIoU) and the considered target domain of FLAIR #1 (51.19% mIoU). The code is at <https://github.com/cliffbb/UDA-NAS>.

1. Introduction

Land cover mapping with very high-resolution optical (VHR) remote sensing (RS) imagery provides detailed information about the Earth’s surface for applications such as environmental monitoring [12, 70], urban planning [23], disaster response and damage assessment [1, 64], and precision agriculture [50]. With supervised learning methods

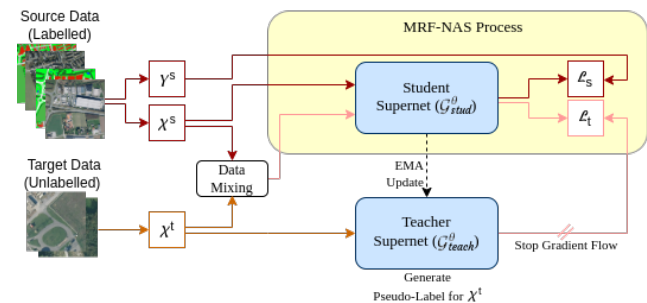


Figure 1. Self-training UDA with MRF-NAS process. The *supernet* can be any neural network. After the search, i.e., learning pairwise factors in MRF of the student supernet $\mathcal{G}_{stud}^\theta$, m optimal sub-nets (lightweight networks) are inference over the learned factors via diverse M-best loopy inference [45]. Then, we retrained the discovered lightweight networks. \mathcal{L}_s and \mathcal{L}_t are cross-entropy loss functions for the source labelled data and target pseudo-labelled data, respectively. The teacher supernet $\mathcal{G}_{teach}^\theta$ is only used to generate pseudo-labels for the target images to train $\mathcal{G}_{stud}^\theta$, in addition to the source labelled data. See Section 3 for more details.

[47], land cover mapping can be fulfilled automatically by semantic categorization of each pixel in a remotely-sensed optical image. Supervised learning methods assume that the training data and test data have the same distribution, which might not be possible in real-world land cover mapping tasks since the varying geographical regions present vastly different land features with different distributions (referred to as domain-shift) [46]. Thus, well-trained classifiers, for example, deep neural networks, perform poorly on the test data (target domain) that is different from the training data (source domain) distribution [4].

Annotating a large dataset of a new geographical region (new domain) for supervised learning is costly and time-consuming [18, 65]. Researchers in the RS community have explored unsupervised domain adaptation (UDA) techniques [67] to adapt neural networks trained on labelled data in one domain (source domain) to a new domain (target domain), which only has unlabelled data. While successful progress has been made in the traditional UDA for land

cover mapping [30, 33, 52], lightweight UDA remains a challenge and under-investigated. The existing UDA methods for land cover mapping rely on handcrafted large neural network architectures [8, 10, 24], which may not be practically suitable for real-world applications across a wide range of resource-constrained platforms such as nanosatellites (CubeSat) [15], drones [42], and mobile devices [40].

Marsocci *et al.*, [38] recently proposed GeoMultiTaskNet with 33M parameters to address the lightweight network challenge in UDA land cover mapping tasks. The size of GeoMultiTaskNet might not fit a wide range of environments where computation resources are limited. Thus, it is important to develop lightweight networks for UDA land cover mapping tasks. This will provide an opportunity for a practical land cover mapping domain adaptation deployment in resource-constrained environments. Doing this manually is a daunting exercise, especially, since the optimal networks can be data-dependent across different geographical regions and scenes. Thus, we turn to automated neural architecture search (NAS) [17] to automatically search for lightweight networks with different computation budgets for resource-constrained platforms in UDA land cover mapping tasks. In computer vision, there are some studies on NAS for UDA, but most of the studies have been performed in UDA for image classification [11, 29, 41, 48, 51, 72]. The challenging UDA for semantic segmentation [16] is barely investigated. On the other hand, NAS for UDA in RS remains unexplored.

To this end, in this paper, we explore lightweight NAS for UDA land cover mapping on two new RS benchmarks, OpenEarthMap [65] and FLAIR #1 [18] datasets, which have been properly curated for UDA tasks in land cover mapping. Specifically, we extend Markov random fields neural architecture search (MRF-NAS) [62] to a UDA scheme to learn a single neural network, called *super-net*, using both source domain labelled data and target domain unlabelled data, and inference over the learnt super-net to find subnets (lightweight networks) for resource-constrained platforms. Adapting NAS into UDA is challenging as the target domain has only unlabelled data [48]. Just searching and building networks, and then applying the networks for domain adaptation might not be effective [11, 16]. It requires different training configurations for the weights and architecture adaptation [41]. Therefore, we propose *self-training UDA-NAS framework* of which the workflow process is shown in Figure 1. Self-training UDA employs a teacher network to produce pseudo-labels for a target domain. Applying pseudo-labelling naively in UDA can lead to training difficulty for a network to learn the hard-to-predict classes [77]. For this reason, in addition to confidence-based pseudo-labelling, which is commonly used in self-training UDA [20, 56, 73], we also investigate energy-based pseudo-labelling technique [69] for

self-training in the context of UDA-NAS.

The key contributions of this paper are summarised as follows. We propose a self-training UDA-NAS framework, which extends MRF-NAS to a UDA scheme, to search for lightweight networks for land cover mapping that has been beyond the reach of previous UDA land cover mapping methods. This offers an opportunity to leverage the advancement made in UDA land cover mapping to address the practical problem of deploying UDA in a resource-constrained environment. Moreover, with very few parameters ($< 2M$), the discovered lightweight networks improve the performance of the handcrafted large networks on the OpenEarthMap and the FLAIR #1 datasets. Furthermore, the extensive experiments on the two benchmarks show promising performance of applying confidence-based pseudo-labelling techniques over energy-based pseudo-labelling in lightweight UDA-NAS for land cover mapping.

2. Related Work

2.1. UDA for Land Cover Mapping

UDA is employed in land cover mapping to alleviate the need for per-pixel annotation, aiming for the model trained with labelled source data and unlabelled target data to attain high performance on the target data by minimising the dissimilarity between the source and target domains. Many methods, including adversarial learning and self-training, are proposed. Ji *et al.* [22] proposed a generative adversarial network (GAN) based domain adaptation for land cover classification. Furthermore, a category-space-constrained adversarial method was proposed to execute category-level adaptive coastal land cover mapping [13]. Colour mapping generative adversarial network (ColorMapGANs) is proposed to reduce the colour gap between source and target domains [54]. Capliez *et al.* [9, 10] adopted adversarial learning, spatially aware self-training, and spatial pseudo-labelling for multi-sensor land cover mapping. Ma *et al.* [37] proposed novel local consistency and global diversity metrics into the UDA framework. Wang *et al.* [60] proposed the LoveCS framework that integrated cross-sensor normalization, self-training domain adaptation, and multi-scale pseudo-labelling. A novel two-stage Domain Adaptation method for Cross-Spatio-Temporal (DASCT) classification was proposed in [35].

2.2. Neural Architecture Search for UDA

NAS for UDA leverages automated architecture search to find an optimal network architecture for UDA tasks. There are few studies on NAS for UDA tasks in computer vision, but it remains unexplored in RS image understanding. AdaptNAS [29] and NASDA [28] investigated the generalization abilities of NAS for UDA with a gradient-based NAS method. AdaXpert [43] proposed a

progressive search to adjust architectures on growing data. Leveraging reinforcement learning, DAMPC-NAS [71] and AASPC [72] learned architectures that can align the source and target domains via population correlation. ABAS [48] learned an auxiliary network branch with Bayesian optimisation in the scheme of adversarial learning UDA. EvoADA [51] adopted an evolutionary algorithm to learn domain-conditioned attention mechanisms for UDA tasks. These NAS for UDA methods produce resource-hungry networks for UDA image classification. AutoAdapt[16] departed from image classification to search for UDA networks for image semantic segmentation using an evolutionary algorithm with maximum mean discrepancy. However, AutoAdapt also produces resource-hungry UDA networks, which are not suitable for resource-constrained environments. Leveraging knowledge distillation with one-shot NAS technique, SlimDA [41] and AnyDA [11] learned lightweight UDA networks for image classification, but the challenging problem of lightweight UDA for semantic segmentation remains uninvestigated. This paper attempts to address this problem by integrating MRF-NAS into self-training UDA as a single framework to search for lightweight UDA for land cover mapping.

3. Methodology

The proposed self-training UDA-NAS aims to search for lightweight networks with MRF-NAS strategy by transferring labelled source domain knowledge to an unlabelled target domain. We provide background material in Section 3.1. Self-training UDA is described in Section 3.2 and MRF-NAS is described in Section 3.3. Finally, the self-training UDA-NAS framework is illustrated in Section 3.4.

3.1. Preliminaries

Neural architecture search (NAS): Given a task, NAS aims to search for an optimal network architecture α^* automatically, together with its parameters θ , via search strategies such as reinforcement learning [75] and evolutionary algorithms [34]. Without loss of generality, the joint search for optimal architecture and its parameters for a specific task can be formulated as a bi-level optimization problem

$$\begin{aligned} \alpha^* &= \arg \min_{\alpha \in \mathcal{A}} \mathcal{L}_{search}(\theta^*(\alpha), \alpha), \\ s.t., \theta^*(\alpha) &= \arg \min_{\theta} \mathcal{L}_{train}(\theta, \alpha), \end{aligned} \quad (1)$$

where \mathcal{A} is the search space containing all possible architectures that can be found. \mathcal{L}_{search} and \mathcal{L}_{train} indicate validation loss and training loss, respectively, that can be minimised by using a validation set to update α and a training set to update θ . In the one-shot NAS approach [31, 63, 66], the architecture search space \mathcal{A} is represented as a directed acyclic graph (DAG) and the optimal network architecture

is a subgraph $\alpha \in \mathcal{A}$. We assume this definition in this work. Traditionally, NAS methods assume that both the training and the validation sets have labels and are from the same distribution, which is not the case in UDA [20, 32, 51].

Unsupervised domain adaptation (UDA): Let \mathcal{S} and \mathcal{T} respectively denote a source domain and a target domain that are related semantically. We are given a labelled dataset $\mathcal{D}_s = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^{n_s}$ of n_s examples from \mathcal{S} and unlabelled dataset $\mathcal{D}_t = \{\mathbf{x}_i^t\}_{i=1}^{n_t}$ of n_t examples from \mathcal{T} . UDA aims at mitigating the domain gap between \mathcal{S} and \mathcal{T} by training a neural network f_θ on the labelled dataset \mathcal{D}_s to exploit meaningful knowledge learned from \mathcal{S} , to perform well on the unlabelled dataset \mathcal{D}_t of \mathcal{T} [20, 32, 51].

3.2. Architecture search strategy

In this work, we leverage MRF-NAS [62] as the architecture search strategy to build the proposed self-training UDA-NAS framework. The goal is to search for lightweight networks that are optimal to achieve domain alignment between \mathcal{D}_s and \mathcal{D}_t . MRF-NAS is a resource-aware architecture search method based on AOWS [6]. It can find architectures that satisfy resource constraints such as the number of floating-point operations (FLOPs) and latency. We briefly describe MRF-NAS and we refer to Wang *et al.*, [62] for more details.

Denote \mathcal{A} as a neural network with n choices of nodes such that $\alpha_n = \{n_i | i \in [n]\}$, node n_i takes value from a finite label set Q_i (e.g., kernel size = 3, 5, or 7), and $\mathcal{A}(\alpha)$ as sub-network (subnet) of \mathcal{A} with α architecture. $[n]$ is a short form for $\{1, 2, \dots, n\}$. The MRF-NAS represents the NAS problem as maximum a posteriori (MAP) inference over pairwise MRF [25]. Let Ψ denotes a set of factors $\{\psi_S | S \subseteq [n] \text{ and } |S| \leq 2\}$ in a pairwise MRF of \mathcal{A} , where $\psi_S : \alpha_S \rightarrow \mathbb{R}$. Let $P_\Psi(\alpha) = \frac{1}{Z} \exp(\sum_S \psi_S(\alpha_S))$ denotes a probabilistic distribution of the set of factors, where Z is the normalizing constant. Given a task-specific performance measurement \mathcal{M} (e.g. classification accuracy), $P_\Psi(\alpha_1) \geq P_\Psi(\alpha_2) \Rightarrow \mathcal{M}(\mathcal{A}(\alpha_1)) \geq \mathcal{M}(\mathcal{A}(\alpha_2))$. The NAS problem then becomes MAP inference over a set of defined factors, which aims to find the optimal α^* as

$$\alpha^* = \arg \max_{\alpha} \mathcal{M}(\mathcal{A}(\alpha)) = \arg \max_{\alpha} E(P_\Psi(\alpha)), \quad (2)$$

where E is an energy function. The factors are learned via differentiable parameter learning [3, 31] by minimising the following objective

$$-\mathbb{E}_{\alpha \sim P_\Psi(\alpha)} [\mathcal{L}(\theta^* | \alpha)] \quad s.t. \quad \theta^* = \arg \min_{\theta} \mathcal{L}(\theta | \alpha) \quad (3)$$

where \mathcal{L} is a cross-entropy loss function. Monte Carlo approximation (e.g., Gibbs sampling) is applied to make the Equation (3) differentiable. Then it is solved by adopting the Gumbel-Softmax reparameterization trick to smooth the discrete categorical distribution [21].

After learning the factors, the diverse M-best loopy inference method [45] is employed to find a set of diverse optimal subnets (lightweight networks) $\{\alpha_1, \alpha_2, \dots, \alpha_m\}$. This makes the MRF-NAS method akin to one-shot NAS methods with weight sharing [5, 14]. The advantage of MRF-NAS is that it imposes factorisation, therefore, the cardinality of its set of factors does not grow exponentially in the order of the number of nodes in the network, as it does in other one-shot NAS methods [5, 14, 19].

3.3. Self-training UDA

Most existing UDA methods can be grouped into self-training [20, 56, 77] and adversarial training [36, 57, 59] approaches. In this work, we employ self-training UDA as adversarial training UDA is less stable, and the self-training methods usually perform better than the adversarial training ones [20, 56]. To address the domain gap between \mathcal{S} and \mathcal{T} , a self-training UDA method adopts a teacher network h_ϕ to generate pseudo-labels $\{\hat{\mathbf{y}}_i^t\}_{i=1}^{n_t}$ for the unlabelled images $\{\mathbf{x}_i^t\}_{i=1}^{n_t}$ in \mathcal{D}_t as

$$p_{(i,j,c)}^t = [c = \arg \max_{c'} h_\phi(\mathbf{x}_i^t)_{(j,c')}], \quad (4)$$

where $[\cdot]$ denotes Iverson bracket and c is the class index for pixel j in \mathbf{x}_i^t . As adopted in DAFormer [20], h_ϕ applies a confidence estimate with threshold τ of maximum softmax probability [56] to generate the pseudo-labels. Hence,

$$\hat{\mathbf{y}}_i^t = p_{(i,j,c)}^t \frac{\sum_{j=1}^{H \times W} [\max_{c'} h_\phi(\mathbf{x}_i^t)_{(j,c')} \geq \tau]}{H \times W}, \quad (5)$$

where H and W are the height and width of \mathbf{x}_i^t , respectively.

As an alternative, energy score threshold τ_e can also be applied to generate the pseudo-labels [69]. The energy score is defined as [26]:

$$E(\mathbf{x}_i^t, h_\phi(\mathbf{x}_i^t)) = -T \log \left(\sum_{c=1}^C e^{h_\phi(\mathbf{x}_i^t)_c / T} \right), \quad (6)$$

where $h_\phi(\cdot)_c$ indicates the corresponding logit value of the c -th class, C is the total number of classes, and T is a temperature that can be tuned. Applying the energy score, the pseudo-labels are generated as:

$$\hat{\mathbf{y}}_i^t = p_{(i,j,c)}^t [E(\mathbf{x}_i^t, h_\phi(\mathbf{x}_i^t)) < \tau_e]. \quad (7)$$

We investigate both the confidence-based pseudo-labelling and the energy-based pseudo-labelling techniques. The parameters θ of a student network f_θ is trained by minimising the following loss:

$$\mathcal{L}(\theta) = \mathbb{E}[H(f_\theta(\mathcal{X}^s), \mathcal{Y}^s) + \lambda H(f_\theta(\mathcal{X}^t), \hat{\mathcal{Y}}^t)], \quad (8)$$

where the expectation \mathbb{E} is over batches of random variables of image-label pairs $(\mathcal{X}^s, \mathcal{Y}^s)$ and $(\mathcal{X}^t, \hat{\mathcal{Y}}^t)$ sampled

uniformly from \mathcal{D}_s and \mathcal{D}_t (with generated pseudo-labels), respectively. H indicates the cross-entropy between the predictions and the reference labels averaged over all images and pixels, and λ is a hyperparameter to determine the amount of the loss in \mathcal{T} that affects the overall training. Note that the parameters ϕ of the teacher network h_ϕ are not updated via backpropagated gradients. They are updated based on the exponentially moving average of the parameters θ of the student network f_θ after each training epoch to improve the prediction stability of the teacher network h_ϕ [53]. And we adopt online self-training [2, 20, 74] in generating the pseudo-labels as the alternative offline self-training has a complex training setup [77, 78].

3.4. UDA-NAS framework

Figure 1 provides an overview of the proposed self-training UDA-NAS framework. The framework integrates NAS into UDA by applying the MRF-NAS strategy (see Section 3.2) for resource-aware search in the self-training UDA scheme (see Section 3.3) to find optimal architectures that can satisfy resource-constrained platforms. Based on the teacher-student scheme, our self-training UDA-NAS framework consists of a teacher supernet $\mathcal{G}_{teach}^\theta$ (which correspond to teacher network h_ϕ in Section 3.3) and a student supernet $\mathcal{G}_{stud}^\theta$ (which correspond to student network f_θ in Section 3.3) with the architectures parameterised by θ_{teach} and θ_{stud} , respectively, and the student supernet $\mathcal{G}_{stud}^\theta$ is configured with MRF-NAS scheme. The teacher supernet $\mathcal{G}_{teach}^\theta$ is only used to generate pseudo-labels for the unlabelled images in \mathcal{D}_t , while the architecture search takes place in the student supernet $\mathcal{G}_{stud}^\theta$ space only.

The student supernet $\mathcal{G}_{stud}^\theta$ takes both the source and target images to learn architectures, whereas the teacher supernet $\mathcal{G}_{teach}^\theta$ is fed with only the target images to generate their pseudo-labels. During the search, that is, learning pairwise and unary factors in the MRF of $\mathcal{G}_{stud}^\theta$ with differentiable parameter learning [3, 31], $\mathcal{G}_{stud}^\theta$ is trained based on supervised learning using both \mathcal{D}_s and \mathcal{D}_t (with the pseudo-labels). $\mathcal{G}_{stud}^\theta$ is trained with knowledge obtained from $\mathcal{G}_{teach}^\theta$ to enable its subnets (lightweight networks) capable of aligning the source and the target domains under limited computation budget. Following DACS [56], the ClassMix [44] mixing strategy is applied to mix the images from \mathcal{D}_s and \mathcal{D}_t to enable the subnets of $\mathcal{G}_{stud}^\theta$ to learn cross-domain robust features. The parameters θ_{teach} of $\mathcal{G}_{teach}^\theta$ are updated via the exponential moving average of the parameters θ_{stud} of $\mathcal{G}_{stud}^\theta$.

Then, after the search, the diverse M-best loopy inference [45] is applied over the learned $\mathcal{G}_{stud}^\theta$ MRF factors to find m optimal subnets (lightweight networks) that achieve good performance under a given computation budget. Finally, the discovered lightweight networks are retrained, as normally done in NAS [7, 16, 76], in the self-training UDA

mode (without the MFR-NAS process) to find the best optimal network.

4. Experimental Setup

4.1. Datasets

OpenEarthMap: OpenEarthMap serves as a benchmark dataset dedicated to advancing global high-resolution land cover mapping. It encompasses 5000 aerial and satellite images with the size of 1024×1024 , each manually annotated with 8-class land cover labels. Covering 97 regions across 44 countries on 6 continents, OpenEarthMap comprises 2.2 million segments captured at ground sampling distances ranging from 0.25 to 0.5 meters [65]. For the UDA setting, the dataset is divided into 73 regions for the source domain and 24 regions for the target domain. This division ensures a balanced representation across both domains, encompassing countries from all six continents and maintaining an equilibrium between urban and rural areas.

FLAIR #1: The French Land cover from Aerospace ImageRy dataset (FLAIR) comprises 50 spatial domains, each representing the varying landscapes and climates of metropolitan France, with each domain corresponding to a French department [18]. Each patch measures 512×512 pixels, with a GSD of 0.2 m, and each domain consists of 1725 to 1800 patches. Following the experimental settings in GeoMultiTaskNet [38], ten (D06, D08, D13, D17, D23, D29, D33, D58, D67, D74) and three (D64, D68, D71) departments are selected as the source and target domains, respectively, consisting of 16k images for training and 5k for testing with 12 classes.

4.2. Baselines

Unfortunately, we could not compare the performance of the proposed method to any UDA-NAS methods for land cover mapping as the code for the previous UDA-NAS method for semantic segmentation [16] is unavailable. However, we compare our work to 9 handcrafted, state-of-the-art, semantic segmentation UDA methods for land cover mapping. These are: AdaptSegNet [57], CLAN [36], TransNorm [61], CBST [77], and IAST [39] on the OpenEarthMap dataset (see Table 2) and AdaptSegNet [57], ADVENT [58], DAFormer [20], UDA_for_RS [27], and GeoMultiTaskNet [38] on the FLAIR #1 dataset (see Table 3).

4.3. Implementation details

All the experiments are implemented in the PyTorch library and run on a single NVIDIA Tesla P100 (DGX-2) with 32GB memory. The search is performed on the OpenEarthMap dataset. We evaluate the discovered architectures on the FLAIR #1 dataset as well.

Search space: The search space contains all the possible architectures that can be found. In this work, we adopt the

same search space provided in Wang *et al.* [62] that was designed on the U-Net architecture [49] with 4×10^{23} configurations. The search space has three neural network operations with different kernel sizes and different width ratios of the network as presented in Table 1.

Operation	Size	Width
Normal	3, 5	0.5, 0.75, 1.0, 1.25, 1.5
Downsampling	3	0.5, 0.75, 1.0, 1.25, 1.5
Upsampling	2	0.5, 0.75, 1.0, 1.25, 1.5

Table 1. The search space based on U-Net architecture.

Searching and training: During the architecture search, the student supernet $\mathcal{G}_{stud}^\theta$ is trained for 40,000 iterations with an initial warmup period of 1,500 iterations before the factors are updated. We employ the sandwich training scheme [68] to train $\mathcal{G}_{stud}^\theta$ at the largest width, smallest width, and other randomly sampled widths altogether in each iteration. We use the AdamW optimizer with a learning rate of 0.003 for the network weights and a weight decay of 0.05. To improve class imbalance, we adopt recall cross-entropy loss [55]. Following DAFormer [20], colour jitter and Gaussian blur are applied as data augmentation after a random crop of size 512×512 is applied. Other search settings such as the learning rate for the MRF factors and the hyperparameters for sampling and diverse M-best inference remain the same as in Wang *et al.* [62]. The architecture search is performed in two different pseudo-labelling scenarios of the teacher supernet $\mathcal{G}_{teach}^\theta$: (1) applying confidence-based pseudo-labelling with confidence threshold $\tau = 0.968$ as in DAFormer [20] and (2) applying energy-based pseudo-labelling with cutoff threshold $\tau_e = -8.0$ and temperature $T = 1$ as in InPL [69], to generate pseudo-labels for training $\mathcal{G}_{stud}^\theta$. For the inference, we select 8 optimal subnet architectures, 4 from each pseudo-labelling scheme, based on a computation budget of 2.5G FLOPs computed on 256×256 image resolution (smaller resolution is used for fast inference). We retrain them and select the top 2 networks from each set as the best-discovered networks. We use the same hyperparameters as in the search phase for the retraining phase, except that we train for 140,000 iterations.

5. Experimental Results

Here, we present and discuss the experimental results of the proposed method. Table 2 presents the evaluation of the discovered lightweight networks against UDA baseline methods in terms of accuracy (mIoU) on the OpenEarthMap regional target domain, the number of parameters (M), the FLOPs (G), and the latency (speed in frames per second (FPS)). The results of the baselines shown in Table 2 are based on the experiments we performed us-

Method	IoU (%)								mIoU (%) \uparrow	Params (M) \downarrow	FLOPs (G) \downarrow	Speed (FPS) \uparrow
	Bareland	Rangeland	Developed	Road	Tree	Water	Agriculture	Building				
AdaptSegNet [57]	28.77	41.47	36.09	45.16	46.65	34.48	68.47	63.74	45.60	30.05	134.80	9.17
CLAN [36]	22.90	42.25	39.49	44.12	58.98	58.99	59.51	64.53	48.85	27.61	136.16	19.06
TransNorm [61]	27.54	45.13	37.99	45.56	57.06	63.84	66.26	64.71	51.01	30.38	135.68	2.23
CBST [77]	<u>29.64</u>	43.79	37.99	49.19	57.33	60.75	<u>71.93</u>	65.46	52.01	46.59	208.18	<u>40.28</u>
IAST [39]	33.68	43.64	37.03	45.16	59.61	<u>72.08</u>	74.72	61.77	53.46	49.36	209.24	41.65
Our Net-C1	28.00	49.41	50.35	58.70	68.75	73.51	69.20	77.20	59.38	1.28	<u>30.16</u>	21.79
Our Net-C2	26.41	49.11	<u>50.22</u>	58.99	67.94	70.38	69.94	<u>77.10</u>	<u>58.76</u>	1.63	30.53	22.53
Our Net-E1	22.99	<u>49.14</u>	48.94	<u>58.74</u>	67.07	69.28	67.10	74.01	57.15	1.65	33.59	21.47
Our Net-E2	24.25	47.60	47.91	57.45	<u>68.21</u>	66.17	68.90	76.40	57.11	<u>1.45</u>	29.16	22.39

Table 2. Comparison with other UDA methods on the regional target domain test set of the OpenEarthMap dataset. In addition to the improved performance in terms of mIoU, the discovered lightweight networks are significantly smaller than that of the handcrafted baselines. The results of the baselines are based on the experiments we performed using their official implementations from GitHub. The best score is in **bold** and the second-best is underlined.

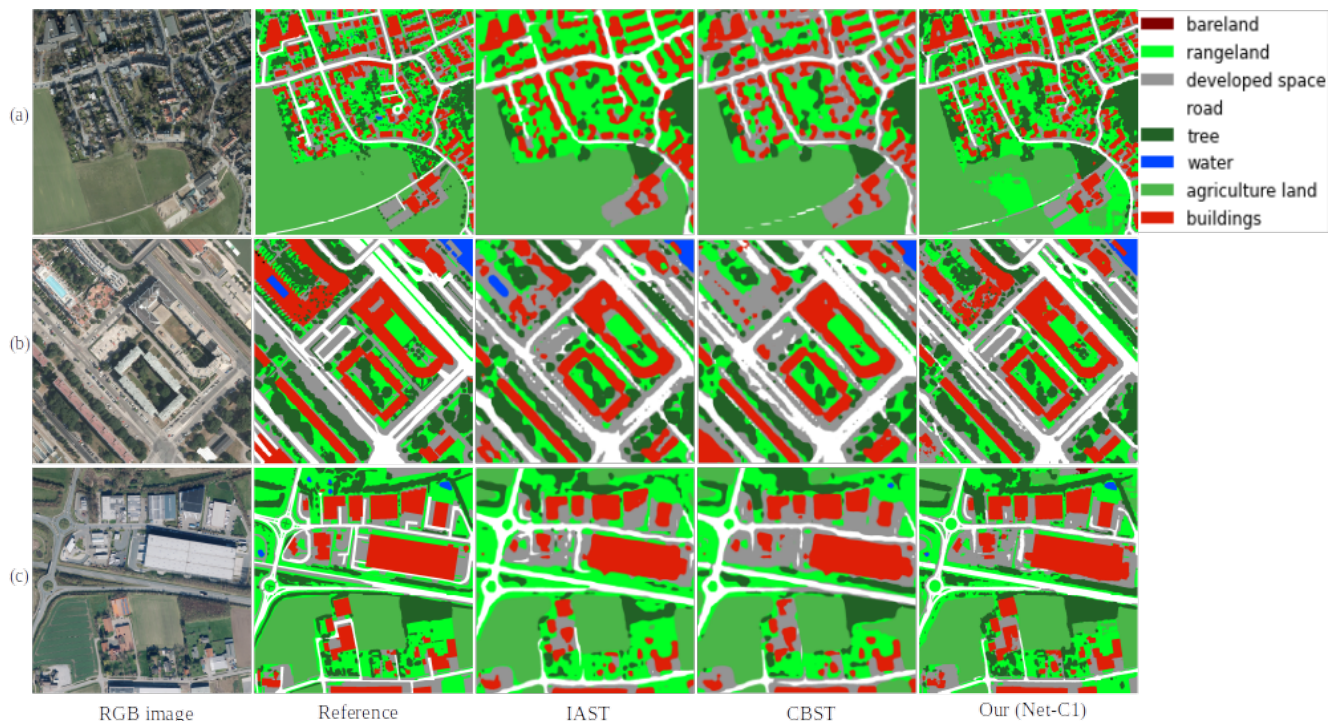


Figure 2. Visual comparison of land cover mapping results of the best-discovered network and some representative baselines in Table 2.

ing the official implementations obtained from their GitHub sites. In addition, to evaluate the transferability of the discovered networks, we employ the best-discovered networks on the considered target domain of the FLAIR #1 dataset and compared the results with other UDA methods that were reported in GeoMultiTaskNet [38] (see Table 3). The networks discovered with confidence-based pseudo-labelling are denoted as Net-C and the ones with energy-based pseudo-labelling are denoted as Net-E.

5.1. Searching on OpenEarthMap

The Net-C1 and Net-C2 in Table 2 are the top 2 networks discovered by applying confidence-based pseudo-labelling,

whereas the Net-E1 and Net-E2 are top 2 discovered by applying energy-based pseudo-labelling. As shown in the Table 2, with the lowest number of parameters (1.28M), Net-C1 outperformed both Net-E1 and Net-E2 by 2.23% (mIoU). Also, Net-C2 surpassed both Net-E1 and Net-E2 by 1.62% (mIoU). This shows that the networks discovered via confidence-based pseudo-labelling achieve better performance compared to the ones discovered via energy-based pseudo-labelling. However, we need to mention that the tunable threshold τ_e and the temperature T of the energy function were not tuned. Tuning them might have improved the performance of the energy-based pseudo-labelling.

Compared with the baseline methods based on the four

Method	IoU (%)												mIoU (%) \uparrow	Params (M) \downarrow
	Build	Perv.	Imp.	Bare	Water	Conif.	Decid.	Brush	Vine	Herb	Agric	Plowed		
AdaptSegNet [57]	39.98	20.75	40.23	20.36	15.25	4.93	35.37	10.99	34.51	42.69	11.06	23.47	24.97	99
ADVENT [58]	35.79	24.38	48.82	6.85	31.98	0.00	51.65	11.79	33.33	25.76	11.46	24.29	25.56	99
DAFormer [20]	67.09	45.56	61.99	<u>55.35</u>	65.12	8.91	54.39	20.31	<u>64.39</u>	38.79	23.74	41.83	45.61	85
UDA_for_RS [27]	66.30	48.05	62.36	59.28	61.24	9.22	60.02	16.52	57.74	40.12	30.32	<u>54.17</u>	47.02	85
GeoMultiTaskNet [38]	67.53	40.86	63.89	55.31	67.02	13.85	<u>60.97</u>	14.08	53.09	40.33	35.02	54.79	47.22	33
Our Net-C1	<u>74.96</u>	<u>49.92</u>	<u>70.34</u>	50.61	72.49	10.99	58.96	25.55	69.58	<u>43.75</u>	42.42	44.78	51.19	1.28
Our Net-E1	75.21	51.17	70.44	52.26	<u>68.95</u>	<u>11.21</u>	61.25	<u>25.31</u>	51.62	44.38	<u>40.16</u>	47.97	<u>49.99</u>	<u>1.65</u>

Table 3. Comparison with other UDA methods on the considered target domain test set of the FLAIR #1 dataset. The discovered lightweight networks improved the performance of the handcrafted baselines, which shows a strong transferability of the discovered networks. Here, all the baselines results were reported in GeoMultiTaskNet [38]. The best score is in **bold** and the second-best is underlined. (Note: *Build*=*Building*, *Perv.*=*Pervious*, *Imp.*=*Impervious*, *Bare*=*Bare soil*, *Conif.*=*Coniferous*, *Decid.*=*Deciduous*, *Brush*=*Brushwood*, *Vine*=*Vineyard*, *Herb*=*Herbaceous*, *Agric*=*Agriculture land*, and *Plowed*=*Plowed land*)

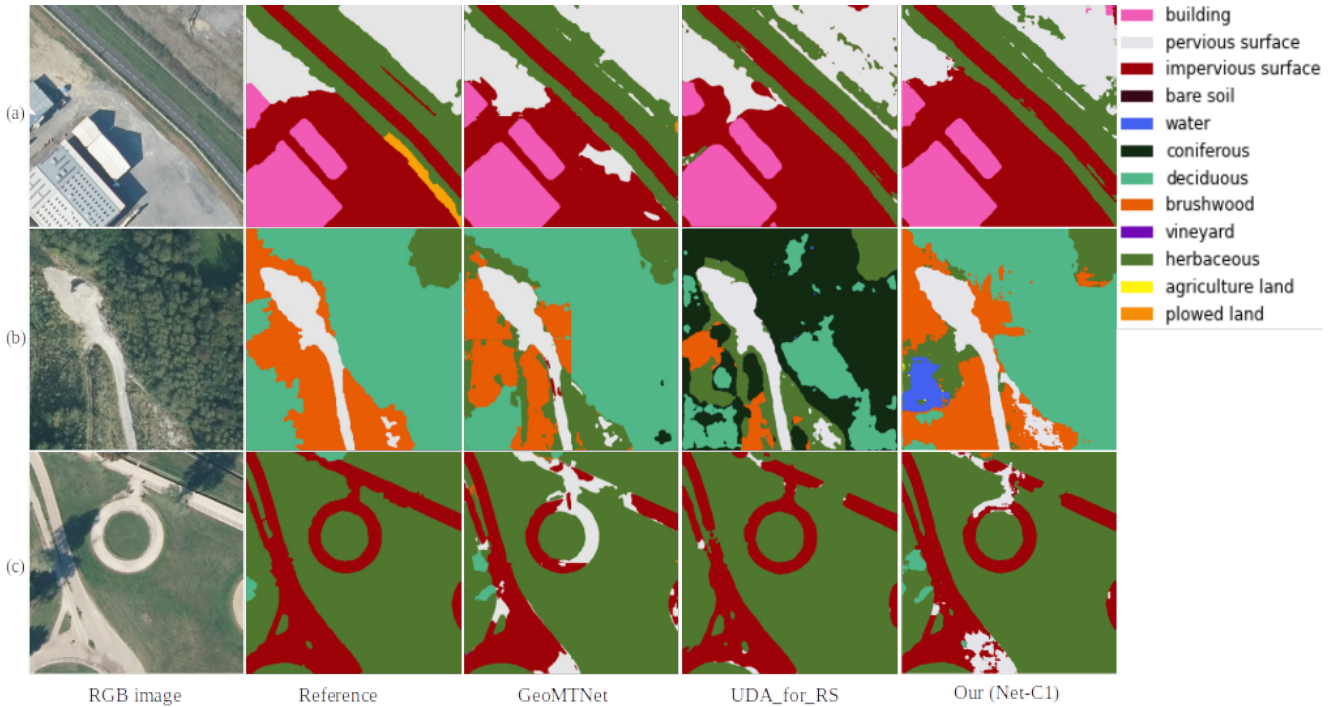


Figure 3. Visual comparison of land cover mapping results of the best-discovered network and some representative baselines in Table 3. The land cover maps of the baselines, GeoMTNet and UDA_for_RS, were obtained from GeoMultiTaskNet [38].

metrics we employed, except for the speed (FPS) that IAST and CBST topped, the discovered lightweight networks achieve better performance over all the baselines. This is because the U-Net-based architecture of discovered networks has multiple skip connections, which are good for the performance of the networks, but are detrimental to the speed of the networks. Using fewer parameters (1.28M) with only 30.16G FLOPs, the best-discovered network (Net-C1) achieved better accuracy results (59.38% mIoU) than all the representative baselines. For example, it can be seen from the Tabel 2 that Net-C1 improved the performance (mIoU) of IAST and CBST by 5.92% and 7.27%, respectively. Furthermore, Net-C1 demonstrated a big performance gap with regard to AdaptSegNet (13.78% mIoU)

and CLAN (10.53% mIoU). Focusing on individual classes, it can be observed that Net-C1 has the best performance in most of the classes, except for *bareland*, *road*, and *agriculture*. IAST gained approximately 5.5% in IoU over Net-C1 for both *bareland* and *agriculture*. The overall results of the discovered lightweight networks show that the proposed method achieves an encouraging performance.

Visualization: In Figure 2, we present examples of land cover mapping results obtained from the best-discovered network (Net-C1) and representative baselines on the OpenEarthMap regional target domain test set (see Table 2). The Net-C1 produces detailed land cover maps. In row (a), except for some portion of *agricultural land* that Net-C1 misclassified as *rangeland*, IAST and CBST wrongly classified

most of the *agricultural land* as *rangeland*. This might be because of the spectra similarity of *agricultural land* and *rangeland*. Although Net-C1 produces the best land cover map in row (b), it failed to identify the *water* at the top-left part of the image, which CBST failed as well but it was identified by IAST. However, in row (c), Net-C1 was able to identify the tiny *water* at the top-right, whereas IAST failed to do so. In all, most of the *buildings* and *roads* were wrongly classified as *developed space* by IAST and CBST because some *developed space* might have cover materials quite similar to *buildings* and *roads*.

5.2. Transferrability on FLAIR

We also employ the discovered lightweight networks on the challenging FLAIR #1 dataset. Compared to the 8 classes in the OpenEarthMap dataset, the FLAIR #1 has 12 classes, which makes it a more difficult benchmark than the OpenEarthMap. Here, we aim to find out if the discovered networks have UDA-oriented capability when they are used on different benchmark datasets. In other words, can the discovered lightweight networks achieve satisfactory performance if they are employed on datasets that they were not searched on? For each of the pseudo-labelling schemes that we employed, only the top network was evaluated on the considered target domain test set of FLAIR #1.

As presented in Table 3, using fewer parameters (less than $2M$), both Net-C1 and Net-E1 of the proposed method improved the performance of all the representative baselines results that were reported in GeoMultiTaskNet [38], the state-of-the-art UDA method on FLAIR #1. For example, Net-C1 (51.19% mIoU) and Net-E1 (49.99% mIoU) improved the performance of GeoMultiTaskNet (47.22% mIoU) and UDA_for_RS (47.02% mIoU) by approximately 4.0% and 2.7%, respectively. The encouraging results of the discovered networks is due to the U-Net-based architecture (which has proven to be good for semantic segmentation) that was employed as the base architecture for the search. For individual class IoU, the Net-C1 performed best in four classes (*water*, *brushwood*, *vineyard* and *agriculture*), and Net-E1 in five classes (*building*, *pervious*, *impervious*, *deciduous*, and *herbaceous*). The GeoMultiTaskNet achieved the best IoUs in *coniferous* and *plowed land*, whereas UDA_for_RS in *bare soil*. Overall, the results on the considered target domain test set of FLAIR #1 show that the discovered lightweight networks have UDA-oriented capability to transfer the knowledge learnt from a source domain to a target domain.

Visualization: Figure 3 presents some examples of land cover mapping results on the test set of the FLAIR #1 target domain that we considered. The maps are obtained from the best-discovered network (Net-C1) and some of the representative baselines provided in Table 3. Net-C1 and UDA_for_RS misclassified some portion of *pervious*

surface as *herbaceous* in the row (a), however, GeoMTNet wrongly classified some *impervious surface* as *pervious surface*. In row (b), while Net-C1 wrongly classified some *deciduous* as *water* and *brushwood* as *herbaceous*, GeoMTNet misclassified most of the *brushwood* as *herbaceous*. UDA_for_RS produced the most misclassification of all the classes. For example, *deciduous* was easily confused with *coniferous* and *brushwood* was confused with *herbaceous*. This might be due to spatial pattern similarities among the images as alluded by GeoMultiTaskNet [38]. In row (c), UDA_for_RS produced the best result for the *impervious surface* but failed to identify the *deciduous* at the centre-left of the image. Net-C1 was able to identify the *deciduous*, but it misclassified some *herbaceous* as *deciduous*. GeoMTNet wrongly classified most *impervious surface* as *pervious surface*. The *impervious surface* and *pervious surface* might be confused easily due to their spectra similarities.

5.3. Limitations

Although the proposed method demonstrated an encouraging performance it has some limitations. First, the overhead of the Gibbs sampling and inference over a complex loopy network. Hence, the inference has to be approximated as it is normally done in MRF [25] to minimise the cost of inference, which might have affected the optimal solution. Second, the speed of the discovered networks might not be suitable for real-time applications. It would make sense to improve the speed of the discovered networks, maybe by pruning the unnecessary skip connections in the network. Finally, we partially answer the question: "What pseudo-labelling technique is best suited for UDA-NAS?". Our observations indicate that confidence-based pseudo-labelling is a good choice. However, we did not tune the hyperparameters of the energy-based pseudo-labelling, hence, the need for further studies in this direction.

6. Conclusion

The existing UDA methods for land cover mapping are resource-hungry systems, and they are not suitable for real-world applications in resource-constrained platforms. In this paper, we introduce a novel approach to search for lightweight neural networks under a limited resource budget for UDA land cover mapping. In particular, we leverage a self-training UDA framework with a Markov random filed architecture search strategy to learn lightweight architectures that can transfer knowledge learned from a source domain to a target domain. We demonstrate the effectiveness of the proposed approach on two recent UDA benchmark datasets of remote sensing, which improve the performance of several competing UDA methods. We are hoping the UDA-NAS framework proposed here will also suggest something inspiring for the research on UDA-NAS for land cover mapping and remote sensing in general.

References

- [1] Bruno Adriano, Naoto Yokoya, Junshi Xia, and Gerald Baier. *Big Earth Observation Data Processing for Disaster Damage Mapping*, pages 99–118. Springer International Publishing, 2021. 1
- [2] Nikita Araslanov and Stefan Roth. Self-supervised augmentation consistency for adapting semantic segmentation. In *Proceedings of the IEEE/CVF Conference on CVPR*, pages 15384–15394, 2021. 4
- [3] Randy Ardywibowo, Shahin Boluki, Xinyu Gong, Zhangyang Wang, and Xiaoning Qian. NADS: Neural architecture distribution search for uncertainty awareness. In *Proceedings of the 37th International Conference on Machine Learning*, 2020. 3, 4
- [4] Prasanna B, Sunandini Sanyal, and R. Venkatesh Babu. Continual domain adaptation through pruning-aided domain-specific weight modulation. In *Proceedings of the IEEE/CVF Conference on CVPR Workshops*, pages 2456–2462, 2023. 1
- [5] Gabriel Bender, Pieter-Jan Kindermans, Barret Zoph, Vijay Vasudevan, and Quoc Le. Understanding and simplifying one-shot architecture search. In *International Conference on Machine Learning*, pages 550–559, 2018. 4
- [6] Maxim Berman, Leonid Pishchulin, Ning Xu, Matthew B Blaschko, and Gérard Medioni. AOWS: Adaptive and optimal network width search with latency constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11217–11226, 2020. 3
- [7] Clifford Broni-Bediako, Yuki Murata, Luiz H Mormille, and Masayasu Atsumi. Evolutionary nas for aerial image segmentation with gene expression programming of cellular encoding. *Neural Computing and Applications*, 34:14185–14204, 2022. 4
- [8] Yuxiang Cai, Yingchun Yang, Qiyi Zheng, Zhengwei Shen, Yongheng Shang, Jianwei Yin, and Zhongtian Shi. BiFDANet: Unsupervised Bidirectional Domain Adaptation for Semantic Segmentation of Remote Sensing Images. *Remote Sensing*, 14(1), 2022. 2
- [9] Emmanuel Capliez, Dino Ienco, Raffaele Gaetano, Nicolas Baghdadi, and Adrien Hadj Salah. Temporal-domain adaptation for satellite image time-series land-cover mapping with adversarial learning and spatially aware self-training. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 16:3645–3675, 2023. 2
- [10] Emmanuel Capliez, Dino Ienco, Raffaele Gaetano, Nicolas Baghdadi, Adrien Hadj Salah, Matthieu Le Goff, and Florient Chouteau. Multisensor temporal unsupervised domain adaptation for land cover mapping with spatial pseudo-labeling and adversarial learning. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–16, 2023. 2
- [11] Omprakash Chakraborty, Adarsh Sahoo, Rameswar Panda, and Abir Das. AnyDA: Anytime domain adaptation. In *The Eleventh International Conference on Learning Representations*, 2023. 2, 3
- [12] Hongruixuan Chen, Cuiling Lan, Jian Song, Clifford Broni-Bediako, Junshi Xia, and Naoto Yokoya. Land-cover change detection using paired openstreetmap data and optical high-resolution imagery via object-guided transformer, 2023. arXiv:2310.02674. 1
- [13] Jifa Chen, Guojun Zhai, Gang Chen, Bo Fang, Ping Zhou, and Nan Yu. Unsupervised domain adaption for high-resolution coastal land cover mapping with category-space constrained adversarial network. *Remote Sensing*, 13(8), 2021. 2
- [14] Xiangxiang Chu, Bo Zhang, and Ruijun Xu. Fairnas: Rethinking evaluation fairness of weight sharing neural architecture search. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12239–12248, 2021. 4
- [15] Bradley Denby and Brandon Lucia. Orbital edge computing: Machine inference in space. *IEEE Computer Architecture Letters*, 18(1):59–62, 2019. 2
- [16] Xueqing Deng, Yi Zhu, Yuxin Tian, and Shawn Newsam. Autoadapt: Automated segmentation network search for unsupervised domain adaptation, 2021. arXiv:2106.13227. 2, 3, 4, 5
- [17] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: a survey. *J. Mach. Learn. Res.*, 20(1):1997–2017, 2019. 2
- [18] Anatol Garioud, Stéphane Peillet, Eva Bookjans, Sébastien Giordano, and Boris Wattrelos. FLAIR #1: Semantic segmentation and domain adaptation dataset, 2022. arXiv:2211.12979v5. 1, 2, 5
- [19] Zichao Guo, Xiangyu Zhang, Haoyuan Mu, Wen Heng, Zechun Liu, Yichen Wei, and Jian Sun. Single path one-shot neural architecture search with uniform sampling. In *Computer Vision–ECCV 2020*, pages 544–560, 2020. 4
- [20] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. DAFormer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *CVPR*, 2022. 2, 3, 4, 5, 7
- [21] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017. 3
- [22] Shunping Ji, Dingpan Wang, and Muying Luo. Generative adversarial network-based full-space domain adaptation for land cover classification from multiple-source remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 59(5):3816–3828, 2021. 2
- [23] Peiyuan Jia, Chen Chen, Delong Zhang, Yulong Sang, and Lei Zhang. Semantic segmentation of deep learning remote sensing images based on band combination principle: Application in urban planning and land use. *Computer Communications*, 217:97–106, 2024. 1
- [24] Rudraksh Kapil, Seyed Mojtaba Marvasti-Zadeh, Nadir Erbilgin, and Nilanjan Ray. Shadowsense: Unsupervised domain adaptation and feature fusion for shadow-agnostic tree crown detection from rgb-thermal drone imagery. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 8266–8276, 2024. 2
- [25] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009. 3, 8
- [26] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and Fugie Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006. 4

- [27] Weitao Li, Hui Gao, Yi Su, and Biffon Manyura Momanyi. Unsupervised Domain Adaptation for Remote Sensing Semantic Segmentation with Transformer. *Remote Sensing*, 14 (19), 2022. 5, 7
- [28] Yichen Li and Xingchao Peng. Network architecture search for domain adaptation, 2020. arXiv:2008.05706. 2
- [29] Yanxi Li, Zhaohui Yang, Yunhe Wang, and Chang Xu. Adapting Neural Architectures Between Domains. In *Advances in Neural Information Processing Systems*, pages 789–798, 2020. 2
- [30] Chenbin Liang, Bo Cheng, Baihua Xiao, and Yunyun Dong. Unsupervised domain adaptation for remote sensing image segmentation based on adversarial learning and self-training. *IEEE Geoscience and Remote Sensing Letters*, 20:1–5, 2023. 2
- [31] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *International Conference on Learning Representations*, 2019. 3, 4
- [32] Xiaofeng Liu, Chaehwa Yoo, Fangxu Xing, Hyejin Oh, Georges El Fakhri, Je-Won Kang, Jonghye Woo, et al. Deep unsupervised domain adaptation: A review of recent advances and perspectives. *APSIPA Transactions on Signal and Information Processing*, 11(1), 2022. 3
- [33] Yikun Liu, Xudong Kang, Yuwen Huang, Kuikui Wang, and Gongping Yang. Unsupervised domain adaptation semantic segmentation for remote-sensing images via covariance attention. *IEEE Geoscience and Remote Sensing Letters*, 19: 1–5, 2022. 2
- [34] Yuqiao Liu, Yanan Sun, Bing Xue, Mengjie Zhang, Gary G. Yen, and Kay Chen Tan. A survey on evolutionary neural architecture search. *IEEE Transactions on Neural Networks and Learning Systems*, 34(2):550–570, 2023. 3
- [35] Muying Luo and Shunping Ji. Cross-spatiotemporal land-cover classification from vhr remote sensing images with deep learning based domain adaptation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 191:105–128, 2022. 2
- [36] 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. 4, 5, 6
- [37] Ailong Ma, Chenyu Zheng, Junjue Wang, and Yanfei Zhong. Domain adaptive land-cover classification via local consistency and global diversity. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–17, 2023. 2
- [38] V. Marsocci, N. Gonthier, A. Garioud, S. Scardapane, and C. Mallet. Geomultitasknet: remote sensing unsupervised domain adaptation using geographical coordinates. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2075–2085, 2023. 2, 5, 6, 7, 8
- [39] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. In *ECCV*, 2020. 5, 6
- [40] M T Melis, F Dessì, P Loddo, C La Mantia, S Da Pelo, A M Deflorio, G Ghiglieri, B T Hailu, K Kalegele, and B N Mwasi. Improving land cover mapping: A mobile application based on esa sentinel-2 imagery. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-3:1263–1266, 2018. 2
- [41] Rang Meng, Weijie Chen, Shicai Yang, Jie Song, Luojun Lin, Di Xie, Shiliang Pu, Xinchao Wang, Mingli Song, and Yueting Zhuang. Slimmable domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7141–7150, 2022. 2, 3
- [42] Taposh Mollick, Md Golam Azam, and Sabrina Karim. Geospatial-based machine learning techniques for land use and land cover mapping using a high-resolution unmanned aerial vehicle image. *Remote Sensing Applications: Society and Environment*, 29:100859, 2023. 2
- [43] Shuaicheng Niu, Jiaxiang Wu, Guanghui Xu, Yifan Zhang, Yong Guo, Peilin Zhao, Peng Wang, and Mingkui Tan. Adaxpert: Adapting neural architecture for growing data. In *The International Conference on Machine Learning*, 2021. 2
- [44] Viktor Olsson, Wilhelm Tranheden, Juliano Pinto, and Lennart Svensson. Classmix: Segmentation-based data augmentation for semi-supervised learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1369–1378, 2021. 4
- [45] Payman, Guzman-Rivera Abner, Shakhnarovich Gregory Batra Dhruv, and Yadollahpour. Diverse m-best solutions in markov random fields. pages 1–16, 2012. 1, 4
- [46] Rongjun Qin and Tao Liu. A Review of Landcover Classification with Very-High Resolution Remotely Sensed Optical Images: Analysis Unit, Model Scalability and Transferability. *Remote Sensing*, 14(3), 2022. 1
- [47] John A Richards, John A Richards, et al. *Remote sensing digital image analysis*. Springer, 2022. 1
- [48] L. Robbiano, M. Ur Rahman, F. Galasso, B. Caputo, and F. Carlucci. Adversarial branch architecture search for unsupervised domain adaptation. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1008–1018, 2022. 2, 3
- [49] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241, 2015. 5
- [50] Santhosh K Seelan, Soizik Laguet, Grant M Casady, and George A Seielstad. Remote sensing applications for precision agriculture: A learning community approach. *Remote Sensing of Environment*, 88(1):157–169, 2003. 1
- [51] Kekai Sheng, Ke Li, Xiawu Zheng, Jian Liang, Weiming Dong, Feiyue Huang, Rongrong Ji, and Xing Sun. On evolving attention towards domain adaptation, 2021. arXiv:2103.13561. 2, 3
- [52] Pedro Juan Soto Vega, Gilson Alexandre Ostwald Pedro da Costa, Raul Queiroz Feitosa, Mabel Ximena Ortega Adarme, Claudio Aparecido de Almeida, Christian Heipke, and Franz Rottensteiner. An unsupervised domain adaptation approach for change detection and its application to deforestation mapping in tropical biomes. *ISPRS Journal of Photogrammetry and Remote Sensing*, 181:113–128, 2021. 2
- [53] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve

- semi-supervised deep learning results. *Advances in Neural Information Processing Systems*, 30, 2017. 4
- [54] Onur Tasar, S. L. Happy, Yuliya Tarabalka, and Pierre Alliez. Colormapgan: Unsupervised domain adaptation for semantic segmentation using color mapping generative adversarial networks. *IEEE Transactions on Geoscience and Remote Sensing*, 58(10):7178–7193, 2020. 2
- [55] Junjiao Tian, Niluthpol Chowdhury Mithun, Zachary Seymour, Han-Pang Chiu, and Zsolt Kira. Striking the right balance: Recall loss for semantic segmentation. In *2022 International Conference on Robotics and Automation (ICRA)*, page 5063–5069, 2022. 5
- [56] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. DACS: Domain adaptation via cross-domain mixed sampling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1379–1389, 2021. 2, 4
- [57] Y.-H. Tsai, W.-C. Hung, S. Schuler, K. Sohn, M.-H. Yang, and M. Chandraker. Learning to adapt structured output space for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 4, 5, 6, 7
- [58] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Mathieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *CVPR*, 2019. 5, 7
- [59] Haoran Wang, Tong Shen, Wei Zhang, Lingyu Duan, and Tao Mei. Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation. In *The European Conference on Computer Vision (ECCV)*, 2020. 4
- [60] Junjue Wang, Ailong Ma, Yanfei Zhong, Zhuo Zheng, and Liangpei Zhang. Cross-sensor domain adaptation for high spatial resolution urban land-cover mapping: From airborne to spaceborne imagery. *Remote Sensing of Environment*, 277: 113058, 2022. 2
- [61] Ximei Wang, Ying Jin, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Transferable normalization: Towards improving transferability of deep neural networks. In *NeurIPS*, 2019. 5, 6
- [62] Zifu Wang and Matthew B. Blaschko. MRF-UNets: Searching UNet with Markov Random Fields. In *ECML-PKDD*, 2022. 2, 3, 5
- [63] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. FbNet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *Proceedings of the IEEE/CVF Conference on CVPR*, pages 10734–10742, 2019. 3
- [64] Haobin Xia, Jianjun Wu, Jiaqi Yao, Hong Zhu, Adu Gong, Jianhua Yang, Liuru Hu, and Fan Mo. A Deep Learning Application for Building Damage Assessment Using Ultra-High-Resolution Remote Sensing Imagery in Turkey Earthquake. *International Journal of Disaster Risk Science*, 14(6): 947–962, 2023. 1
- [65] J. Xia, N. Yokoya, B. Adriano, and C. Broni-Bediako. Openearthmap: A benchmark dataset for global high-resolution land cover mapp. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6243–6253, 2023. 1, 2, 5
- [66] Sirui Xie, Hehui Zheng, Chunxiao Liu, and Liang Lin. SNAS: stochastic neural architecture search. *arXiv preprint arXiv:1812.09926*, 2018. 3
- [67] Mengqiu Xu, Ming Wu, Kaixin Chen, Chuang Zhang, and Jun Guo. The Eyes of the Gods: A Survey of Unsupervised Domain Adaptation Methods Based on Remote Sensing Data. *Remote Sensing*, 14(17), 2022. 1
- [68] Jiahui Yu and Thomas S Huang. Universally slimmable networks and improved training techniques. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1803–1811, 2019. 5
- [69] Zhuoran Yu, Yin Li, and Yong Jae Lee. InPL: Pseudo-labeling the inliers first for imbalanced semi-supervised learning. In *The Eleventh International Conference on Learning Representations*, 2023. 2, 4, 5
- [70] Qiangqiang Yuan, Huanfeng Shen, Tongwen Li, Zhiwei Li, Shuwen Li, Yun Jiang, Hongzhang Xu, Weiwei Tan, Qianqian Yang, Jiwen Wang, Jianhao Gao, and Liangpei Zhang. Deep learning in environmental remote sensing: Achievements and challenges. *Remote Sensing of Environment*, 241: 111716, 2020. 1
- [71] Zhixiong Yue, Pengxin Guo, and Yu Zhang. Domain adaptation by maximizing population correlation with neural architecture search, 2021. arXiv:2109.06652. 3
- [72] Zhixiong Yue, Pengxin Guo, Yu Zhang, and Christy Liang. Learning feature alignment architecture for domain adaptation. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2022. 2, 3
- [73] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on CVPR*, pages 12414–12424, 2021. 2
- [74] Qianyu Zhou, Zhengyang Feng, Qiqi Gu, Jiangmiao Pang, Guangliang Cheng, Xuequan Lu, Jianping Shi, and Lizhuang Ma. Context-aware mixup for domain adaptive semantic segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(2):804–817, 2022. 4
- [75] Barret Zoph and Quoc Le. Neural architecture search with reinforcement learning. In *International Conference on Learning Representations*, 2017. 3
- [76] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE Conference on CVPR*, pages 8697–8710, 2018. 4
- [77] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 289–305, 2018. 2, 4, 5, 6
- [78] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5982–5991, 2019. 4