# Dense Dilated Convolutions' Merging Network for Land Cover Classification

Qinghui Liu[1,2], *Student Member, IEEE,* Michael Kampffmeyer[2], *Member, IEEE,* Robert Jenssen[2,1], *Member, IEEE,* and Arnt-Børre Salberg[1], *Member, IEEE*

***Abstract*—Land cover classification of remote sensing images is a challenging task due to limited amounts of annotated data, highly imbalanced classes, frequent incorrect pixel-level annotations, and an inherent complexity in the semantic segmentation task. In this work, we propose a novel architecture called the Dense Dilated Convolutions Merging Network (DDCM-Net) to address this task. The proposed DDCM-Net consists of dense dilated image convolutions merged with varying dilation rates. This effectively utilizes rich combinations of dilated convolutions that enlarge the network's receptive fields with less parameters and features compared to the state-of-the-art approaches in the remote sensing domain. Importantly, DDCM-Net obtains fused local and global context information, in effect incorporating surrounding discriminative capability for multi-scale and complex shaped objects with similar color and textures in very high resolution aerial imagery. We demonstrate the effectiveness, robustness and flexibility of the proposed DDCM-Net on the publicly available ISPRS Potsdam and Vaihingen data, as well as the DeepGlobe land cover dataset. Our single model, trained on 3-band Potsdam and Vaihingen data, achieves better accuracy in terms of both mean intersection over union (mIoU) and F1-score compared to other published models trained with more than 3-band data. We further validate our model on the DeepGlobe dataset, achieving state-of-the-art result 56.2% mIoU with much less parameters and at a lower computational cost compared to related recent work.**

***Index Terms*—Deep learning, very high resolution (VHR) optical imagery, land cover classification, semantic segmentation**

## I. Introduction

AUTOMATIC semantic classification of land cover in remote sensing data is of great importance for sustainable development, autonomous agriculture, and urban planning. Thanks to the progress achieved in the deep learning and computer vision community on natural images, most deep learning architectures [1], [2], [3], [4], [5], [6] for semantic segmentation can also be used for land cover classification tasks in the remote sensing domain. Semantic segmentation refers to the assignment of a semantic category to every pixel in the images, which in this work consist of very high resolution (VHR) aerial images. Currently, the state-of-the-art end-to-end semantic segmentation models are mostly inspired by the idea of fully convolutional networks (FCNs), which generally consist of an encoder-decoder architecture [1]. All layers in the encoder and decoder modules are based on convolutional neural networks (CNN). However, to achieve higher performance, FCN-based end-to-end methods normally rely on deep and wide multi-scale CNN architectures that typically require a large number of trainable parameters and computation resources. In addition, there is also a lot of redundancy in deep CNNs that often results in vanishing gradients in backward propagation, diminishing feature reuse in forward propagation, and long training time [7].

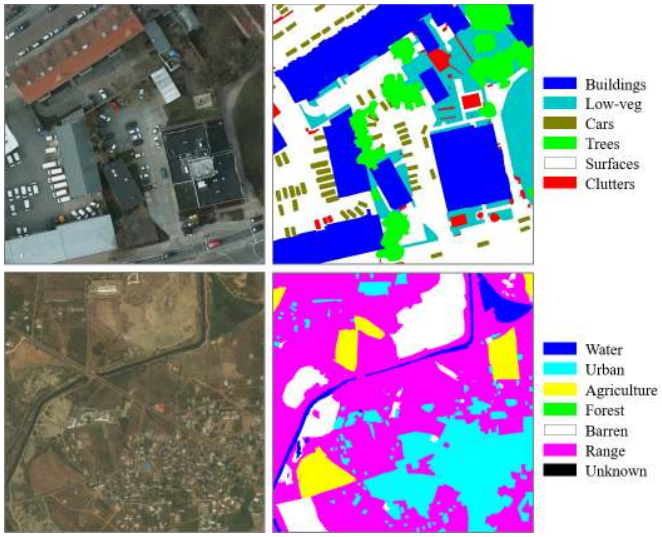



Fig. 1. Examples of land cover labels (right) and corresponding remote sensing images (left) from two different datasets (ISPRS Potsdam [8] and DeepGlobel [9]) separately. Semantic label colors are shown beside the ground truths.

In general, VHR remote sensing images contain diverse objects and intricate variations in their aspect-ratio and color-textures (e.g. roads, roofs, shadows of buildings, low plants and branches of trees [8]). Moreover, many remote sensing images are completely consisting of "stuff" classes (amorphous regions such as forest, agricultural areas, water and so on). This brings challenges for semantic mapping in remote sensing images. Fig. 1 shows illustrative examples of land cover classification for remote sensing images. Thus, richer and multi-scale global contextual representations play a key role in land cover mapping for VHR aerial images.

In this work, we propose a novel network architecture, called the dense dilated convolutions merging network (DDCM-Net), which utilizes multiple dilated convolutions with various dilation rates. The proposed network learns with densely linked dilated convolutions and outputs a fusion of all intermediate features without losing resolutions during the

[1]Norwegian Computing Center, Dept. SAMBA, P.O. Box 114 Blindern, NO-0314 OSLO, Norway

[2]UiT Machine Learning Group, Department of Physics and Technology, UiT the Arctic University of Norway, Tromsø, Norway

extraction of multi-scale features. This significantly reduces the computational redundancies and costs. Our experiments demonstrate that the network achieves robust and accurate results on the representative ISPRS 2D semantic labeling datasets [8]. Motivated by the recent success of depthwise separable convolutions [10], we also explore grouped convolutions [11] with strided operations adapted into our DDCM-Net, which is shown to further improve net speed and accuracy. We finally demonstrate the effectiveness of the adapted DDCM-Net on the DeepGlobe land cover challenge dataset [9]. In summary, our contributions are:

1) We propose a new computationally light-weight and scalable architecture based on dilated convolutions [12] that can be used as a simple, yet effective, encoder or decoder module for semantic segmentation tasks.
2) In the proposed network, one can arbitrarily control the depths, widths, groups and strides of the modules with various dilation rates in order to address different problems.
3) Our proposed end-to-end model outperforms or achieves competitive performance on different representative remote sensing datasets compared to other published related methods.

A preliminary version of this paper appeared in [13]. Here, we extend our work by (i) extending the methodology with two variants where group and strided convolutions are exploited to further boost the model's flexibility; (ii) expanding the experiment section by including more public datasets, providing more detailed training details and presenting additional result comparisons; (iii) providing in-depth discussions in terms of model's dilation and density policies, generalization, as well as a more detailed analysis of computation complexity; (iv) providing a more thorough review of related work.

The paper is structured as follows. Section II provides an overview of the related work. Section III introduces the datasets used in our work. In Section IV, we present the methodology in details. Experimental procedure and evaluation of the proposed method is performed in Section V. Section VI provides discussion of our results, and, finally in Section VII, we draw conclusions.

## II. Background

Deep learning and CNNs have been revolutionary for computer vision and image classification [14], [15] in particular. Even though segmentation can be viewed as a pixel-to-pixel classification problem, most modern CNN models for semantic segmentation are inspired by fully convolutional networks (FCNs) [1]. The FCN was the first CNN model without any fully connected layers that was trained in an end-to-end manner directly classifying each pixel to its corresponding label. However, vanilla FCNs generally cause loss of spatial information due to the presence of pooling layers that reduce the resolutions of feature maps by sacrificing the positional information of objects. In order to alleviate this issue, U-Net [2] extends the FCN by introducing skip connections between the encoder and decoder modules. In the decoder module, the spatial information is gradually recovered by fusing skipped connections with upsampling layers or de-convolution layers. Since then, the encoder-decoder architecture has been widely extended in recent works including SegNet [16], GCN [6], PSPNet [4], DUC [5], DeepLabV3+ [10] and so on. In general, these architectures differ from each other in how they capture rich and global contextual information at multiple scales. For instance, PSPNet [4] introduces a pyramid pooling module to aggregate the context by applying large kernel pooling layers, while DeepLabV3+ [10] utilizes several parallel atrous convolution with different rates (called Atrous Spatial Pyramid Pooling). Similarly, the authors of [17] presented a unified descriptor network for dense matching tasks, so called SDC-stacked dilated convolution, which combines parallel dilated convolutions with different dilation rates of ($[1, 2, 3, 4]$). Instead of the parallel combination methods, a cascading structure of dilated convolution layers was first presented in [12] with exponentially increasing rates of dilation that achieved state-of-the-art results on a natural image segmentation benchmarks in that year. The authors of [18] have also proposed a sequential structure of iterating dilated convolutions which demonstrated higher accuracy with impressive speed improvements in contrast to the previously best performing model BI-LSTM-CRF [19], for the sequence labeling tasks when processing entire documents at a time.

In contrast, our novel architecture has three major differences. Firstly, we sequentially stack the output of each layer with its input features before feeding it to the next layer in order to alleviate context information loss. Secondly, our final output is computed on all features generated by intermediate layers, which can effectively aggregate the fused receptive field of each layer and maximally utilize multi-scale context information. Thirdly, our method is much more flexible and extendable with group and strided convolutions to address different domain problems.

Our applied focus in this paper is land cover classification based on remote sensing. Lately, the FCNs and encoder-decoder architectures have been widely adapted and applied to the ISPRS [8] Semantic Labeling Contest [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], and the DeepGlobe CVPR-2018 [9] challenge of automatic classification of land cover types [30], [31], [32], [33], [34], [35], [36]. Paisitkriangkrai et al. [20] proposed a scheme for high-resolution land cover classification using a combination of a patch-based CNN and a random forest classification that is trained on hand-crafted features. Further, Sherrah [21] applied FCNs to semantic labelling of aerial imagery and illustrated that higher accuracy can be achieved than with more traditional patch-based approaches. In addition, deep learning has also been exploited for multi-modal data processing in remote sensing. For instance, Audebert et al. [25] proposed a multi-scale SegNet approach (so-called FuseNet) to leverage both a large spatial context and the high resolution data, while early and late fusion strategies of multi-modality data are also exploited. However, such fusion techniques require all modalities to be available to the classification during both training and testing. The authors of [28] therefore presented a novel CNN architecture based on so-called hallucination networks for urban land cover classification, able to replace

missing data modalities in the test phase. This enables fusion capabilities even when data modalities are missing in testing.

Recently, the authors of [27] proposed a recurrent network in fully convolutional network (RiFCN) trying to better fuse multi-level features with boundary-aware features to achieve fine-gained inferences. Similarly, the stacked U-Nets architecture is proposed in [34] for ground material segmentation in remote sensing imagery, which merges high-resolution details and the long distance context information captured at low-resolution to generate segmentation maps. Further, Kuo et al. [35] introduced an aggregation decoder in combination with DeepLabV3 architecture to fuse different-level features progressively from the encoder for final prediction, while the authors of [36] proposed a dense fusion classmate network (DFCNet) which tried to fuse auxiliary training data as "classmate" to capture supplementary features for land cover classification. One of the main ideas behind all the architectures is to take into account the multi-level context to improve the prediction of the segmentation. Even though these state-of-the-art designs could alleviate the loss of global contextual information, they are often computationally expensive with a lot of redundancy in order to capture dense and multi-scale contextual features [7]. In the following, we demonstrate that the DDCM-Net achieves competitive results or outperforms for land cover classification on benchmark datasets at a lower computational cost compared to related recent work.

## III. Benchmark Datasets

In this paper, we focus on two publicly used databases, namely the ISPRS 2D semantic labeling contest datasets [8], and the DeepGlobe land cover challenge dataset [9]. The ISPRS datasets are comprised of aerial images over two cities in Germany: Potsdam[1] and Vaihingen[2], which have been labelled with six of the most common land cover classes: impervious surfaces, buildings, low vegetation, trees, cars and clutter. The DeepGlobe land cover dataset consists of satellite data collected from the DigitalGlobe Vivid+ dataset [9], and focuses on rural areas. This includes seven types of land covers: urban (man-made, built up areas with human artifacts), agriculture (farms, cropland, orchards, vineyards, ornamental horticultural areas, and so on), rangeland (any non-forest, non-farm, green land and grass), forest (any land with at least 20% tree crown density plus clear cuts), water (rivers, oceans, lakes, wetland, ponds), barren (mountain, rock, dessert, beach, land with no vegetation), and unknown (clounds and others). Each dataset provides online leaderboards and reports test metrics measured on hold-out test images.

*1) Potsdam:* The Potsdam dataset consists of 38 tiles of size $6000\times6000$ pixels with a ground resolution of 5cm. 14 of these are used as hold-out test images. Tiles consist of Red-Green-Blue-Infrared (RGB-IR) four-channel images. While both the digital surface model (DSM) and normalized DSM (nDSM) data are also included in the dataset, we only focus on the 3-channel RGB images in this work.

[1]http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-potsdam.html

[2]http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-vaihingen.html

*2) Vaihingen:* The Vaihingen dataset contains 33 tiles of varying size (on average approximately $2100 \times 2100$ pixels) with a ground resolution of 9cm, of which 17 are used as hold-out test images. Tiles are composed of Infrared-Red-Green (IRRG) 3-channel images. Though DSMs and nDSMs data are also available for all images in the dataset, we only use IRRG data in this paper.

*3) DeepGlobe:* DeepGlobe Land Cover data contains $1146$ RGB images of size $2448 \times 2448$ pixels with a ground resolution of 50cm. 803 of these images have a publicly available ground truth and are used as the training set, while the remaining images are split into a hold-out validation and test set consisting of 171 and 172 images, respectively. Due to the variety of land cover types and density of annotations [9], this dataset is more challenging than the two above-mentioned datasets.

## IV. Methods

We first briefly revisit the concept of dilated convolutions. We then present our proposed DDCM architecture, based on such dilated convolutions. Furthermore, we provide detailed information regarding the procedure for training the network.

### A. Dilated Convolutions

Dilated convolutions [12] have been demonstrated to improve performance in many classification and segmentation tasks [10], [11], [37], [38]. One key advantage is that they allow us to flexibly adjust the filter's receptive field to capture multi-scale information without resorting to down-scaling and up-scaling operations. A 2D dilated convolution operator can be defined as

$$g(x_\ell) = \sum_{c\in C_\ell} \theta^c_{k,r} * x^c_\ell \tag{1}$$

where, $*$ denotes a convolution operator, $g : \mathbb{R}^{H_\ell\times W_\ell\times C_\ell} \rightarrow \mathbb{R}^{H_{\ell+1}\times W_{\ell+l}}$ convolves the input feature map $x_\ell \in \mathbb{R}^{H_\ell\times W_\ell\times C_\ell}$. A dilated convolution $\theta_{k,r}$ with a filter size $k$ and dilation rate $r \in \mathbb{Z}^+$ is only nonzero for a multiple of $r$ pixels from the center. In a dilated convolution, a kernel size $k$ is effectively enlarged to $k+(k-1)(r-1)$ with the dilation factor $r$. As a special case, a dilated convolution with dilation rate $r = 1$ corresponds to a standard convolution.

### B. Dense Dilated Convolutions Merging Module

The Dense Dilated Convolutions Merging Module (DDCM-Module) consists of a number of Dilated CNN-stack (DC) blocks with a merging module as output as shown in Fig. 2. A basic DC block is composed of a dilated convolution followed by PReLU [39] non-linear activation and batch normalization (BN) [40]. It then stacks the output with its input and feeds the stacked data to the next layer. The final network output is computed by a merging layer composed of $1 \times 1$ filters with PReLU and BN in order to efficiently combine all stacked features generated by intermediate DC blocks. In practice, densely connected DC blocks, typically configured with linearly or exponentially increasing dilation

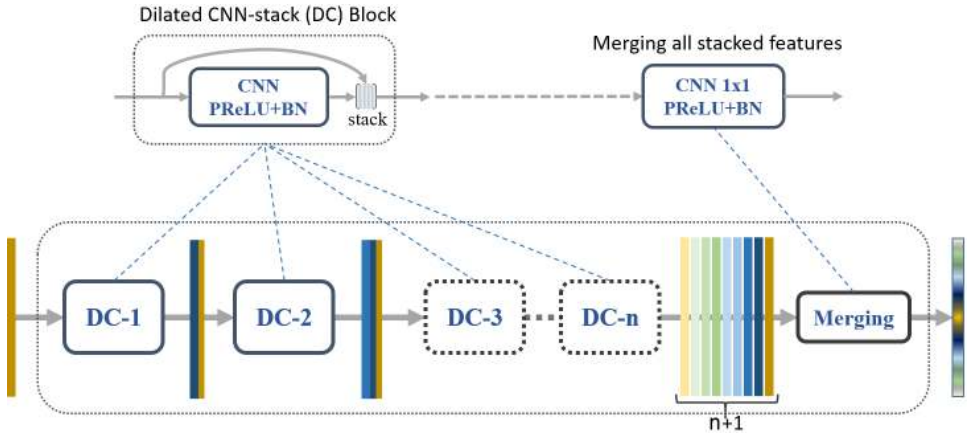



Fig. 2. Example of the DDCM architecture composed of $n$ $\{1, 2, 3, ..., n\}$ DC blocks with various dilation rates.

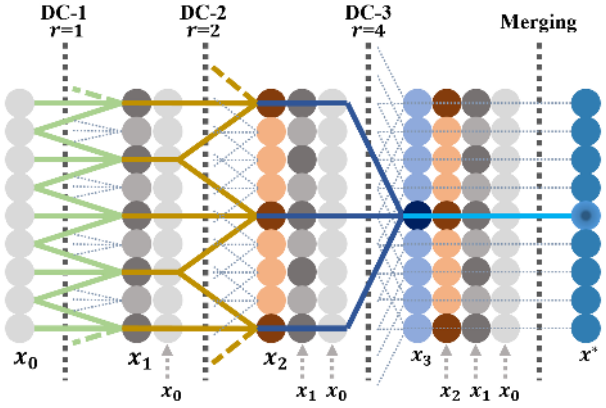



Fig. 3. A simple 1-D example of the DDCM module composed of three DC blocks (kernel size = 3) with dilation rates of 1, 2 and 4. Here we can see that $x_1$ is produced from $x_0$ by a 1-dilated convolution with a receptive field of 3. $x_2$ is produced from $[x_1, x_0]$ by a 2-dilated convolution with fused receptive fields of $[7, 5]$. $x_3$ is produced from $[x_2, x_1, x_0]$ by a 4-dilated convolution with fused receptive fields of $[15, 11, 9]$. The final output $x^*$ is produced from $[x_3, x_2, x_1, x_0]$ by the so-called merging layer, which fuses multi-scale context information by aggregating various receptive fields of $[15, 11, 9, 7, 5, 3, 1]$.

factors, enable DDCM networks to have very large receptive fields with just a few layers. Please note that we apply zero padding to the input of every DC block in order to keep the resolution of its output equal to the resolution of the input.

Fig. 3 illustrates a simple 1-D example of the DDCM module composed of three DC blocks (kernel size equal 3) with dilation rates of $[1, 2, 4]$. In the DC-1 layer, $x_1$ is produced from $x_0$ by a 1-dilated convolution, where each element of $x_1$ has a receptive field of 3. In the DC-2 layer, $x_2$ is produced from $[x_1, x_0]$ by a 2-dilated convolution. Note, the receptive field for the elements of $x_2$ are $[7, 5]$. Similarly, in the DC-3 layer, $x_3$ is produced from $[x_2, x_1, x_0]$ by a 4-dilated convolution with fused receptive fields of $[15, 11, 9]$. The final output $x^*$ is thus produced from $[x_3, x_2, x_1, x_0]$ by the so-called merging layer, which fuses multi-scale context information by aggregating various receptive fields of $[15, 11, 9, 7, 5, 3, 1]$. It is easy to see that the number of parameters associated with each DC layer grows linearly, while the fused receptive field size is nearly exponentially increasing.

### C. Variants of the DDCM module

*1) Grouped convolutions:* Inspired by ResNeXt [41], grouped convolutions [14] are also exploited in the DC blocks in order to further reduce the depth and parameter size of DDCM-net, especially when the DDCM modules are used as the decoders of high-level features. ResNeXt has demonstrated that increasing cardinality (group number) is a more effective way of gaining accuracy than going deeper or wider [41]. We therefore introduce a variant of DDCM, i.e. DDCM$(g = 2)$, where "$g = 2$" denotes the fact that grouped convolutions with 2 groups are used in the DC blocks.

*2) Strided convolutions:* To further reduce the computational cost when increasing dilation rates, we can apply a dilated convolution with a stride of greater than one pixel, which samples only every $s$ pixels in each direction in the output. Here $s$ denotes the stride of this dilated convolution. This is similar to a two-step approach with unit stride convolution followed by downsampling, but reduces computational cost. When using a strided dilated convolution in a DC block, we therefore need to apply bilinear upsampling to scale the output to the same resolution as the input before concatenation. There are three variants of the DDCM-Module evaluated in this work, i.e. DDCM$(s = 2)$, DDCM$(s = 3)$ and DDCM$(s = r + 1)$ with different striding strategies: a stride of 2, a stride of 3 and a dynamic stride of $(r + 1)$ separately, where $r$ denotes the corresponding dilation rate.

### D. The DDCM network

DDCM modules define building blocks from which a more complex network can be built. This is illustrated in Fig. 4 showing the end-to-end pipeline of the DDCM-Net combined with a pre-trained model for land cover classification. Compared to other encoder-decoder architectures, our proposed DDCM-Net only fuses low-level features one time before the final prediction CNN layers, instead of aggregating multi-scale features captured at many different encoder layers [2], [6], [27], [31], [32], [33], [34], [35], [36]. This makes our model simple and neat, yet effective with lower computational cost. In

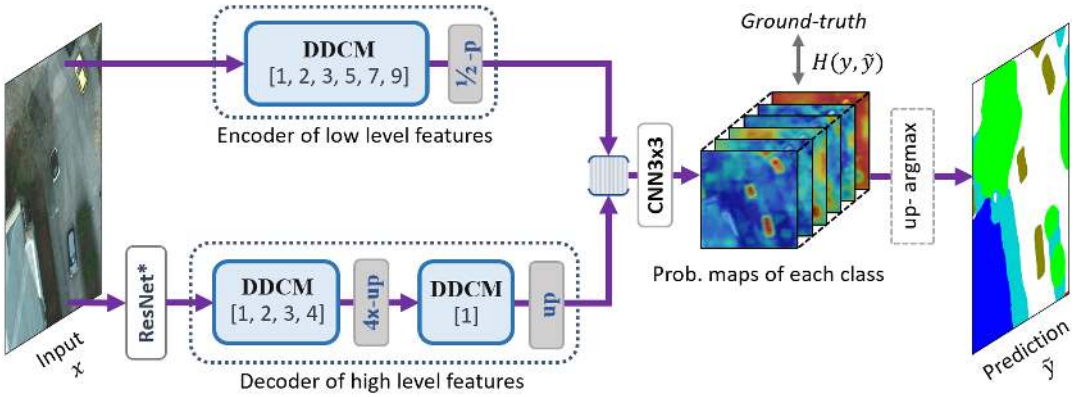



Fig. 4. End-to.end pipeline of DDCM-Net for semantic mapping of VHR Potsdam images. The encoder of low level features encodes multi-scale contextual information from the initial input images by a DDCM module (output 3-channel) using $3 \times 3$ kernels with 6 different dilation rates $[1, 2, 3, 5, 7, 9]$. The decoder of high level features decodes highly abstract representations learned from a ResNet-based backbone (output 1024-channel) by 2 DDCM modules with rates $[1, 2, 3, 4]$ (output 36-channel) and $[1]$ (output 18-channel) separately. The transformed low-level and high-level feature maps by DDCMs are then fused together to infer pixel-wise class probabilities. Here, 'p' and 'up' denote pooling and up-sampling respectively.

particular, this model is easy to adapt by adjusting the density (number of the output feature maps) and dilation strategy of the encoder and/or decoder features to tackle different tasks, depending on different domains.

In our work, we only utilize the first three bottleneck layers of pretrained ResNet-based [42] backbones (both ResNet50 [42] and SE-ResNeXt50 [43]) and remove the last bottleneck layer and the fully connected layers to reduce the number of parameters to train. Furthermore, due to the larger complexity and variety of the DeepGlobe dataset compared to the ISPRS data, we utilize a DDCM($s = 2$) module configured with larger dilation growing rates $[1, 2, 4, 8, 16, 32]$ as the low-level encoder, and two DDCM($g = 2, s = 2$) modules configured by $[1, 2, 4]$ and $[1]$ as the high-level decoder. This configuration results in feature maps of size 64-channel and 32-channel, rather than 36-channel and 18-channel for the model on ISPRS data. We also choose SE-ResNeXt50 as the backbone, instead of ResNet50.

## V. Experiments and Results

In this section, we investigate the proposed network on the Potsdam (Section V-C), Vaihingen (Section V-C) and DeepGlobe (Section V-D) datasets and report both qualitative and quantitative results of multi-class land cover classification.

### A. Training details

According to best practices, we train using Adam [44] with AMSGrad [45] as the optimizer with weight decay $2 \times 10^{-5}$ applied to all learnable parameters except biases and batch-norm parameters, and polynomial learning rate (LR) decay $(1 - \frac{cur\_iter}{max\_iter})^{0.9}$ with the maximum iterations of $10^8$. We also set $2 \times LR$ to all bias parameters in contrast to weights parameters. We use initial LRs of $\frac{8.5 \times 10^{-5}}{\sqrt{2}}$ and $\frac{8.5 \times 10^{-4}}{\sqrt{2}}$ for the ISPRS data and DeepGlobe data, respectively. For the training on ISPRS data (both Potsdam and Vaihingen), we utilized a stepwise LR schedule method that reduces the LR by a factor of 0.85 every 15 epochs based on our training observations and empirical evaluations, while for the training on DeepGlobe data, we utilized multi-step LR policy, which reduces the LR by a factor of 0.56 at epochs $[4, 8, 16, 24, 32, 96, 128]$ guided by our empirical results.

We apply a cross-entropy loss function with median frequency balancing (MFB) weights as defined in the equations 2 and 3 [46].

$$W_c = \frac{\text{median}(\{f_c | c \in \mathcal{C}\})}{f_c}, \tag{2}$$

$$Loss = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} {l_c}^{(n)} \log ({p_c}^{(n)}) W_c \tag{3}$$

where $W_c$ is the weight for class $c$, $f_c$ the pixel- frequency of class $c$, ${p_c}^{(n)}$ is the probability of sample belonging to class $c$, ${l_c}^{(n)}$ denotes the class label of sample $n$ in class $c$.

We train and validate the networks for the Potsdam and Vaihingen datasets with ramdomly sampled 5000 patches of size $256 \times 256$ as input and batch size of five. For the experiments on the DeepGlobe dataset, we use two down-sampled resolutions of $1224 \times 1224$ and $816 \times 816$ (down-scaled from $2448 \times 2448$), then we train with $4000$ crops ($765 \times 765$) and batch size of four. The training data is sampled uniformly and randomly shuffled for each epoch. We conduct all experiments in this paper using PyTorch [47] on a single computer with one NVIDIA 1080Ti GPU.

### B. Augmentation and evaluation methods

During training on Potsdam and Vaihingen data, we randomly flip or mirror images for data augmentation (with probability $0.5$), while on the DeepGlobe data, we also augment the crops by randomly shifting (limit 0.0625), scaling (limit 0.1) and rotating them (limit 10). The albumentations library [48] for data augmentation is utilized in this work. Please note that all training images are normalized to [0.0, 1.0] after data augmentation.

We apply test time augmentation (TTA) in terms of flipping and mirroring. For the Potsdam and Vaihingen data, we use sliding windows (with $448 \times 448$ size at a 100-pixel stride) on

a test image and stitch the results together by averaging the predictions of the over-lapping TTA regions to form the output. While for DeepGlobe data, we first apply TTA on a down-sampled (3x) test image ($816 \times 816$) and then up-sample all the predictions back to original sizes and average them to get the final output. The performance is measured by both the F1-score [46], and the mean Intersection over Union (IoU) [49]. Please note that the mIoU metric was computed by averaging over the six classes (excluding the 'Unknown' class) in the DeepGlobe contest.

### C. Potsdam and Vaihingen

For evaluation, the labeled part of the Potsdam dataset is split into a training set (19 images), a validation set (2 images of 4_10 and 7_10), and a local test set (3 images of areas 5_11, 6_9 and 7_11). The Vaihingen dataset is similarly divided into training (11 images), validation (2 images of areas 7 and 9) and local test set (4 images of areas 5, 15, 21 and 30). While the hold-out test sets contain 14 images (areas: 2_13, 2_14, 3_13, 3_14, 4_13, 4_14, 4_15, 5_13, 5_14, 5_15, 6_13, 6_14, 6_15 and 7_13) and 17 images (areas: 2, 4, 6, 8, 10, 12, 14, 16, 20, 22, 24, 27, 29, 31, 33, 35 and 38) for the Potsdam and Vaihingen datasets, respectively. Table I shows our results on the hold-out test sets and our local test sets of ISPRS Potsdam and Vaihingen separately with a single trained model. The mean F1-score (mF1) and the mean IoU (mIou) are computed as the average measure of all classes except the clutter class.

TABLE I
RESULTS ON THE HOLD-OUT TEST IMAGES OF ISPRS POTSDAM AND VAIHINGEN DATASETS WITH A SINGLE TRAINED DDCM-R50 MODEL SEPARATELY.

| **F1-score** | **OA** | **Surface** | **Building** | **Low-veg** | **Tree** | **Car** | **mF1** |
|---|---|---|---|---|---|---|---|
| Potsdam | 0.908 | 0.929 | 0.969 | 0.877 | 0.894 | 0.949 | 0.923 |
| | 0.931* | 0.946* | 0.983* | 0.865* | 0.892* | 0.939* | 0.925* |
| Vaihingen | 0.904 | 0.927 | 0.953 | 0.833 | 0.894 | 0.883 | 0.898 |
| | 0.921* | 0.934* | 0.973* | 0.814* | 0.914* | 0.909* | 0.909* |
| **IoU** | | | | | | | **mIoU** |
| Potsdam | 0.908 | 0.867 | 0.940 | 0.781 | 0.809 | 0.902 | 0.860 |
| | 0.931* | 0.898* | 0.966* | 0.762* | 0.805* | 0.885* | 0.863* |
| Vaihingen | 0.904 | 0.863 | 0.909 | 0.713 | 0.808 | 0.790 | 0.817 |
| | 0.921* | 0.876* | 0.948* | 0.686* | 0.842* | 0.832* | 0.837* |

* Results marked with * were measured on our local test images, others were measured on hold-out test sets (14 images and 17 images for the Potsdam and Vaihingen separately).

We also compare our results to other related published work on the ISPRS Potsdam RGB dataset and Vaihingen IRRG dataset. These results are shown in Table II and III respectively. Our single model with overall F1-score (92.3%) on Potsdam RGB dataset, achieves around 0.5 percent higher score compared to the second best model - FuseNet+OSM [25]. Similarly, our model trained on Vaihingen IRRG images, also obtained the best overall performance with 89.8% F1-score that is around 1.1% higher than the score of the second best model - GSN [26]. It is also worth noting that, although our OA is only marginal better ($+0.1\%$) for Vaihingen, and even worse ($-1.5\%$) for Potsdam, our model obtained better F1 scores. We therefore believe that our proposed method has better capability to handle extremely unbalanced classes. Further, by balancing and modelling the surrounding classes (such as road/surface and buildings) more accurately with our model, the car class will be easier to distinguish and thus has better results with increased receptive fields as shown in Table III.

Fig. 5 shows the qualitative comparisons of the land cover mapping results from our model and the ground truths on the test set. We observe that our model is able to segment both large multi-scale objects (such as buildings) and small objects (such as cars) very well with fine-gained boundary recovery without any post-processing.

TABLE II
COMPARISONS BETWEEN OUR METHOD WITH OTHER PUBLISHED METHODS ON THE HOLD-OUT RGB TEST IMAGES OF ISPRS POTSDAM DATASET.

| **Models** | **OA** | **Surface** | **Building** | **Low-veg** | **Tree** | **Car** | **mF1** |
|---|---|---|---|---|---|---|---|
| HED+SEG.H-Sc1 [23] | 0.851 | 0.850 | 0.967 | 0.842 | 0.686 | 0.858 | 0.846 |
| RiFCN [27] | 0.883 | 0.917 | 0.930 | 0.837 | 0.819 | 0.937 | 0.861 |
| RGB+I-ensemble [28] | 0.900 | 0.870 | 0.936 | 0.822 | 0.845 | 0.892 | 0.873 |
| Hallucination [28] | 0.901 | 0.873 | 0.938 | 0.821 | 0.848 | 0.882 | 0.872 |
| DNN_HCRF [29] | 0.884 | 0.912 | 0.946 | 0.851 | 0.851 | 0.928 | 0.898 |
| SegNet RGB [25] | 0.897 | 0.930 | 0.929 | 0.850 | 0.851 | 0.951 | 0.902 |
| DST_2 [21] | 0.903 | 0.925 | 0.964 | 0.867 | 0.880 | 0.947 | 0.917 |
| FuseNet+OSM [25] | **0.923** | **0.953** | 0.959 | 0.863 | 0.851 | **0.968** | 0.918 |
| **Ours** | | | | | | | |
| DDCM-R50 | 0.908 | 0.929 | **0.969** | **0.877** | **0.894** | 0.949 | **0.923** |
| | (-1.5%) | (-2.4%) | (+0.5%) | (+1.0%) | (+1.4%) | (-1.9%) | (+0.5%) |
| DDCM($s = 2$) | 0.908 | 0.930 | 0.968 | 0.876 | 0.895 | **0.952** | **0.924** |
| DDCM($s = 3$) | 0.910 | 0.932 | 0.967 | 0.878 | 0.895 | 0.937 | 0.922 |
| DDCM($s = r + 1$) | 0.911 | 0.933 | 0.968 | 0.876 | 0.894 | 0.950 | 0.924 |

TABLE III
COMPARISONS BETWEEN OUR METHOD WITH OTHER PUBLISHED METHODS ON THE HOLD-OUT IRRG TEST IMAGES OF ISPRS VAIHINGEN DATASET.

| **Models** | **OA** | **Surface** | **Building** | **Low-veg** | **Tree** | **Car** | **mF1** |
|---|---|---|---|---|---|---|---|
| UOA [22] | 0.876 | 0.898 | 0.921 | 0.804 | 0.882 | 0.820 | 0.865 |
| DNN_HCRF [29] | 0.878 | 0.901 | 0.932 | 0.814 | 0.872 | 0.720 | 0.848 |
| ADL_3 [20] | 0.880 | 0.895 | 0.932 | 0.823 | 0.882 | 0.633 | 0.833 |
| DST_2 [21] | 0.891 | 0.905 | 0.937 | 0.834 | 0.892 | 0.726 | 0.859 |
| ONE_7 [24] | 0.898 | 0.910 | 0.945 | **0.844** | 0.899 | 0.778 | 0.875 |
| DLR_9 [23] | 0.903 | 0.924 | 0.952 | 0.839 | 0.899 | 0.812 | 0.885 |
| GSN [26] | 0.903 | 0.922 | 0.951 | 0.837 | **0.899** | 0.824 | 0.887 |
| **Ours** | | | | | | | |
| DDCM-R50 | **0.904** | **0.927** | **0.953** | 0.833 | 0.894 | **0.883** | **0.898** |
| | (+0.1%) | (+0.3%) | (+0.1%) | (-1.1%) | (-0.5%) | (+5.9%) | (+1.1%) |
| DDCM($s = 2$) | 0.901 | 0.924 | 0.951 | 0.826 | 0.891 | **0.890** | 0.896 |
| DDCM($s = 3$) | 0.901 | 0.923 | 0.951 | 0.828 | 0.891 | 0.879 | 0.894 |
| DDCM($s = r + 1$) | 0.901 | 0.923 | 0.949 | 0.829 | 0.892 | 0.888 | 0.896 |

TABLE IV
PERFORMANCE COMPARISONS ON THE HOLD-OUT VALIDATION SET OF DEEPGLOBE DATA WITH OTHER PUBLISHED METHODS.

| **Models** | **mIoU** | **GFLOPs** |
|---|---|---|
| NU-Net [30] | 0.428 | - |
| InceptionV3+Haralick [31] | 0.476 | - |
| GCN-based [32] | 0.485 | - |
| FPN [33] | 0.493 | - |
| Stacked U-Nets [34] | 0.507 | - |
| DeepLabv3+ [35] | 0.510 | - |
| ClassmateNet [36] | 0.519 | - |
| DFCNet [36] | 0.526 | - |
| Deep Aggregation Net [35] | 0.527 | - |
| **Ours** | | |
| DDCM-SER50 | **0.562** | **4.68** |

### D. DeepGlobe

Our DDCM-SER50 model achieves new state-of-the-art result with 56.2% mIoU on DeepGlobe land cover classification challenge dataset. As shown in Table IV, we compare our DDCM network with other published models ([30], [31], [32], [33], [34], [35], [36]) on the hold-out validation set (the public leaderboard [3] up to the date of May 1, 2019). Our

[3]https://competitions.codalab.org/competitions/18468#results

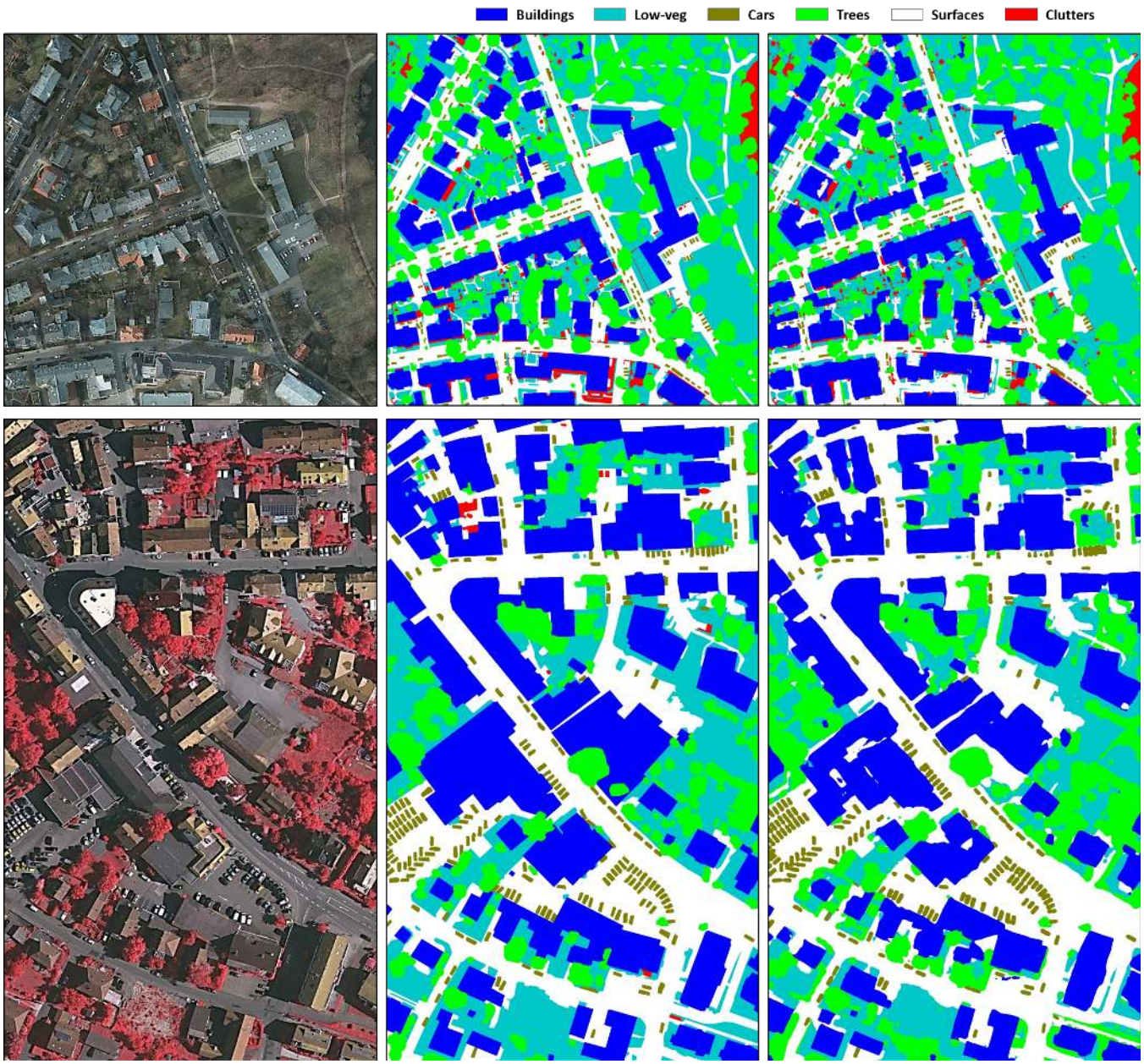



Fig. 5. Mapping results for test images of Potsdam tile-3_14 (top) and Vaihingen tile-27 (bottom). From the left to right, the input images (left), the ground truths (middle) and the predictions of our single DDCM-R50 model.

model obtained above 3.5% higher mIoU than the second best model [35]. Fig. 6 shows some examples of the predictions on the hold-out validation images. We didn't apply any post-processing (e.g. graph-based fine tuning as utilized in [35]), nor any multi-scale prediction fusion methods as applied in [36]. We just make the prediction on the $3\times$ down scaling of the test image and then perform up-sampling back to its original resolution.

## VI. Discussion

### A. Preliminary analysis

As a baseline, we re-implemented, trained and evaluated some popular architectures on the local Potsdam test set [49]. We compared our methods to them in terms of parameters

TABLE V
Quantitative Comparison of parameters size, FLOPs (measured on input image size of $3 \times 256 \times 256$), Inference time on CPU and GPU separately, and mIoU on ISPRS Potsdam RGB dataset.

| Models | Backbones | Parameters (Million) | FLOPs (Giga) | Inference time (ms - CPU/GPU) | mIoU* |
|---|---|---|---|---|---|
| U-Net [2] | VGG16 | 31.04 | 15.25 | 1460 / **6.37** | 0.715 |
| FCN8s [1] | VGG16 | 134.30 | 73.46 | 6353 / 20.68 | 0.728 |
| SegNet [3] | VGG19 | 39.79 | 60.88 | 5757 / 15.47 | 0.781 |
| GCN [6] | ResNet50 | 23.84 | 5.61 | 593 / 11.93 | 0.774 |
| PSPNet [4] | ResNet50 | 46.59 | 44.40 | 2881 / 81.08 | 0.789 |
| DUC [5] | ResNet50 | 30.59 | 32.26 | 2086 / 68.24 | 0.793 |
| **Ours** | | | | | |
| DDCM-R50 | ResNet50† | **9.99** | **4.86** | **238** / 10.23 | **0.808** |
| DDCM($s = 2$) | ResNet50† | 9.99 | 4.48 | 159 / 11.39 | **0.811** |
| DDCM($s = 3$) | ResNet50† | 9.99 | 4.43 | 144 / 11.25 | 0.798 |
| DDCM($s = r + 1$) | ResNet50† | 9.99 | 4.42 | **132** / 11.50 | 0.810 |

* mIoU was measured on full reference ground truths of our local test images 5_11, 6_9 and 7_11 in order to fairly compare with our previous work [49].
* Inference time was measured on CPU - AMD Ryzen Threadripper 1950X and GPU - NVIDIA GeForce GTX 1080Ti respectively.

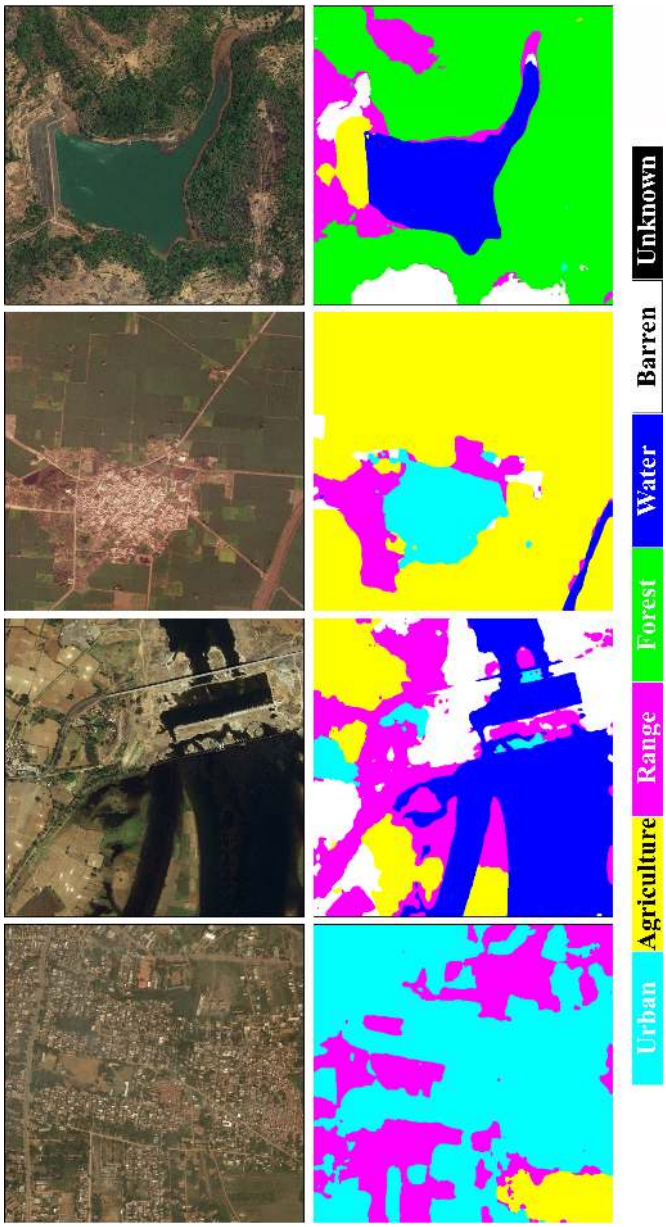



Fig. 6. Mapping results on hold-out validation images of DeepGlobe. From the left to right, the input satellite images and the predictions of our model.

TABLE VI
ABLATION STUDIES FOR OUR PROPOSED METHOD ON THE HOLD-OUT RGB TEST IMAGES OF THE ISPRS POTSDAM DATASET.

| Models | OA | Surface | Building | Low-veg | Tree | Car | mF1 |
|---|---|---|---|---|---|---|---|
| DDCM-R50 | **0.908** | **0.929** | **0.969** | **0.877** | **0.894** | **0.949** | **0.923** |
| No-LL-Encoder | 0.899 (-0.9%) | 0.919 (-1.0%) | 0.948 (-2.1%) | 0.869 (-0.8%) | 0.893 (-0.1%) | 0.936 (-1.3%) | 0.913 (-1.0%) |
| No-Dilation | 0.892 (-1.6%) | 0.910 (-1.9%) | 0.938 (-3.1%) | 0.868 (-0.9%) | 0.892 (-0.2%) | 0.929 (-2.0%) | 0.908 (-1.5%) |

* No-LL-Encoder means the model was configured without the low-level encoder stream, while adjusting the output channels of the high level decoder to 21 instead of 18 in the standard DDCM-R50 model. The reason for increasing the number of the high level channels is to counter the loss of features from the low-level encoder stream.
* No-Dilation means that each DDCM module in the DDCM-R50 model only used unit dilation rates for its convolution layers.

sizes, computational cost (FLOPs), inference time on both CPU (AMD Ryzen Threadripper 1950X) and GPU (NVIDIA GeForce GTX 1080Ti), and mIoU evaluated on the full reference ground truths of the dataset. Table V details the quantitative results of our DDCM-R50 model against others. Our model consumes about 9x and 13x less FLOPs with 4x and 4.7x fewer parameters and 12x and 24x faster inference speed on CPU, but achieves +1.9% and +2.7% higher mIoU than PSPNet [4] and SegNet [3] respectively.

Additionally, we also investigated the effectiveness of strided convolutions with the purpose of reducing the computational cost of dilated modules. We observe that with a dynamic stride of $r+1$, our model has the best speed without loss of the F1-scores in comparison to a stride of 2 or stride-1 model as shown the final hold-out tests as shown in Table II. And interestingly, we find that both dynamic strided and stride-2 policies could improve the models accuracy of small objects (i.e cars). Overall, DDCM$(s=2)$ based mode obtained the best performance on small objects (cars) with $+0.3\%$ and $+0.7\%$ higher IoU than standard convolutions on both the Potsdam and Vaihingen test datasets, respectively, as shown in Tables II and III. We therefore believe that dilated convolutions with strided operations (i.e a stride of 2) could not only improve a model's computational efficiency, but also capture better contextual representations that further boost the model's capability for detailed object boundary recovery.

### *B. Flexible dilation and density policies*

We used two different dilation settings strategies. For ISPRS data, we configured a DDCM module with linearly growing dilation rates ($[1,2,3,5,7,9]$) as the low-level features encoder, while for the DeepGlobe dataset, we build the encoder with exponentially growing dilation rates ($[1,2,4,8,16,32]$) with a stride of two, since we see that the DeepGlobe images contain more spatially chaotic objects with lower resolutions, larger scales and less geometrical attributes than the ISPRS images. We believe DDCM modules with bigger dilation configurations could capture larger multi-scale and global context in this case, but require more computational cost. Hence, one has to make some trade-offs on the dilation policies and the densities based both on the dataset and on the budget of computation resources. Similarly, as for the decoders of high-level features, we also follow different strategies in terms of the dilation settings and densities of DDCM modules for the ISPRS and DeepGlobe data as described in Section IV-D. In particular, we adopt both strided and grouped convolutions together that use stride equal to two and group equal to two in the DDCM modules with the output densities of 64-channel and 32-channel respectively. These settings strike the best trade-off between speed and accuracy on the DeepGlobe database in our experiments.

We also evaluated the the influence of the low-level feature encoder and the dilated strategies. We performed two ablation studies, by training the following two models: 1) The No-LL-Encoder model that was configured without the low-level encoder stream, but only used the high level decoder branch which output 21 channels instead of 18 channels in the standard DDCM-R50 model; 2) The No-Dilation model that only

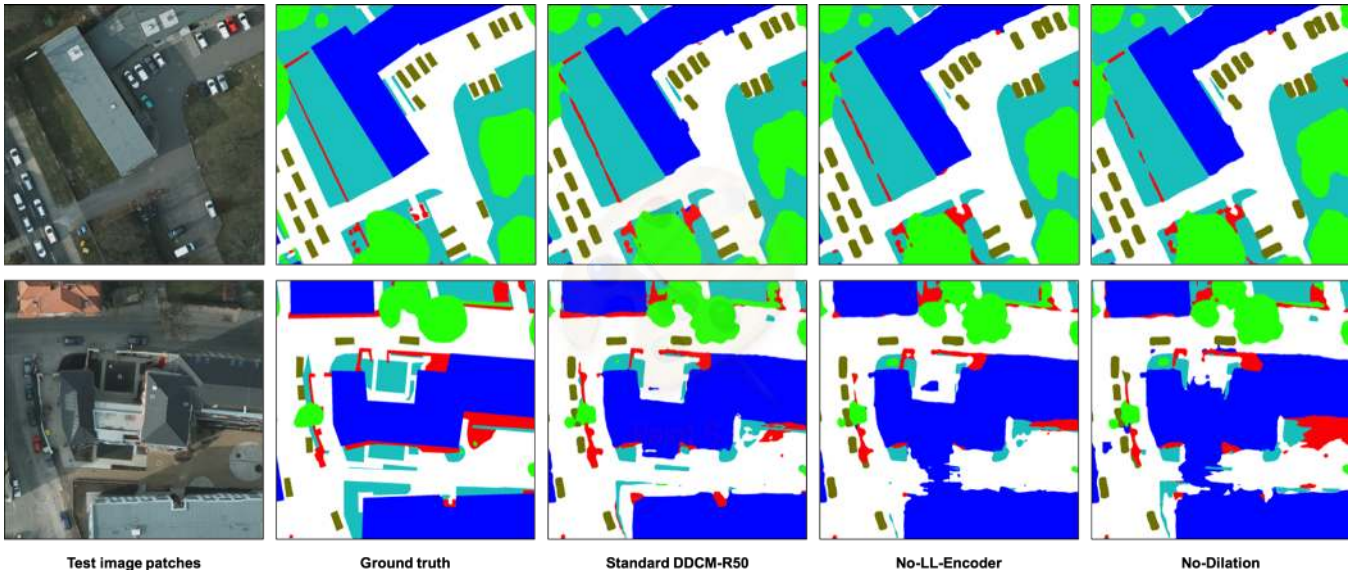



Fig. 7. Mapping results comparison. From the left to right, the input image patches, the ground truth (left), the standard DDCM-R50, the no-low-level-encoder DDCM-R50, and the no-dilation DDCM-R50 (right).

uses standard convolutions by fixing the dilation rate $r = 1$ for each of the convolutional layers in the DDCM-R50 model. Table VI shows the final test results on the Potsdam dataset. The performance, in terms of mF1, dropped off overall 1.0% and 1.5% with No-LL-Encoder and No-Dilation, respectively. Performance is particularly decreasing for the building (-2.1% and -3.1%) and car (-1.3% and -2.0%) classes. There findings are also validated by the qualitative visualization of the results that is shown in Fig. 7. We therefore consider that the low-level features encoder and merged dilated convolutions can obtain better local and global context information and thus segment both multi-scale objects (such as buildings and surface) and small classes (such as cars) accurately.

### C. Generalization

Our models demonstrated very good generalization capabilities on both the Potsdam and Vaihingen dataset. As shown in Table I, there are only -0.2% and -1.1% gaps in terms of mean F1-score between our local validation sets and the hold-out test sets of the Potsdam and Vaihingen, respectively. It is also worth noting that our model is the only one that works equally well on both Vaihingen IRRG dataset and Potsdam RGB dataset, which outperforms the DST_2 [21] model with 3.9% and 0.6% higher F1-score on Vaihingen and Potsdam dataset, respectively, as shown in Tables II and III. Furthermore, our model achieves better performance (+0.5%) in terms of mean F1-score with fewer labeled training data than FuseNet+OSM [25] that used OpenStreetMap (OSM) as an additional data source.

However, we observed a bigger performance drop (approximately -15.9%) on the DeepGlobe dataset when comparing results on the hold-out test set (mIoU: 56.2%) and local validation results (K-Fold avg. mIoU: 72.1%) as shown in Table VII. We see there is higher uncertainty (more false predictions) between range land (magenta), agriculture land (yellow), and forest (green) from the confusion matrix between classes for our model in Table VIII. Note that the model incorrectly classified some agriculture land to rangeland, and predicted some rangeland as forest. These observations are also supported by the qualitative visualization of errors of our predictions as shown in Fig. 8. Furthermore, in our experiments on the DeepGlobe data, we found that there are some annotation inaccuracies, mainly introduced by highly ambiguous objects and lower ground resolutions. What is worse, we observed that the hold-out test images have different contrast and darker shadows than in the training set [30]. This obviously affected the model's final performance on the test sets.

TABLE VII
IOU SCORES OF OUR 5 K-FOLD MODELS ON LOCAL VALIDATION SETS OF DEEPGLOBE DATASET.

| K-Fold | Urban | Agriculture | Range | Forest | Water | Barren | mIoU |
|---|---|---|---|---|---|---|---|
| k0 | 0.783 | **0.901** | **0.488** | 0.760 | 0.605 | 0.723 | 0.710 |
| k1 | 0.735 | 0.876 | 0.391 | 0.772 | 0.730 | **0.739** | 0.707 |
| k2 | **0.821** | 0.873 | 0.381 | 0.757 | 0.832 | 0.723 | 0.731 |
| k3 | 0.796 | 0.883 | 0.421 | 0.789 | 0.849 | 0.716 | **0.742** |
| k4 | 0.724 | 0.830 | 0.431 | **0.795** | **0.857** | 0.654 | 0.715 |
| **Avg.** | **0.772** | **0.872** | **0.422** | **0.775** | **0.775** | **0.711** | **0.721** |

## VII. CONCLUSIONS

In this paper, we presented a dense dilated convolutions merging (DDCM) architecture for land cover classification for aerial imagery. The proposed architecture applies dilated convolutions to learn features at varying dilation rates, and merges the feature map of each layer with the feature maps from all previous layers. On both the Potsdam and Vahingen datasets, our single model based on the DDCM-Net architecture achieves the best mean F1-score compared to the other architectures, but with much fewer parameters and feature maps. DDCM-Net is easy to adapt to address a wide range of different problems, is fast to train, and achieves accurate

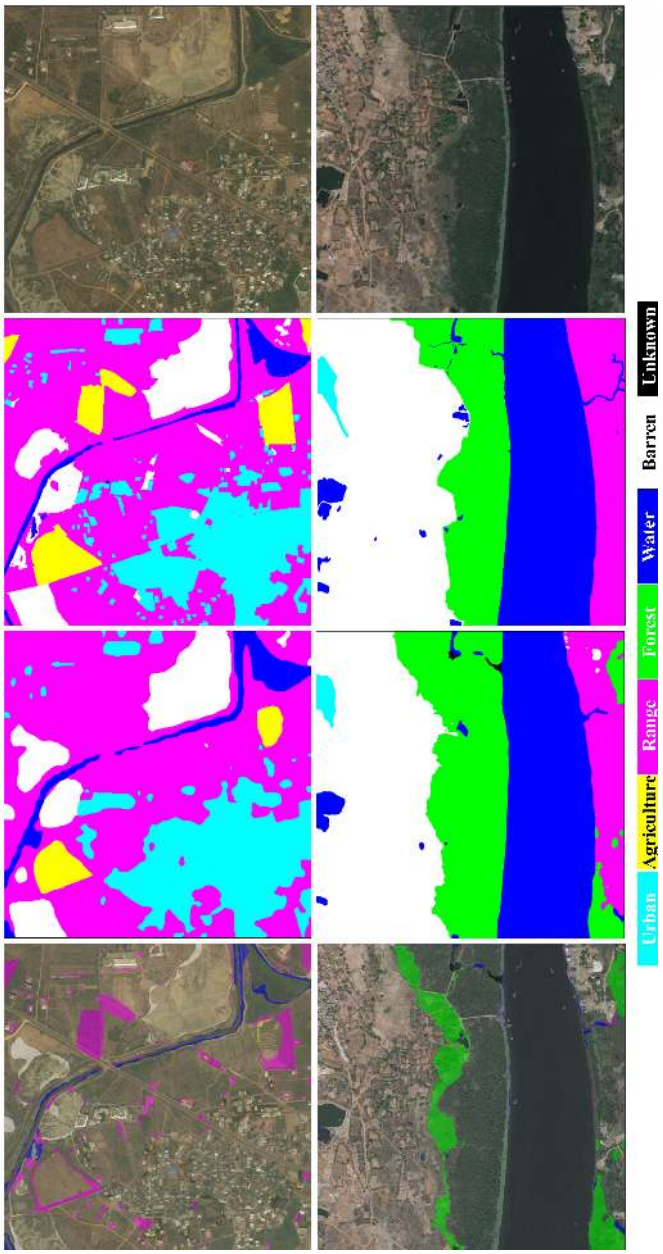


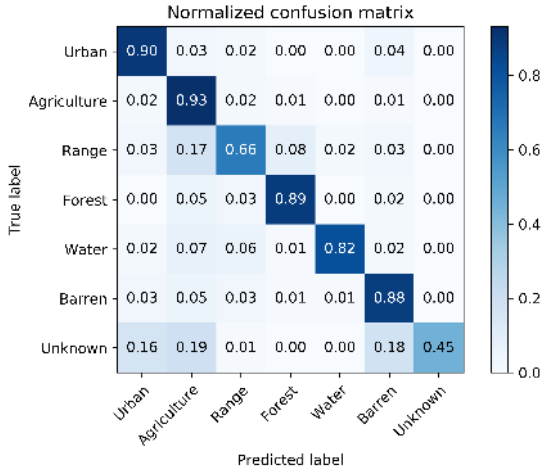

Fig. 8. Mapping results on local validation images of DeepGlobe. From top to bottom, the input satellite images, the ground truths, the predictions of DDCM-SER50 model, and the errors of predictions.

results even on small datasets. The variants of our DDCM-Net by using different combinations of dilations and densities for the DeepGlobe dataset also demonstrated better performance, but consumed much fewer computation resources compared to other published methods.

## ACKNOWLEDGMENT

This work is supported by the foundation of the Research-Council of Norway under Grant 220832. The authors would also like to thank the ISPRS for making the Potsdam and Vaihingen datasets publicly available.

TABLE VIII
NORMALIZED CONFUSION MATRIX ON A LOCAL VALIDATION SET OF DEEPGLOBE DATA.




## REFERENCES

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[3] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, 2017.

[4] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2881–2890.

[5] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018, pp. 1451–1460.

[6] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters–improve semantic segmentation by global convolutional network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4353–4361.

[7] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep Networks with Stochastic Depth," in *European Conference on Computer Vision*. Springer, 2016, pp. 646–661.

[8] I. S. for Photogrammetry and R. S. (ISPRS), "2D Semantic Labeling Contest," online, 2018. [Online]. Available: http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html

[9] I. Demir, K. Koperski, D. Lindenbaum, G. Pang, J. Huang, S. Basu, F. Hughes, D. Tuia, and R. Raskar, "DeepGlobe 2018: A challenge to parse the earth through satellite images," in *the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2018.

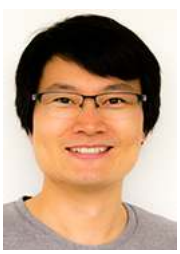
[10] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2018.

[11] D. M. Pelt and J. A. Sethian, "A mixed-scale dense convolutional neural network for image analysis," *Proceedings of the National Academy of Sciences*, vol. 115, no. 2, pp. 254–259, 2018.

[12] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[13] Q. Liu, M. Kampffmeyer, R. Jenssen, and A.-B. Salberg, "Dense dilated convolutions merging network for semantic mapping of remote sensing images," *2019 Joint Urban Remote Sensing Event (JURSE)*, May 2019. [Online]. Available: http://dx.doi.org/10.1109/JURSE.2019.8809046

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
[15] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning Hierarchical Features for Scene Labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, 2013.
[16] N. Audebert, B. Le Saux, and S. Lefèvre, "Segment-before-detect: Vehicle detection and classification through semantic segmentation of aerial images." *Remote Sensing*, vol. 9, no. 4, 2017.
[17] R. Schuster, O. Wasenmuller, C. Unger, and D. Stricker, "SDC-Stacked dilated convolution: A unified descriptor network for dense matching tasks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2556–2565.
[18] E. Strubell, P. Verga, D. Belanger, and A. McCallum, "Fast and accurate entity recognition with iterated dilated convolutions," *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2670–2680, 2017.
[19] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.
[20] S. Paisitkriangkrai, J. Sherrah, P. Janney, V.-D. Hengel *et al.*, "Effective semantic pixel labelling with convolutional networks and conditional random fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2015, pp. 36–43.
[21] J. Sherrah, "Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery," *CoRR*, vol. abs/1606.02585, 2016. [Online]. Available: http://arxiv.org/abs/1606.02585
[22] G. Lin, C. Shen, A. Van Den Hengel, and I. Reid, "Efficient piecewise training of deep structured models for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3194–3203.
[23] D. Marmanis, K. Schindler, J. D. Wegner, S. Galliani, M. Datcu, and U. Stilla, "Classification with an edge: Improving semantic image segmentation with boundary detection," *CoRR*, vol. abs/1612.01337, 2016. [Online]. Available: http://arxiv.org/abs/1612.01337
[24] N. Audebert, B. Le Saux, and S. Lefèvre, "Semantic segmentation of earth observation data using multimodal and multi-scale deep networks," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 180–196.
[25] N. Audebert, B. Le Saux, and S. Lefèvre, "Joint learning from earth observation and openstreetmap data to get faster better semantic maps," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 67–75.
[26] H. Wang, Y. Wang, Q. Zhang, S. Xiang, and C. Pan, "Gated convolutional neural network for semantic segmentation in high-resolution images," *Remote Sensing*, vol. 9, no. 5, p. 446, 2017.
[27] L. Mou and X. X. Zhu, "RiFCN: Recurrent network in fully convolutional network for semantic segmentation of high resolution remote sensing images," *CoRR*, vol. abs/1805.02091, 2018. [Online]. Available: http://arxiv.org/abs/1805.02091
[28] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, "Urban land cover classification with missing data modalities using deep convolutional neural networks," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 6, pp. 1758–1768, 2018.
[29] Y. Liu, S. Piramanayagam, S. T. Monteiro, and E. Saber, "Semantic segmentation of multisensor remote sensing imagery with deep convnets and higher-order conditional random fields," *Journal of Applied Remote Sensing*, vol. 13, no. 1, p. 016501, 2019.
[30] M. Samy, K. Amer, K. Eissa, M. Shaker, and M. ElHelw, "NU-Net: Deep residual wide field of view convolutional neural network for semantic segmentation," in *the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 267–2674.
[31] A. Davydow, O. Neuromation, and S. Nikolenko, "Land cover classification with superpixels and jaccard index post-optimization," in *the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 280–2804.
[32] G. Pascual, S. Seguí, and J. Vitria, "Uncertainty gated network for land cover segmentation." in *the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 276–279.
[33] S. Seferbekov, V. Iglovikov, A. Buslaev, and A. Shvets, "Feature pyramid network for multi-class land segmentation," in *the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 272–275.
[34] A. Ghosh, M. Ehrlich, S. Shah, L. Davis, and R. Chellappa, "Stacked U-Nets for Ground Material Segmentation in Remote Sensing Imagery," in *the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 252–2524.
[35] T.-S. Kuo, K.-S. Tseng, J.-W. Yan, Y.-C. Liu, and Y.-C. Frank Wang, "Deep aggregation net for land cover classification," in *the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 252–256.
[36] C. Tian, C. Li, and J. Shi, "Dense fusion classmate network for land cover classification," in *the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 262–2624.
[37] Y. Li, X. Zhang, and D. Chen, "CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1091–1100.
[38] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T. S. Huang, "Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7268–7277.
[39] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.
[40] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.
[41] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1492–1500.
[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[43] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *CoRR*, vol. abs/1709.01507, 2017. [Online]. Available: http://arxiv.org/abs/1709.01507
[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980
[45] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of Adam and beyond," *arXiv preprint arXiv:1904.09237*, 2018.
[46] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, "Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2016, pp. 1–9.
[47] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
[48] A. Buslaev, A. Parinov, E. Khvedchenya, V. I. Iglovikov, and A. A. Kalinin, "Albumentations: fast and flexible image augmentations," *arXiv preprint arXiv:1809.06839*, 2018.
[49] Q. Liu, A. Salberg, and R. Jenssen, "A comparison of deep learning architectures for semantic mapping of very high resolution images," in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, July 2018, pp. 6943–6946.



**Qinghui Liu** received the Master's degrees in systems engineering with embedded systems from the University of Southeast Norway and in Control Theory and Control Engineering from Shanghai Jiao Tong University, in 2017 and 2003, respectively. Since August 2017, he has been with the Norwegian Computing Center, Oslo, where he is working toward the Ph.D degree in machine learning cooperated with UiT The Arctic University of Norway, Tromsø, Norway. His research interests focus on deep learning methodologies for remote sensing.

**Michael Kampffmeyer** is an Associate Professor in the Machine Learning Group at UiT The Arctic University of Norway, Tromsø, Norway, where he received his PhD in 2018. He has been a visiting researcher at Carnegie Mellon University (September 2017 to July 2018) and the Technical University of Berlin (September to December 2019). His research interests include the development of unsupervised deep learning methods for representation learning and clustering by utilizing ideas from kernel machines and information theoretic learning. Further, he is interested in computer vision, especially related to remote sensing and health applications. For more details visit https://sites.google.com/view/michaelkampffmeyer/.

**Robert Jenssen** is Professor and Head of the Machine Learning Group at UiT The Arctic University of Norway: `machine-learning.uit.no`. He is also an Adjunct Professor at the Norwegian Computing Center in Oslo, Norway. Jenssen received the Dr. Scient (PhD) degree from UiT in 2005. He has had long-term research stays at the University of Florida, at the Technical University of Berlin, and at the Technical University of Denmark. Jenssen is a member of the IEEE Technical Committee on Machine Learning for Signal Processing, a member of the Governing Board of IAPR, and an Associate Editor for the journal Pattern Recognition. Jenssen's research interests are at the intersection of deep learning and kernel machines, with applications in industry, in data-driven health technology, and in remote sensing.

**Arnt-Børre Salberg** received the Diploma in applied physics and Dr. Scient degree in physics from the University of Tromsø, Norway, in 1998 and 2003, respectively. He is currently a senior research scientist in earth observation with the Norwegian Computing Center, Oslo, Norway. From February 2003 to December 2005, he had a Postdoctoral and research position with the Institute of Marine Research, Tromsø. From December 2005 to October 2008 he was head of R&D at Dolphiscan AS, Moelv, Norway. Since October 2008 he has been with Norwegian Computing Center. From August 2001 to June 2002, he was a Visiting Researcher at the U.S. Army Research Laboratory, Adelphi, MD. His research interests are in the area of earth observation, computer vision, machine learning, and statistics.