This CVPR 2020 workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Multi-view Self-Constructing Graph Convolutional Networks with Adaptive Class Weighting Loss for Semantic Segmentation

Qinghui Liu[1,2], Michael Kampffmeyer[2], Robert Jenssen[2,1], Arnt-Børre Salberg[1]
[1]Norwegian Computing Center, Oslo, NO-0314, Norway
[2]UiT Machine Learning Group, UiT the Arctic University of Norway, Tromsø, Norway
liu@nr.no, {michael.c.kampffmeyer, robert.jenssen}@uit.no, salberg@nr.no

Abstract

We propose a novel architecture called the Multi-view Self-Constructing Graph Convolutional Networks (MSCG-Net) for semantic segmentation. Building on the recently proposed Self-Constructing Graph (SCG) module, which makes use of learnable latent variables to self-construct the underlying graphs directly from the input features without relying on manually built prior knowledge graphs, we leverage multiple views in order to explicitly exploit the rotational invariance in airborne images. We further develop an adaptive class weighting loss to address the class imbalance. We demonstrate the effectiveness and flexibility of the proposed method on the Agriculture-Vision challenge dataset [1] and our model achieves very competitive results (0.547 mIoU) [2] with much fewer parameters and at a lower computational cost compared to related pure-CNN based work.

1. Introduction

Currently, the end-to-end semantic segmentation models are mostly inspired by the idea of fully convolutional networks (FCNs) [14] that generally consist of an encoder-decoder architecture. To achieve higher performance, CNN-based end-to-end methods normally rely on deep and wide multi-scale CNN architectures to create a large receptive field in order to obtain strong local patterns, but also capture long range dependencies between objects of the scene. However, this approach for modeling global context relationships is highly inefficient and typically requires a large number of trainable parameters, considerable computational resources, and large labeled training datasets.

Recently, graph neural networks (GNNs) [1] and Graph Convolutional Networks (GCNs) [8] have received increasing attention and have been applied to, among others, image classification [10], few-shot and zero-shot classification [4], point clouds classification [16] and semantic segmentation [11]. However, these approaches are quite sensitive to how the graph of relations between objects is built and previous approaches commonly rely on manually built graphs based on prior knowledge [11]. In order to address this problem and learn a latent graph structure directly from 2D feature maps for semantic segmentation, the Self-Constructing Graph module (SCG) [13] was recently proposed and has obtained promising results.

In this work, we extend the SCG to explicitly exploit the rotation invariance in airborne images, by extending it to consider multiple views. More specifically, we augment the input features to obtain multiple rotated views and fuses the multi-view global contextual information before projecting the features back onto the 2-D spatial domain. We further propose a novel adaptive class weighting loss that addresses the issue of class imbalance commonly found in semantic segmentation datasets. Our experiments demonstrate that

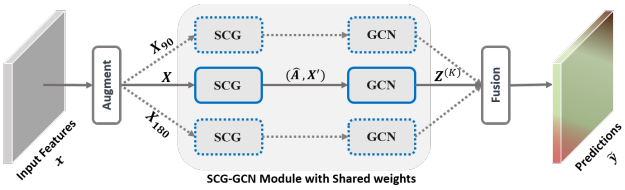

Figure 1. Overview of the MSCG-Net. The Self-Constructing Graph module (SCG) learns to transform a 2D feature map into a latent graph structure and assign pixels (X') to the vertices of the graph ($\hat{A}$). The Graph Convolutional Networks (GCN) is then exploited to update the node features ($Z^{(K)}$, K here denotes the number of layer of GCN) along the edges of graph. The combined module of SCG and GCN (SCG-GCN) can takes augmented multi-view input features (X, X_{90} and X_{180}, where the index indicates degree rotation) and finally the updated multi-view representations are fused and projected back onto 2D maps.

[1] https://www.agriculture-vision.com/dataset
[2] The leaderboard at: https://competitions.codalab.org/competitions/23732?secret_key=dba10d3a-a676-4c44-9acf-b45dc92c5fcf#results

the MSCG-Net achieves very robust and competitive results on the Agriculture-Vision challenge dataset, which is a subset of the Agriculture-Vision dataset [2].

The rest of the paper is organized as follows. In the method Section 2, we present the methodology in details. Experimental procedure and evaluation of the proposed method is performed in Section 3. Finally in Section 4, we draw conclusions.

2. Methods

In this section, we briefly present graph convolutional networks and the self-constructing graph (SCG) approach that are the foundation of our proposed model, before presenting our end-to-end trainable Multi-view SCG-Net (MSCG) for semantic labeling tasks with the proposed adaptive class weighting loss.

2.1. Graph Convolutional Networks

Graph Convolutional Networks (GCNs) [8] are neural networks designed to operate on and extract information from graphs and were originally proposed for the task of semi-supervised node classification. $G = (A, X)$ denotes an undirected graph with n nodes, where $A \in \mathbb{R}^{n \times n}$ is the adjacency matrix and $X \in \mathbb{R}^{n \times d}$ is the feature matrix. At each layer, the GCN aggregates information in one-hop neighborhoods, more specifically, the representation at layer $l+1$ is computed as

$$Z^{(l+1)} = \sigma\left(\hat{A} Z^{(l)} \theta^{(l)}\right) , \tag{1}$$

where $\theta^{(l)} \in \mathbb{R}^{d \times f}$ are the weights of the GCN, $Z^{(0)} = X$, and $\hat{A}$ is the symmetric normalization of A including self-loops [13]:

$$\hat{A} = D^{-\frac{1}{2}}(A + I)D^{\frac{1}{2}} , \tag{2}$$

where $D_{ii} = \sum_j (A + I)_{ij}$ is the degree matrix, I is the identity matrix, and σ denotes the non-linearity function (e.g. $ReLU$).

Note, in the remainder of the paper, we use $Z^{(K)} = \mathrm{GCN}(A, X)$ to denote the activations after a K-layer GCN. However, in practice the GCN could be replaced by alternative graph neural network modules that perform K steps of message passing based on some adjacency matrix A and input node features X.

2.2. Self-Constructing Graph

The Self-Constructing Graph (SCG) module [13] allows the construction of undirected graphs, capturing relations across the image, directly from feature maps, instead of relying on prior knowledge graphs. It has achieved promising performance on semantic segmentation tasks in remote sensing and is efficient with respect to the number of trainable parameters, outperforming much larger models. It is inspired by variational graph auto-encoders [9]. A feature map $X \in \mathbb{R}^{h \times w \times d}$ consisting of high-level features, commonly produced by a CNN, is converted to a graph $G = (\hat{A}, X')$. $X' \in \mathbb{R}^{n \times d}$ are the node features, where $n = h' \times w'$ denotes the number of nodes and where $(h' \times w') \leq (h \times w)$. Parameter-free pooling operations, in our case adaptive average pooling, are employed to reduce the spatial dimensions of X to h' and w', followed by a reshape operation to obtain X'. $\hat{A} \in \mathbb{R}^{n \times n}$ is the learned weighted adjacency matrix.

The SCG module learns a mean matrix $\boldsymbol{\mu} \in \mathbb{R}^{n \times c}$ and a standard deviation matrix $\boldsymbol{\sigma} \in \mathbb{R}^{n \times c}$ of a Gaussian using two single-layer convolutional networks. Note, following convention with variational autoencoders [7], the output of the model for the standard deviation is $\log(\sigma)$ to ensure stable training and positive values for σ. With help of reparameterization, the latent embedding Z is $Z \leftarrow \boldsymbol{\mu} + \boldsymbol{\sigma} \cdot \boldsymbol{\varepsilon}$ where $\varepsilon \in \mathbb{R}^{N' \times C}$ is an auxiliary noise variable and initialized from a standard normal distribution ($\boldsymbol{\varepsilon} \sim \mathrm{N}(0, I)$). A centered isotropic multivariate Gaussian prior distribution is used to regularize the latent variables, by minimizing a Kullback-Leibler divergence loss

$$\mathcal{L}_{kl} = -\frac{1}{2n} \sum_{i=1}^{n} \left(1 + \log\left(\sigma_i\right)^2 - \mu_i^2 - \sigma_i^2\right) . \tag{3}$$

Based on the learned embeddings, A' is computed as $A' = \mathrm{ReLU}(ZZ^T)$, where $A'_{ij} > 0$ indicates the presence of an edge between node i and j.

Liu et al. [13] further introduce a diagonal regularization term

$$\mathcal{L}_{dl} = -\frac{\gamma}{n^2} \sum_{i=1}^{n} \log(|A'_{ii}|_{[0,1]} + \epsilon) , \tag{4}$$

where γ is defined as

$$\gamma = \sqrt{1 + \frac{n}{\sum_{i=1}^{n}(A'_{ii}) + \epsilon}}$$

and a diagonal enhancement approach

$$A^{\star} = A' + \gamma \cdot \mathrm{diag}(A') \tag{5}$$

to stabilize training and preserve local information.

The symmetric normalized $\hat{A}$ that SCG produces and that will be the input to later graph operations is computed as

$$\hat{A} = D^{-\frac{1}{2}} \left(A^{\star} + I\right) D^{\frac{1}{2}} . \tag{6}$$

The SCG further produces an adaptive residual prediction $\hat{\boldsymbol{y}} = \gamma \cdot \boldsymbol{\mu} \cdot (1 - \log \boldsymbol{\sigma})$, which is used to refine the final prediction of the network after information has been propagated along the graph.

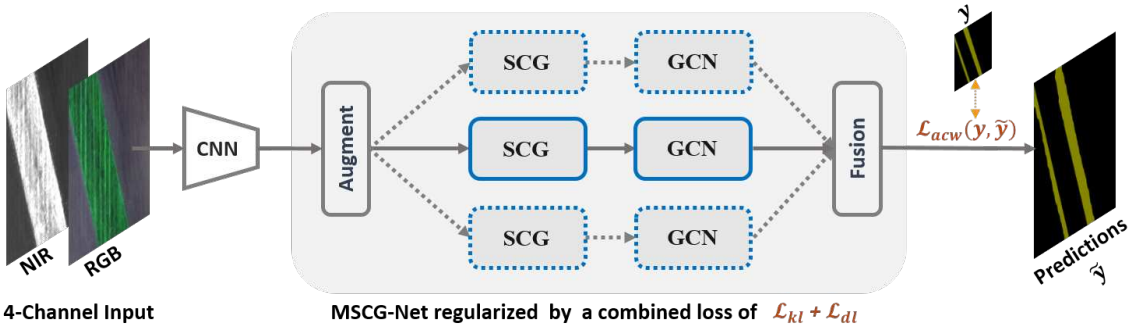

Figure 2. Model architecture of MSCG-Net for semantic labeling includes the CNN-based feature extractor (e.g. customized Se_ResNext50_32x4d taking 4-channel input and output 1024-channel in this work), SCG module taking 3-view (augment the original input by 90°and180°) inputs and K-layer GCNs (K=2 in this work), the Fusion block merging 3-view outputs together, the fused output is projected and upsampled back to 2D maps for final prediction.

2.3. The MSCG-Net

We propose a so-called Multi-view SCG-Net (MSCG-Net) to extend the vanilla SCG and GCN modules by considering multiple rotated views in order to obtain and fuse more robust global contextual information in airborne images in this work. Fig. 2 shows an illustration of the end-to-end MSCG-Net model for semantic labeling tasks. The model architecture details are shown in Table 2.3. We first augment the features (X) learned by a backbone CNN network to multiple views (X_{90} and X_{180}) by rotating the features. The employed SCG-GCN module then outputs multiple predictions: $(\hat{\boldsymbol{y}}, Z^{(2)}), (\hat{\boldsymbol{y}}_{90}, Z^{(2)}_{90}), (\hat{\boldsymbol{y}}_{180}, Z^{(2)}_{180})$ with different rotation degrees (the index indicates the degree of rotation). The fusion layer merges all the predictions together by reversed rotations and element-wise additions as shown in Table 2.3. Finally, the fused outputs are projected and up-sampled back to the original 2-D spatial domain.

We utilize the first three bottleneck layers of a pretrained Se_ResNext50_32x4d or Se_ResNext101_32x4d [3] as the backbone CNN to learn the high-level representations. The output size of the CNN is $\frac{h}{16} \times \frac{w}{16} \times 1024$. Note that, we duplicate the weights corresponding to the Red channel of the pretrained input convolution layer in order to take NIR-RGB 4 channels in the backbone CNN, and GCNs (Equation 1) are used in our model. We utilize ReLU activation and batch normalization only for the first layer GCN. Note, we set $n = 32^2$ and $d = 128$ in this work, and c here is equal to the number of classes, such that $c = 7$ for the experiments performed in this paper.

2.4. Adaptive Class Weighting Loss

The distribution of the classes is highly imbalanced in the dataset (e.g. most pixels in the images belongs to the background class and only few belong to classes such as planter skip and standing water). To address this problem, most existing methods make use of weighted loss functions with pre-computed class weights based on the pixel frequency of the entire training data [5] to scale the loss for each class-pixel according to the fixed weight before computing gradients. In this work, we introduce a novel class weighting method based on iterative batch-wise class rectification, instead of pre-computing the fixed weights over the whole dataset.

The proposed adaptive class weighting method is derived from median frequency balancing weights [5]. We first compute the pixel-frequency of class j over all the past training steps as follows

$$f_j^t = \frac{\hat{f}_j^t + (t-1) * f_j^{t-1}}{t} \ . \tag{7}$$

where, $t \in \{1, 2, ..., \infty\}$ is the current training iteration number, $\hat{f}_j^t$ denotes the pixel-frequency of class j at the current t-th training step that can be computed as $\frac{\text{SUM}(y_j)}{\sum_{j \in C} \text{SUM}(y_j)}$, and $f_j^0 = 0$.

The iterative median frequency class weights can thus be computed as

$$w_j^t = \frac{\text{median}(\{f_j^t | j \in C\})}{f_j^t + \epsilon} \ . \tag{8}$$

here, C denotes the number of labels (7 in this paper), and $\epsilon = 10^{-5}$.

Then we normalize the iterative weights with adaptive broadcasting to pixel-wise level such that

$$\tilde{w}_{ij} = \frac{w_j^t}{\sum_{j \in C}(w_j^t)} * (1 + y_{ij} + \tilde{y}_{ij}) \ , \tag{9}$$

where $\tilde{y}_{ij} \in (0, 1)$ and $y_{ij} \in \{0, 1\}$ denote the ij-th prediction and the ground-truth of class j separately in the current training samples.

In addition, instead of using traditional cross-entropy function which focuses on positive samples, we introduce a

Layers	Outputs	Sizes
CNN	X	$\frac{h}{16} \times \frac{w}{16} \times 1024$
Augment	(X, X_{90}, X_{180})	$3 \times (\frac{h}{16} \times \frac{w}{16} \times 1024)$
SCG	$(\hat{A}, X', \hat{\boldsymbol{y}}), (\hat{A}_{90}, X'_{90}, \hat{\boldsymbol{y}}_{90}), (\hat{A}_{180}, X'_{180}, \hat{\boldsymbol{y}}_{180})$	$3 \times [(n \times n), (n \times 1024), (n \times c)]$
GCN^1	$(Z^{(1)}, Z^{(1)}_{90}, Z^{(1)}_{180})$	$3 \times (n \times d)$
GCN^2	$(Z^{(2)}, Z^{(2)}_{90}, Z^{(2)}_{180})$	$3 \times (n \times c)$
Fusion	$(\hat{\boldsymbol{y}} + Z^{(2)}) \oplus (\hat{\boldsymbol{y}}_{90} + Z^{(2)}_{90})_{r90} \oplus (\hat{\boldsymbol{y}}_{180} + Z^{(2)}_{180})_{r180}$	$(n \times c) \longrightarrow (\frac{h}{16} \times \frac{w}{16} \times c)$
Projection	$\tilde{\boldsymbol{y}}$	$h \times w \times c$

Table 1. MSCG-Net Model Details with one sample of input image size of $h \times w \times 4$. Note: $\oplus$ denotes an element-wise addition, the index (i.e. 90, 180) indicates the rotated degree, while $r90$ and $r180$ denote the reversed rotation degrees.

positive and negative class balanced function (PNC) which is defined as

$$\boldsymbol{p} = \boldsymbol{e} - \log\left(\frac{1-\boldsymbol{e}}{1+\boldsymbol{e}}\right) , \tag{10}$$

where $\boldsymbol{e} = (\boldsymbol{y} - \tilde{\boldsymbol{y}})^2$.

Building on the dice coefficient [15] with our adaptive class weighting PNC function, we develop an adaptive multi-class weighting (ACW) loss function for multi-class segmentation tasks

$$\mathcal{L}_{acw} = \frac{1}{|Y|} \sum_{i \in Y} \sum_{j \in C} \tilde{w}_{ij} * p_{ij} - \log\left(\text{MEAN}\{d_j | j \in C\}\right) , \tag{11}$$

where Y contains all the labeled pixels and d_j is the dice coefficient given as

$$d_j = \frac{2 \sum_{i \in Y} y_{ij} \tilde{y}_{ij}}{\sum_{i \in Y} y_{ij} + \sum_{i \in Y} \tilde{y}_{ij}} . \tag{12}$$

The overall cost function of our model, with a combination of two regularization terms $\mathcal{L}_{kl}$ and $\mathcal{L}_{dl}$ as defined in the equations 3 and 4, is therefore defined as

$$\mathcal{L} \leftarrow \mathcal{L}_{acw} + \mathcal{L}_{kl} + \mathcal{L}_{dl} . \tag{13}$$

3. Experiments and results

We first present the training details and report the results. We then conduct an ablation study to verify the effectiveness of our proposed methods.

3.1. Dataset and Evaluation

We train and evaluate our proposed method on the Agriculture-Vision challenge dataset, which is a subset of the Agriculture-vision dataset [2]. The challenge dataset consists of $21,061$ aerial farmland images captured throughout 2019 across the US. Each image contains four 512x512 color channels, which are RGB and Near Infra-red (NIR). Each image has a boundary map that indicates the region of the farmland, and a mask that indicates valid pixels in the image. Seven types of annotations are included: Background, Cloud shadow, Double plant, Planter skip, Standing Water, Waterway and Weed cluster. Models are evaluated on the validation set with $4,431$ NIR-RGB images segmentation pairs, while the final scores are reported on the test set with $3,729$ images. The mean Intersection-over-Union (mIoU) is used as the main quantitative evaluation metric. Due to the fact that some annotations may overlap in the dataset, for pixels with multiple labels, a prediction of either label will be counted as a correct pixel classification for that label.

3.2. Training details

We use backbone models pretrained on ImageNet in this work. We randomly sample patches of size 512×512 as input and train it using mini batches of size 10 for the MSCG-Net-50 model and size 7 for the MSCG-Net-101 model. The training data (containing 12901 images) is sampled uniformly and randomly flipped (with probability 0.5) for data augmentation and shuffled for each epoch.

According to our best practices, we first train the model using Adam [6] combined with Lookahead [17] as the optimizer for the first 10k iterations and then change the optimizer to SGD in the remaining iterations with weight decay 2×10^{-5} applied to all learnable parameters except biases and batch-norm parameters. We also set $2 \times LR$ to all bias parameters compared to weight parameters. Based on our training observations and empirical evaluations, we use initial LRs of $\frac{1.5 \times 10^{-4}}{\sqrt{3}}$ and $\frac{2.18 \times 10^{-4}}{\sqrt{3}}$ for MSCG-Net-50 and MSCG-Net-101 separately, and also apply cosine annealing scheduler that reduces the LR over epochs. All models are trained on a single NVIDIA GeForce GTX 1080Ti. It took roughly 10 hours to train our model for 25 epochs with batch size 10 over $12,901$ NIR-RGB training images.

3.3. Results

We evaluated and tested our trained models on the validation sets and the hold out test sets with just single feed-forward inference without any test time augmentation

Models	mIoU	Background	Cloud shadow	Double plant	Planter skip	Standing water	Waterway	Weed cluster	mIoU*
MSCG-Net-50	0.547	0.780	**0.507**	0.466	**0.343**	**0.688**	0.513	**0.530**	0.508
MSCG-Net-101	**0.550**	0.798	0.448	**0.550**	0.305	0.654	**0.592**	0.506	**0.509**
Ensemble-TTA	0.599	0.801	0.503	0.576	0.520	0.696	0.560	0.538	0.566

Table 2. mIoUs and class IoUs of our models on Agriculture-Vision test set. Note: mIoU is the mean IoU over all 7 classes while mIoU* is over 6-class without the background, and Ensemble-TTA denotes the two models ensemble (MSCG-Net-50 with MSCG-Net-101) combined with TTA methods [12].

(TTA) or models ensemble. However, we do include results for a simple two-model ensemble (MSCG-Net-50 together with MSCG-Net-101) with TTA for completeness. The test results are shown in Table 2. Our MSCG-Net-50 model obtained very competitive performance with 0.547 mIoU with very small training parameters (9.59 million) and has low computational cost (18.21 Giga FLOPs with input size of $4 \times 512 \times 512$), resulting in fast training and inference performance on both CPU and GPU as shown in Table 3.3. A qualitative comparisons of the segmentation results from our trained models and the ground truths on the validation data are shown in Fig. 3.

3.4. Ablation studies

Effect of the Multi-view. To investigate how the multiple views help, we report the results of the single-view models and the multi-view models trained with both Dice loss and ACW loss in Table 4. Note that, for simplicity, we fixed the backbone encoder as Se_ResNext50 and other training parameters (e.g. learning rate, decay police, and so on.). Also, the mIoUs are computed on the validation set without considering multiple labels. The results suggest that multiple views could improve the overall performance from 0.456 to 0.516 ($+6\%$) mIoU when using Dice loss, and from 0.472 to 0.527 ($+5.5\%$) with the proposed ACW loss.

Effect of the ACW loss. As shown in Table 4, we note that for the single-view models, the overall performance can be improved from 0.456 to 0.472 ($+1.6\%$) mIoU. For the multi-view models, the performance improved $+1.1\%$, increasing from 0.516 to 0.527. Compared to the single-view model SCG-Net with Dice loss, which was proposed in [13] and achieved state-of-the-art performance on a commonly used segmentation benchmark dataset, our Multi-view MSCG-Net model with ACW loss achieved roughly $+7.1\%$ higher mIoU accuracy. We show some qualitative results in Fig. 4 that illustrate the proposed multi-view model and the adaptive class weighting method and show that they help to produce more accurate segmentation results for both larger and smaller classes.

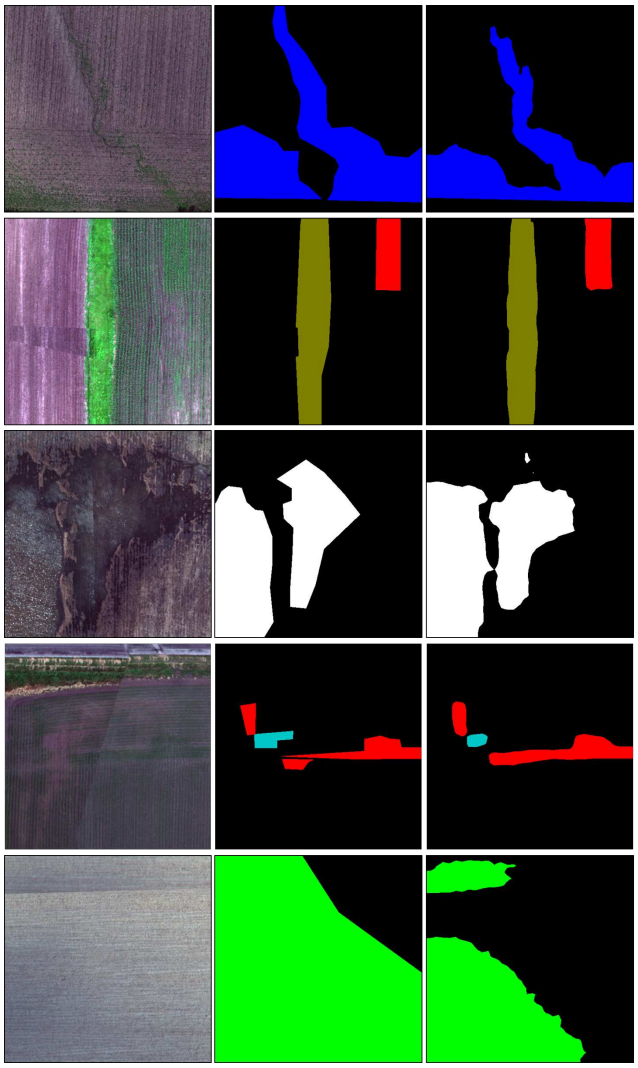

Figure 3. Segmentation results on validation data. From the left to right, the input images, the ground truths and the predictions of our trained models.

4. Conclusions

In this paper, we presented a multi-view self-constructing graph convolutional network (MSCG-Net) to extend the SCG module which makes use of learnable latent variables to self-construct the underlying graphs, and to explicitly capture multi-view global context representations with rotation invariance in airborne images. We further developed a novel adaptive class weighting loss that alleviates the issue of class imbalance commonly found in semantic segmentation datasets. On the Agriculture-Vision

Models	Backbones	Parameters (Million)	FLOPs (Giga)	Inference time (ms - CPU/GPU)
MSCG-Net-50	Se_ResNext50	9.59	18.21	522 / 26
MSCG-Net-101	Se_ResNext101	30.99	37.86	752 / 45

Table 3. Quantitative Comparison of parameters size, FLOPs (measured on input image size of $4 \times 512 \times 512$), Inference time on CPU and GPU separately.

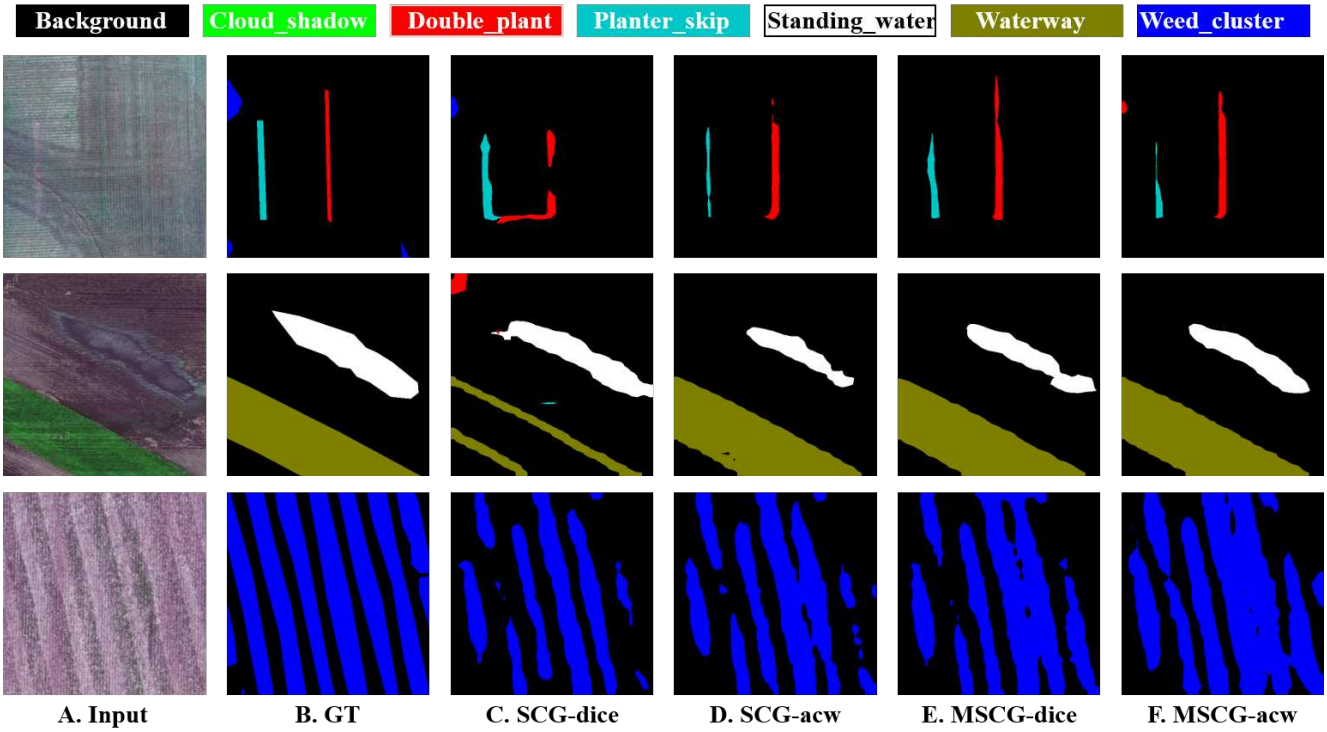

Figure 4. Segmentation results using different models. From the left to right, the input images, the ground truths and SCG-Net with dice loss, SCG-Net with ACW loss, MSCG-Net with dice loss, and MSCG-Net with ACW loss.

Models	Configurations			mIoU
	Multi-view	Dice loss	ACW loss	
SCG-dice		✓		0.456
SCG-acw			✓	0.472
MSCG-dice	✓	✓		0.516
MSCG-acw	✓		✓	0.527

Table 4. Ablation study of our proposed network. Note that, for simplicity, we fixed the learning high-parameters and the backbone encoder, and mIoU is evaluated on validation set without considering overlapped annotations.

challenge dataset, our MSCG-Net model achieves very robust and competitive results, while making use of fewer parameters and being computationally more efficient.

Acknowledgments

This work is supported by the foundation of the Research Council of Norway under Grant 220832.

References

[1] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. 1

[2] Mang Tik Chiu, Xingqian Xu, Yunchao Wei, Zilong Huang, Alexander Schwing, Robert Brunner, Hrant Khachatrian, Hovnatan Karapetyan, Ivan Dozier, Greg Rose, David Wilson, Adrian Tudor, Naira Hovakimyan, Thomas S. Huang, and Honghui Shi. Agriculture-vision: A large aerial image database for agricultural pattern analysis, 2020. 2, 4

[3] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 3

[4] Michael Kampffmeyer, Yinbo Chen, Xiaodan Liang, Hao Wang, Yujia Zhang, and Eric P Xing. Rethinking knowledge graph propagation for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11487–11496, 2019. 1

[5] Michael Kampffmeyer, Arnt-Borre Salberg, and Robert Jenssen. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–9, 2016. 3

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 4

[7] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[8] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 1, 2

[9] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016. 2

[10] Boris Knyazev, Xiao Lin, Mohamed R Amer, and Graham W Taylor. Image classification with hierarchical multigraph networks. *arXiv preprint arXiv:1907.09000*, 2019. 1

[11] Xiaodan Liang, Zhiting Hu, Hao Zhang, Liang Lin, and Eric P Xing. Symbolic graph reasoning meets convolutions. In *Advances in Neural Information Processing Systems*, pages 1853–1863, 2018. 1

[12] Qinghui Liu, Michael Kampffmeyer, Robert Jenssen, and Arnt-Borre Salberg. Dense dilated convolutions' merging network for land cover classification. *IEEE Transactions on Geoscience and Remote Sensing*, page 1–12, 2020. 5

[13] Qinghui Liu, Michael Kampffmeyer, Robert Jenssen, and Arnt-Børre Salberg. Self-constructing graph convolutional networks for semantic labeling. *arXiv preprint arXiv:2003.06932*, 2020. 1, 2, 5

[14] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 1

[15] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571. IEEE, 2016. 4

[16] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 38(5):146, 2019. 1

[17] Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. Lookahead optimizer: k steps forward, 1 step back. In *Advances in Neural Information Processing Systems*, pages 9593–9604, 2019. 4