3M-CDNet: A Universal Powerful Benchmark for Remote Sensing Image Change Detection

Kaiqiang Song and Jie Jiang

Abstract—While deep learning-based methods have gained popularity and have made remarkable progress in remote sensing (RS) image change detection (CD), the limited amount of available data hinders the performance of most supervised methods. The CD networks transferred or derived from other fields can be fronted with a weak generalization capability. Developing a universal benchmark for performance evaluations based on the available datasets is urgent. To address these problems, we proposed a lightweight network, termed 3M-CDNet, which only requires about 3.12 *M* parameters. The lighter the network, the easier it is to train and alleviate overfitting the limited amount of data, resulting in a better generalization capability. 3M-CDNet has a flexible modular design that achieves performance improvements by incorporating plug-and-play modules. 3M-CDNet gains accuracy improvements in two ways: (1) the application of deformable convolutions (*DConv*) in the backbone network to gain a good geometric transformation modeling capacity for CD and (2) the application of an effective two-level feature fusion strategy to enhance the feature representation capacity. 3M-CDNet gains a good generalization capacity by incorporating effective “tricks” to alleviate overfitting, in which online data augmentation (*Online DA*) is applied to increase the diversity of the training samples, and *Dropout* regularization is applied in the classifier. Extensive ablation studies have proved the effectiveness of the core components. Experiment results suggest that 3M-CDNet outperforms state-of-the-art methods on several optical RS datasets and serves as a new universal benchmark. Specifically, 3M-CDNet achieves the best F1-score, i.e., LEVIR-CD (0.9161), Season-Varying (0.9473), and DSIFN (0.7031).

Index Terms—Change detection (CD), convolutional neural networks (CNNs), deformable convolutions (*DConv*), optical remote sensing (RS) images.

I. INTRODUCTION

CHANGE detection (CD) aims to identify and locate the change footprints from multitemporal remote sensing (RS) images acquired over the same geographical region at different times. CD has a wide range of applications, such as land cover and land use identification, urbanization monitoring, and damage assessment [1]. With the rapid development of RS techniques, optical RS images are among the most representative data widely utilized for CD. Especially the high- and very high-resolution RS images, which can reflect abundant spectral and spatial information of geospatial objects, allow us to retain more details and obtain high-quality change maps. In this study, the main concern was binary CD based on optical RS images. CD methods usually generate a pixel-level change map, in which pixels are classified as changed or unchanged.

With the impressive breakthroughs made in deep learning,

CD methods have gradually evolved from traditional [2] to deep neural network (DNN)-based methods [1]. Supervised CD methods built on convolutional neural networks (CNNs) have gained popularity and show promising performance. Some attempts adopted the U-shaped architecture [3]–[5], which concatenates feature maps from different levels through multiple skip connections to improve accuracy. These studies demonstrated that both high-level semantic information and low-level detail information are important in CD. Unfortunately, which multilevel feature fusion strategy is the better is not clear, and dense skip connections bring about heavy computational costs. Alternatively, recent works proposed attention-mechanism-based networks to learn discriminative features and alleviate distracting by pseudo-changes [6]–[10]. However, they usually require a large amount of data to achieve satisfactory performance; otherwise, it is easy to overfit.

Despite the increasing number of raw RS images, the time-consuming and labor-intensive work of manual interpretation still hinders the development of CD methods considering the data-hungry nature of DNNs. Currently, only a few open available labelled datasets can be used for model training and evaluation, such as LEVIR-CD [6], Season-Varying [11], DSIFN [8], HRSCD [4]. Unfortunately, all of them have less data than the ImageNet and COCO datasets. While DNN-based CD methods have made remarkable progress, the limited data hinder the performance of most supervised methods. The CD networks derived from other CV tasks can be fronted with a weak generalization capability. Thus, overfitting becomes one of the main concerns. To address the problem, a lightweight CD model with few parameters becomes an intuitive solution.

Moreover, due to some works adopting different datasets for evaluation, it is difficult to say which one achieves the best performance. Although some methods are evaluated on the same dataset, unfortunately, they adopt different criteria to split the datasets [6], [7], which makes it difficult to compare them directly. In addition, only a few works are openly available to the public. The lack of implementation details results in poor reproducibility, which heavily hinders the development of CD algorithms from research to application. Therefore, developing a universal benchmark based on available datasets is urgent.

The main concerns of this study are as follows: (1) achieve a good generalization capability and (2) gain considerable accuracy improvements. Inspired by Occam’s Razor (“*Non sunt multiplicanda entia sine necessitate*”, or “Entities are not to be multiplied without necessity”), we propose a lightweight network called 3M-CDNet. To provide a universal benchmark with good reproducibility, all the implementation details will be

The authors are with the School of Instrumentation and Optoelectronic Engineering, Key Laboratory of Precision Opto-mechatronics Technology, Ministry of Education, Beijing Advanced Innovation Center for Big Data-Based Precision Medicine, Beihang University, Beijing 100191, China (e-mail: kaiqiangsong@buaa.edu.cn; jiangjie@buaa.edu.cn).

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Fig. 1. Network architecture of 3M-CDNet. Zoom in for an improved view. (a) DConv-based backbone network. (b) Pixelwise classifier. (c) DConv-based bottleneck residual blocks. (d) DConv. Best viewed in color. For brevity, ReLU and batch normalization (BN) at the tail of convolution layers are not fully shown. released at <https://github.com/songkq/3M-CDNet>. The main contributions are summarized as follows.

1) This paper presents a lightweight but powerful 3M-CDNet as a universal benchmark for CD that only involves about **3.12 M** parameters. 3M-CDNet has a modular design with high flexibility for easy plug-in of plug-and-play modules, e.g., *DConv* [12], to gain performance improvements.

2) To the best of our knowledge, this study is among the first to incorporate *DConv* into the backbone network to enhance the geometric transformation modeling capacity for CD.

3) This study explores a “bag of tricks” to achieve considerable performance improvements. Specifically, *Online DA* and *Dropout* regularization [13] were used to improve the generalization capability. A two-level feature fusion strategy was applied to improve the feature representation capacity.

II. PROPOSED METHOD

We proposed a universal benchmark for CD, termed 3M-CDNet, which only involves about 3.12 *M* parameters. As shown in Fig. 1, 3M-CDNet mainly consists of two core components: (a) DConv-based backbone and (b) Pixelwise classifier. The former is used for feature extraction from input $I^{(1,2)} \in \mathbb{R}^{6 \times H \times W}$, e.g., taking as an input with 6 bands a pair of bitemporal RGB images. The latter is used to classify the extracted features into two classes, and then generate a binary change map $CM \in \mathbb{R}^{1 \times H \times W}$, where pixels are either changed or unchanged. 3M-CDNet has a modular structure with high flexibility. It allows achieving performance improvement by incorporating some plug-and-play modules, such as *DConv*.

A. Network Architecture

1) *DConv-based Backbone Network*. As shown in Fig. 1(a), the backbone network of 3M-CDNet is composed of the **Input Layer**, **Layer1**, and **Layer2**. The main concern is to reduce the size of input through consecutive downsampling and convolution operations, and extract features maps with varying degrees of semantics. Specifically, **Input Layer** consists of three stacked 3×3 convolutional layers followed by a MaxPool layer. **Input Layer** is applied to downsample the input by 4 times and transform $I^{(1,2)} \in \mathbb{R}^{6 \times H \times W}$ into a 3-D tensor $X_0 \in \mathbb{R}^{128 \times \frac{H}{4} \times \frac{W}{4}}$, of which the spatial resolution is 1/4 of the input

size. $X_{1st} \in \mathbb{R}^{256 \times \frac{H}{4} \times \frac{W}{4}}$ and $X_{2nd} \in \mathbb{R}^{512 \times \frac{H}{4} \times \frac{W}{4}}$ are feature maps extracted from **Layer1** and **Layer2**, respectively.

(a) *Introducing residual network* [14]. When the CNN goes deeper, which could hamper the convergence, it may suffer from a degradation problem. Therefore, **Layer1** and **Layer2** adopt bottleneck residual blocks [14] as basic units, since residual blocks have the advantage of alleviating the degradation problem and promoting convergence during training. As shown in Fig. 1(c), bottleneck blocks can be formulated as follows:

$$X_{out}^l = \text{ReLU}(H(X_{in}^l) + \zeta(X_{in}^l, \Omega_l)), \quad (1)$$

where X_{in}^l and X_{out}^l are the input and output tensors of the l th residual block, respectively. $\zeta(\cdot)$ indicates the residual function, i.e., the right branch that consists of three stacked convolution layers. $H(\cdot)$ indicates the identity mapping function, i.e., the left branch. $H(\cdot)$ applies a downsampling projection shortcut through a Conv1x1_BN layer only if *stride* is set to 2, e.g., the first block of **Layer2**, otherwise applying an identity shortcut. $\text{ReLU}(\cdot)$ is the rectified linear unit activation function for enhancing the non-linear fitting ability, and batch normalization (BN) is also applied at the tail end of each convolution kernel to facilitate training procedure more stable.

(b) *Introducing deformable convolutions* [12]. Let $X_{in}(p)$ and $X_{out}(p)$ denote the feature at location p of the input and output feature maps, respectively. Given a convolution kernel of K sampling locations, let w_k and p_k denote the weight value and default offset for the k th location of the kernel, respectively, e.g., a 3×3 kernel with $p_k \in \{(-1, -1), (-1, 0), \dots, (1, 1) | K = 9\}$. The deformable convolution [12] can be formulated as follows:

$$X_{out}(p) = \sum_{k=1}^K w_k \cdot X_{in}(p + p_k + \Delta p_k) \cdot \Delta m_k, \quad (2)$$

where Δp_k and Δm_k are the learnable 2-D offsets and modulation factor for the k th location, respectively, and w_k enumerates the weights of the kernel. More precisely, two convolution layers *OffsetConv* and *ModConv* of the same kernel size are separately applied over the input feature maps to obtain Δp_k and Δm_k . Due to the fractional coordinate

$p + p_k + \Delta p_k$, the value of $X_m(p + p_k + \Delta p_k)$ is calculated based on the values of the four surrounding integer points by bilinear interpolation. **DConv** consists of two steps: (1) generate deformable feature maps from the input feature maps based on the learned offsets in the x and y directions $\{\Delta p_k = (\Delta x_k, \Delta y_k)\}_{k=1}^K$ and apply the learned modulation factors $\{\Delta m_k\}_{k=1}^K \in (0, 1)$ to modulate the activation of each location; and (2) apply a regular convolution over the deformable feature maps and then generate the output feature maps (see Fig. 1(d)).

DConvs are applied to replace all the 3×3 convolution layers of the bottleneck blocks in **Layer1** and **Layer2**, termed **DCNBottleneck** (see Fig. 1(c)). In this way, the geometric transformation modeling capability of the backbone network can be enhanced. Therefore, 3M-CDNet has an advantage of addressing the pseudo-changes and overcoming the adverse effects of scale variations of objects with various shapes.

2) **Pixelwise Classifier**. The pixelwise classifier adopts a plain design with only four convolution layers. To obtain a change map of the same spatial resolution as the input, a 2-fold bilinear upsampling is applied after the first and last 1×1 convolution layers. The classifier classifies the extracted features by the backbone network into two classes and predicts a change probability map $CM_{prob} \in \mathbb{R}^{I \times H \times W}$ through a sigmoid layer, of which the values lie in the range $(0, 1)$. Finally, the binary change map $CM \in \mathbb{R}^{I \times H \times W}$ can be generated by applying thresholding over $CM_{prob} \in \mathbb{R}^{I \times H \times W}$ with a threshold of 0.5.

B. Bag of Tricks for Performance Improvements

We used a “bag of tricks” to gain considerable performance improvements. In particular, multilevel feature fusion serves as an indispensable role in improving accuracy of CD networks. Meanwhile, an intuitive solution to achieve good generalization capability comes from two aspects: (1) develop a lightweight model with few parameters and apply regularization; and (2) increase the diversity of the training samples.

1) **Multilevel Feature Fusion Strategies**. Previous works demonstrate that both high-level semantics and low-level detail information are important in CD. However, it is rare for these studies to clearly state which feature fusion strategy is effective. This work compares three kinds of feature fusion strategies as follows: (1) only apply the high-level feature maps X_{2nd} , termed the **one-level strategy**; (2) apply the fusion feature maps X obtained by concatenating the high-level X_{2nd} and low-level X_{1st} along the channel axis, i.e., $X = X_{1st} \odot X_{2nd}$, termed the **two-level strategy**; and (3) apply the **two-level strategy** and then apply an extra fusion feature map $\hat{X} \in \mathbb{R}^{384 \times \frac{H}{4} \times \frac{W}{4}}$, which is obtained by concatenating the output feature maps of the first 1×1 convolution layer of the classifier and X_0 extracted by **Input Layer**, termed the **three-level strategy**. Although some channel attention modules, e.g., CBAM [15], can be applied for improvements, the concatenation operation was selected for its simplicity to achieve high computational efficiency using the minimal number of parameters.

2) **Dropout Regularization** [13]. **Dropout** is a simple yet effective way to prevent neural networks from overfitting.

During training, **Dropout** randomly drops units from the network with a certain probability p_d , which can be equivalent to training numerous different networks simultaneously, i.e., $D(X) = D_m \odot X$, where D_m indicates a binary mask of the same size with the feature map X , and $\odot$ indicates the element-wise multiplication operation. D_m is randomly generated from a **Bernoulli** distribution with a probability p_d , and the units of the feature maps corresponding to the locations of zeros are to be discarded during training. At test time, a neural unit is always presented, and the weights are multiplied by p_d so that the output of the unit would be the same as the expected output at training time, i.e., $D(X) = p_d \cdot X$. As shown in Fig. 1(b), two **Dropout** layers with probabilities of 0.5 and 0.1 are applied at the tail end of the classifier’s 3×3 convolution layers.

3) **Online Data Augmentation**. Data augmentation (**DA**) is a simple yet effective technique for regularizing the network. **DA** can be used to simulate scale variations, illumination variations, and pseudo-changes, such as the spectra changes between bitemporal images. **Online DA** means that do **DA** only when training instead of expanding the training set at the cost of expensive training time. **Online DA** was randomly applied over every batch data with a probability of 0.8 by randomly shifting-rotating-scaling with padding zeros, rotating by 90° , 180° , and 270° , and flipping in horizontal and vertical directions, and applying color jitter. Each kind of augmentation was randomly applied with a probability of 0.5.

C. Loss Function Definition

During training, network parameters are iteratively updated by minimizing the loss between the forward output of 3M-CDNet and the reference change map with the **BP** algorithm according to a specific loss function. Since CD aims to classify the pixels as changed or unchanged, binary cross-entropy (**BCE**) loss is a natural candidate. Due to the widespread class imbalance, dominant unchanged pixels would make models tend to collapse. Thus, a **soft Jaccard** term is introduced, and the loss function can be formulated as shown in Equation (3).

$$L_{BCD} = -\lambda \frac{1}{N} \sum_{n=1}^N (y_n \log(\hat{y}_n) + (1 - y_n) \log(1 - \hat{y}_n)) + (1 - \lambda) \frac{1}{N} \log \sum_{n=1}^N \frac{y_n \cdot \hat{y}_n}{y_n + \hat{y}_n - y_n \cdot \hat{y}_n}, \quad (3)$$

where two terms indicate the **BCE** loss and **soft Jaccard** with the weights λ and $1 - \lambda$ respectively, and λ is empirically set to 0.7. N is the number of training samples, and $\hat{y}_n$ and y_n are pixels of predicted change maps and the reference, respectively.

III. EXPERIMENTS

A. Evaluation Metrics and Experiment Settings

The most common metrics related to the *changed* category were adopted for the quantitative evaluation, including overall accuracy (**OA**), precision (**Pr**), recall (**Re**), F1-score (**F1**), and the intersection of union (**IoU**). **F1** and **IoU** are comprehensive indicators; the higher the value, the better the performance.

The proposed 3M-CDNet was implemented using **PyTorch** framework and optimized by minimizing Eq. (3) through the

TABLE II
RESULTS OF ABLATION STUDIES ON THE LEVIR-CD, SEASON-VARYING, AND DSIFN DATASET

Method	LEVIR-CD			Season-Varying			DSIFN		
	OA (%)	IoU	F ₁	OA (%)	IoU	F ₁	OA (%)	IoU	F ₁
w/ two-level	99.15*	0.8452*	0.9161*	98.62	0.8995	0.9471	89.21	0.5422*	0.7031*
w/ three-level	99.05	0.8291	0.9066	98.48	0.8899	0.9417	85.80	0.4783	0.6471
w/ one-level	99.03	0.8243	0.9037	98.19	0.8720	0.9316	88.06	0.4926	0.6600
w/o DA	99.01	0.8212	0.9018	98.64*	0.8998*	0.9473*	89.72*	0.5128	0.6779
w/o DA/Dropout	98.95	0.8109	0.8956	98.47	0.8892	0.9414	88.00	0.4879	0.6558
w/o DA/DConv	98.92	0.8069	0.8932	98.25	0.8731	0.9323	88.45	0.4683	0.6379
w/o DConv	99.04	0.8283	0.9061	98.06	0.8635	0.9268	89.07	0.5125	0.6777

AdamW optimizer [16] with $\beta_1 = 0.9$ and $\beta_2 = 0.99$, of which the initial learning rate and weight decay were set to 0.000125 and 0.0005, respectively. The minibatch size was set to 16 on an NVIDIA RTX 3090 GPU with 24 GB memory. For evaluation, we adopted three representative datasets, i.e., LEVIR-CD [6], Season-Varying [11], and DSIFN [8]; we applied the criteria as recommended by the creators to split the datasets (see Table I).

TABLE I
EXPERIMENTAL DATASETS

Datasets	Size of Samples	Number of Samples		
		training	validation	test
LEVIR-CD	512 × 512	4,016	1,024	512
Season-Varying	256 × 256	10,000	3,000	3,000
DSIFN	512 × 512	3,546	394	48

B. Ablation Studies

Ablation studies have been conducted to verify each component's contribution of 3M-CDNet. Table II presents the quantitative results on several public datasets, where “w/o” and “w/” mean “without” and “with”, respectively.

1) *Effects of Multilevel Feature Fusion Strategies.* The first three rows of the **Method** applied with all components were used to verify the contributions of three different kinds of multilevel feature fusion strategies. Table II suggests that the **two-level strategy** always achieves the best performance in terms of F_1 , i.e., LEVIR-CD (0.9161), Season-Varying (0.9471), and DSIFN (0.7031). Compared with the **one-level strategy** that lacks low-level details, it increased IoU and F_1 significantly by about 2.09% and 1.24% on LEVIR-CD, 2.75% and 1.55% on Season-Varying, as well as 4.96% and 4.31% on DSIFN, respectively. Unfortunately, the impact of the **three-level strategy** is negligible when using samples with size 512 × 512 for model training, e.g., as shown in the LEVIR-CD and DSIFN. However, the **three-level strategy** helps to achieve an improvement of F_1 (1.01%) and IoU (1.79%), compared to the **one-level strategy**, because it provides more spatial details in the case of using samples with the smaller size of 256 × 256 on Season-Varying. We can conclude that the **two-level strategy** is enough for improvements in our case, while introducing either insufficient or excessive features could bring about an unexpected degradation problem.

2) *Effects of Different Tricks.* The last four rows of the **Method** based on the **two-level strategy** were used to verify the effectiveness of **Online DA**, **Dropout**, and **DConv**. Table II suggests that **Online DA** makes the most impressive contributions on the smaller datasets, i.e., LEVIR-CD and DSIFN. Compared with the situation without **DA**, applying **DA** achieves considerable improvements of F_1 (1.43%) and IoU (2.40%) on LEVIR-CD. Similar situations can be observed on

DSIFN. It demonstrates that **Online DA** is an effective trick to achieve immediate gains through improving the diversity of samples, especially when lacking enough training samples. Otherwise, we can observe that its impact is negligible in the case of Season-Varying dataset with sufficient samples. Meanwhile, **Dropout** can be an effective complementary regularization with **Online DA** for achieving good generalization capacity. In addition, Table II shows that **DConv** serves as an indispensable component for achieving high accuracy. For an instance, the last row shows that performance drops significantly without **DConv**, where F_1 decreased by about 1%, 2.03%, and 2.54% on the three datasets, and IoU decreased by 1.69%, 3.60%, and 2.97%, respectively. In particular, **DConv** helps to improve the generalization capacities in the challenging case of DSIFN, in which training and testing samples are collected from different cities.

C. Comparisons with Other Approaches

1) *Comparisons of Network Parameters and Computational Cost.* Table III shows the computational costs (**GFLOPs**) and number of parameters (**M**) of the CD networks. When calculating the computational cost during testing, it takes as an input 6 × 256 × 256 and 6 × 512 × 512 fixed-size. 3M-CDNet costs 23.66 **G** and 94.64 **GFLOPs** according to the input sizes.

2) *Comparisons on LEVIR-CD.* W-Net [3], FC-EF-Res [4], and Peng *et al.* [5], and attention-based methods STANet [6], DDCNN [7], and FarSeg [17] were selected as benchmarks. In particular, STANet was proposed by the dataset's creator. Table IV presents the quantitative results and suggests that 3M-CDNet outperforms the state-of-the-art FarSeg/DDCNN (w/ **Online DA**) and achieves the best performance in terms of IoU (0.8452) and F_1 (0.9161), which increased by about 1.66% and 0.98% compared to FarSeg, respectively. Moreover, 3M-CDNet makes a better trade-off between precision and recall than other approaches. 3M-CDNet has broken the record and become the new benchmark for the LEVIR-CD dataset. For intuitive comparisons, the visual results are shown in Fig. 2. Compared with the change maps generated by other methods, 3M-CDNet achieves more complete and accurate boundaries and higher internal compactness on objects with various scales and shapes, which is more consistent with the reference. Moreover, 3M-CDNet successfully identifies the tiny gap among the crowded building groups and overcomes the pseudo-changes caused by the spectra changes. However, other methods suffer from more false alarms.

3) *Comparisons on Season-Varying.* More attention-based methods IFN [8], BA²Net [9], and DASNet [10] were selected as benchmarks for the quantitative evaluation. Table V suggests that 3M-CDNet is consistently superior to other approaches in

TABLE III

COMPARISON OF PARAMETERS AND COMPUTATIONAL COSTS

Method	Number of Parameters (<i>M</i>)	Computational Costs (<i>GFLOPs</i>)	
		512×512	256×256
DDCNN [6]	60.21	855.59	213.90
IFN [7]	35.99	328.87	82.22
W-Net [2]	42.55	290.82	72.70
STANet [5]	16.93	155.33	19.58
Peng <i>et al.</i> [4]	9.04	125.35	31.34
FarSeg [16]	31.38	41.74	11.94
FC-EF-Res [3]	1.10	6.89	1.72
3M-CDNet	3.12	94.64	23.66

TABLE IV

COMPARISON OF RESULTS ON THE LEVIR-CD DATASET

Method	<i>Pr</i> (%)	<i>Re</i> (%)	<i>IoU</i>	<i>F₁</i>
STANet [5]	85.01	91.38*	0.7869	0.8808
W-Net [2]	90.37	85.92	0.7871	0.8809
FC-EF-Res [3]	90.22	88.25	0.8055	0.8923
Peng <i>et al.</i> [4]	91.53	88.70	0.8197	0.9009
DDCNN [6]	92.15*	89.07	0.8279	0.9059
FarSeg [16]	91.04	90.22	0.8286	0.9063
3M-CDNet	91.99	91.24	0.8452*	0.9161*

TABLE V

COMPARISON OF RESULTS ON THE SEASON-VARYING DATASET

Method	<i>OA</i> (%)	<i>Pr</i> (%)	<i>Re</i> (%)	<i>F₁</i>
Peng <i>et al.</i> [4]	96.73	89.54	87.11	0.8756
FC-EF-Res [3]	97.25	89.91	87.37	0.8862
IFN [7]	97.71	94.96	86.08	0.9030
BA ² Net [8]	98.94*	88.12	95.28	0.9136
DASNet [9]	98.20	92.20	93.20	0.9270
FarSeg [16]	98.23	87.75	99.56*	0.9328
DDCNN [6]	98.64	96.71*	92.32	0.9446
3M-CDNet	98.64	90.78	99.04	0.9473*

Fig. 2. Comparisons on the LEVIR-CD dataset. Zoom in for an improved view. (a) Image T1. (b) Image T2. (c) Reference change map. (d) 3M-CDNet. (e) FarSeg. (f) DDCNN. (g) STANet. (h) FC-EF-Res. (i) Peng *et al.* (j) W-Net.

Fig. 3. CD results on the Season-Varying dataset. Zoom in for an improved view. (a) Image T1. (b) Image T2. (c) Reference change map. (d) 3M-CDNet.

terms of F_1 (0.9473). Although DDCNN achieves very similar results to 3M-CDNet, 3M-CDNet requires fewer parameters than DDCNN and remarkably reduces the computational cost. Table III shows the number of parameters of DDCNN (60.21 *M*) is about **19 times** that of 3M-CDNet (3.12 *M*), and it has dense skip connections that bring about 213.90 *GFLOPs* computational costs, i.e., about **9 times** of 3M-CDNet. Fig. 3 shows even in the challenging case of large seasonal variations (summer-to-winter/autumn), 3M-CDNet accurately identifies the changed objects of various scales and appearance, including the road changes and building changes.

IV. CONCLUSIONS

The universal network 3M-CDNet is proposed for CD. The lightweight 3M-CDNet exhibits a good generalization capacity by incorporating effective tricks, i.e., *Online DA* and *Dropout* regularization. The flexible 3M-CDNet achieves accuracy improvements in two ways. One is by applying *DConv* in the backbone network to gain good geometric transformation modeling capacity for objects with various scales and appearance. Another is by applying a two-level feature fusion strategy to enhance the feature representation capacity. Experiment results show 3M-CDNet broke the record for accuracy compared to the state-of-the-art methods and is the new benchmark on three public datasets.

REFERENCES

- [1] W. Shi, Z. Min, R. Zhang, S. Chen, and Z. Zhan, "Change detection based on artificial intelligence: State-of-the-art and challenges," *Remote Sensing*, vol. 12, pp. 1688, May 2020.
- [2] T. Celik, "Unsupervised change detection in satellite images using principal component analysis and k-Means clustering," *IEEE Geosci. Remote Sens. Lett.*, vol. 6, no. 4, pp. 772-776, Oct. 2009.
- [3] B. Hou, Q. Liu, H. Wang, and Y. Wang, "From W-Net to CDGAN: Bitemporal change detection via deep learning techniques," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 3, pp. 1790-1802, Mar. 2020.
- [4] R. C. Daudt, B. Le Saux, A. Boulch, and Y. Gousseau, "Multitask learning for large-scale semantic change detection," *Comput. Vision Image Understanding*, vol. 187, pp. 102783, Oct. 2019.
- [5] D. Peng, Y. Zhang, and H. Guan, "End-to-end change detection for high resolution satellite images using improved UNET++," *Remote Sensing*, vol. 11, no. 11, pp. 1382, Jun. 2019.
- [6] H. Chen and Z. Shi, "A spatial-temporal attention-based method and a new dataset for remote sensing image change detection," *Remote Sensing*, vol. 12, no. 10, pp. 1662, May 2020.
- [7] X. Peng, R. Zhong, Z. Li, and Q. Li, "Optical remote sensing image change detection based on attention mechanism and image difference," *IEEE Trans. Geosci. Remote Sens.*, pp. 1-12, 2020.
- [8] C. Zhang, P. Yue, D. Tapete, L. Jiang, B. Shangguan, L. Huang, and G. Liu, "A deeply supervised image fusion network for change detection in high resolution bitemporal remote sensing images," *ISPRS J. Photogramm. Remote Sens.*, vol. 166, pp. 183-200, Aug. 2020.
- [9] Y. Zhang, S. Zhang, Y. Li, and Y. Zhang, "Coarse-to-fine satellite images change detection framework via boundary-aware attentive network," *Sensors*, vol. 20, no. 23, pp. 6735, Nov. 2020.
- [10] J. Chen, Z. Yuan, J. Peng, L. Chen, H. Haoze, J. Zhu, Y. Liu, and H. Li, "DASNet: Dual attentive fully convolutional siamese networks for change detection of high resolution satellite images," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, pp. 1-1, Nov. 2020.
- [11] M. Lebedev, Y. V. Vizilter, O. Vygolov, V. Knyaz, and A. Y. Rubis, "Change detection in remote sensing images using conditional adversarial networks," *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 42, no. 2, pp. 565-571, Jun. 2018.
- [12] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable ConvNets v2: More deformable, better results," in *Proc. CVPR*, LGB, CA, USA, Jun. 2019, pp. 9300-9308.
- [13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929-1958, 2014.
- [14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, LAS, NV, USA, Jun. 2016, pp. 770-778.
- [15] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "CBAM: Convolutional block attention module," in *Proc. ECCV*, Munich, Germany, Sep. 2018, pp. 3-19.
- [16] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. ICLR*, NO, LA, USA, May 2019.
- [17] Z. Zheng, Y. Zhong, J. Wang, and A. Ma, "Foreground-aware relation network for geospatial object segmentation in high spatial resolution remote sensing imagery," in *Proc. CVPR*, SEA, WA, USA, Jun. 2020, pp. 4095-4104.