How to Reduce Change Detection to Semantic Segmentation

Guo-Hua Wang[a], Bin-Bin Gao[b,*], Chenjie Wang[b,c]

[a]*State Key Laboratory for Novel Software Technology, Nanjing University, China*
[b]*Youtu Lab, Tencent, China*
[c]*Department of Computer Science and Engineering, Shanghai Jiao Tong University, China*

Abstract

Change detection (CD) aims to identify changes that occur in an image pair taken different times. Prior methods devise specific networks from scratch to predict change masks in pixel-level, and struggle with general segmentation problems. In this paper, we propose a new paradigm that reduces CD to semantic segmentation which means tailoring an existing and powerful semantic segmentation network to solve CD. This new paradigm conveniently enjoys the mainstream semantic segmentation techniques to deal with general segmentation problems in CD. Hence we can concentrate on studying how to detect changes. We propose a novel and importance insight that different change types exist in CD and they should be learned separately. Based on it, we devise a module named MTF to extract the change information and fuse temporal features. MTF enjoys high interpretability and reveals the essential characteristic of CD. And most segmentation networks can be adapted to solve the CD problems with our MTF module. Finally, we propose C-3PO, a network to detect changes at pixel-level. C-3PO achieves state-of-the-art performance without bells and whistles. It is simple but effective and can be considered as a new baseline in this field. Our code is at `https://github.com/DoctorKey/C-3PO`.

Keywords: Change Detection, Semantic Segmentation, Feature Fusion

---

*Corresponding author
Email address:* `csgaobb@gmail.com` (Bin-Bin Gao)

1. Introduction

Change detection (CD) aims to identify changes that occur in a pair of images captured at different times as illustrated in Fig. 1. A CD algorithm is required to predict change masks in pixel-level. It can be used in updating urban map, visual surveillance, disaster evaluation, agricultural industry and so on. Due to these various applications, CD has attracted more and more attention in the fields of computer vision.

Recently, the deep neural network has achieved remarkable success in various computer vision tasks [1]. It is also adopted to deal with the CD task and achieves state-of-the-art performance. However, most researchers design specific networks from scratch to predict change masks and struggle with the general segmentation problems, such as how to deal with multi-scaled objects [2] and how to improve the mask's details [3]. In this paper, we decouple the change detection and the segmentation, and propose a new paradigm to solve CD as illustrated in Fig. 2. Thanks to it, a general semantic segmentation network can solve the CD problem effectively and efficiently. And general segmentation problems are alleviated by directly applying advantageous semantic segmentation networks. Compared with previous methods, the proposed paradigm make us get free of those segmentation problems so that we can concatenate on the core question: how to detect changes?

We propose a novel and essential insight that there are three possible change types in the CD task, and they should be learned separately. As illustrated by Fig. 1, we name these changes as "appear", "disappear" and "exchange", respectively. These different change types are neglected by previous works (as the change mask in Fig. 1). In this paper, we argue that the fusion feature should extract information of all these changes to help the segmentation head. We propose a module named MTF to Merge two Temporal Features. And MTF is devised with a multi-branch structure and different branches focus on different change types. Experimental results demonstrate distinguishing three change types matters the performance significantly (cf. Tab. 1) and MTF in-

Figure 1: Illustration of change detection task. We propose there are three possible change types (i.e., "appear", "disappear" and "exchange", respectively) in this task. The network should model these three different change types separately.

deed extracts these three changes (cf. Fig. 8). Compared with previous fusion manners, our MTF enjoys high interpretability and reveals the essential characteristic of CD. And it is more generalized that can be easily inserted between the backbone and the segmentation head (as in Fig. 2).

At last, we propose C-3PO, namely Combine 3 POssible change types, to solve the change detection task. C-3PO utilizes the mature techniques of semantic segmentation. So it is reliable and easily deployed. With our MTF module, C-3PO can effectively solve the CD task with high interpretability. It outperforms previous methods and achieves state-of-the-art performance.

Our contributions can be summarized as follows.

• We propose a new paradigm to solve the CD problem. That is reducing change detection to semantic segmentation. Compared with previous paradigm that devises specific networks from scratch, our paradigm is more advantageous that it can make the most techniques in the fields of semantic segmentation.

• We argue that three possible change types are contained in the CD task, and they should be learned separately. The MTF module is proposed to combine two temporal features and extract all three change types. Our MTF is
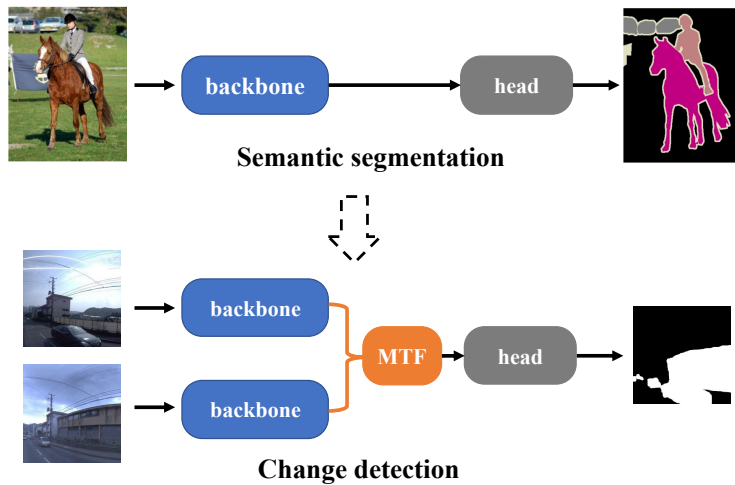
3

Figure 2: Illustration of our paradigm that reducing change detection to semantic segmentation. We propose the MTF module to fuse temporal features. The semantic segmentation network is transformed to solve CD by inserting MTF.

generalized well and can work with lots of segmentation networks. Experiments demonstrate the importance of distinguishing three change types and MTF indeed extracts information to identify them.

• With our paradigm and MTF, we propose C-3PO, a network to detect changes in pixel-level. C-3PO is simple but effective. It achieves state-of-the-art performance without bells and whistles. It can be considered as a new baseline network in the CD field.

2. Related Works

Change detection requires the algorithm to identify changes according to image pairs. There are two scenarios in this field, i.e., remote-sensing change detection (RSCD) [4] and street scene change detection (SSCD) [5, 6, 7]. RSCD is based on remote sensing images and aims at information analysis on the earth's surface. [8] proposes a high frequency attention Siamese network for a finer recognition of changed building objects in very-high-resolution remote sensed images. [9] proposes an unsupervised iterative structure transforma-

tion and conditional random field based multimodal change detection method. SDACD [10] unifies image adaptation and feature adaptation in an end-to-end trainable manner, and handles cross-domain change detection. SSCD cares about street scene images. In this paper, we concentrate on SSCD because it has more applications. Early researches utilize handcrafted feature and feature matching to solve the CD problem [11, 12]. Recently, the deep neural network is adopted and achieves better performance. However, most researchers design specific networks from scratch and struggle with the general segmentation problems. HPCFNet [2] hierarchically combines features at multiple levels to deal with multi-scaled objects. Further, it introduces the MPFL [2] strategy to alleviate the problem that the locations and the scales of changed regions are unbalanced. CosimNet [13] proposes a threshold contrastive loss to deal with the noise in the change masks. DR-TANet [3] introduces the attention mechanism into the CD field and proposes CHVA [3] to refine the strip entity changes. The performance of their methods is determined by both segmentation and change detection. And it is hard to distinguish which one contributes more. Although previous methods also use the framework of semantic segmentation for change detection. They consider these two problems at the same time, i.e., coupling semantic segmentation and CD. The previous paradigm can be considered as designing specific networks from scratch. In this paper, we propose a new paradigm that reducing CD to semantic segmentation, which means directly using a segmentation network to solve CD with minimal changes. Ours explicitly decouples CD into feature fusion and semantic segmentation. MTF is proposed and used to reduce CD to semantic segmentation, whereas previous methods cannot make this reduction explicitly. We can directly adopt a powerful semantic segmentation network, and therefore enjoy their techniques to deal with general segmentation problems. On other hand, we can concentrate on how to improve change detection, i.e., how to design and improve our MTF.

Previous works also study how to fuse two temporal features. The correlation layer is utilized by CSCDNet [14] to estimate optical flow for localizing changes. HPCFNet [2] concatenates features and applies parallel atrous group

5

convolution to capture multi-scaled information. Inspired by the self-attention mechanism, DR-TANet [3] introduces temporal attention for the feature fusion. However, these fusion manners are heavily coupled with their specific network structure and hard to be applied with the general semantic segmentation networks. In this paper, we focus on how to build features for the general segmentation network to identify changes. We propose a novel and essential insight that distinguishing three change types will result in better performance. Finally, the proposed MTF module is more generalized that can be easily inserted in most semantic segmentation networks.

Reduction is one of the most important concepts in theoretical computer science, e.g. NP-Complete problems can be reduced to each other [15]. Transfer learning [16] can be considered as reduction in the fields of machine learning. When it comes to computer vision, RCNN [17] reduces object detection to classification by extracting region proposals and classifying which proposal contains objects. FCOS [18] reduces object detection to semantic segmentation by predicting bounding boxes in a per-pixel fashion. ViT [19] reduces image classification to natural language processing (NLP) by splitting an image into patches and treating them as words in NLP. In this paper, we study how to reduce change detection to semantic segmentation. It is reasonable that both two tasks need to predict labels at the pixel level. And the techniques in the semantic segmentation field can indeed help to solve CD.

Semantic segmentation is a basic computer vision task that predicts the semantic classes on pixel-level. FCN [20] is an early work based on deep learning and achieve remarkable success in this task. Then a serial of DeepLab [21, 22] are proposed and gradually become the mainstream approach. CENet [23] introduces a novel encoder-decoder architecture to capture multi-scale context via ensemble deconvolution. GPNet [24] proposes a gated pyramid module to incorporate both low-level and high-level features.. Recently, attention-based models [25, 26] are introduced in this field, e.g. SETR [27] uses the transformer to deal with this task. In this paper, for robustness and reliability, we adopt FCN and DeepLab to solve the CD task. Note that with our proposed MTF module,
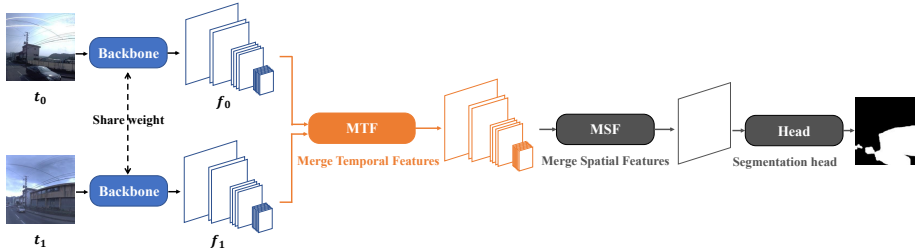
Figure 3: The overall architecture of C-3PO. Two temporal feature pyramids are extracted by the same backbone and merged by MTF. Then, MSF generates a single feature map for the segmentation head to predict change masks.

it is easy to adjust recent segmentation networks to solve the CD problem.

3. The Proposed Method

Fig. 3 illustrates the overall architecture of our network. The main difference between our network and previous ones is that ours decouples change detection into feature fusion and semantic segmentation. Without the MTF module, the remained parts can form into a semantic segmentation network. So we can directly adopt the mainstream architectures of semantic segmentation in these parts. Our MTF module aims at extracting the requisite information from two temporal features for the general segmentation network to identify changes. The designation of MTF is based on our insight that different change types should be learned separately.

Backbone: A general-purpose backbone network is used to extract hierarchical feature maps in our framework. We adopt VGG [28], ResNet [29], and MobileNetV2 [30] in this paper. All backbones can generate pyramid features of multiple scales from $1/32$ to $1/4$ resolution. Given paired images t_0 and t_1, two types of pyramids f_0 and f_1 are extracted by the *same* backbone, respectively.

Merge temporal features (MTF): Fig. 4 shows the detailed architecture of our MTF module which has five branches. The upper three branches use the minus operation to model three different change types, while the under two branches directly use raw features to provide the basic semantic information of
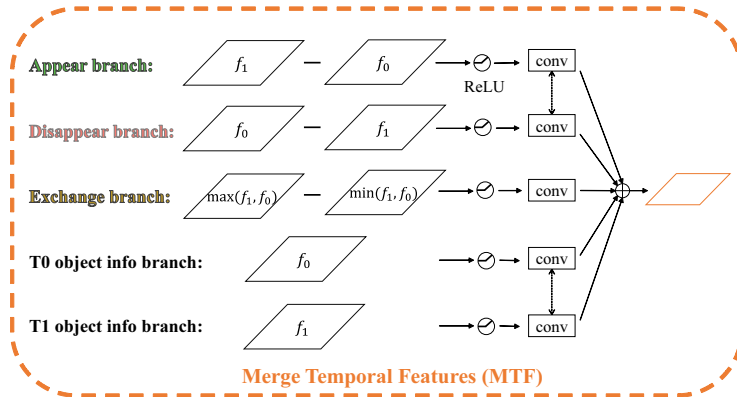
7

Figure 4: The detailed architecture of the MTF module. This module aims at merging two temporal feature maps into one. We use upper three branches to model "appear", "disappear" and "exchange", respectively.

objects. "Appear" means the object only exists in t_1, so we expect f_1 is larger than f_0 in these positions. And we use $\text{ReLU}(f_1 - f_0)$ to model this change type. "Disappear" means the object only exists in t_0. Similar to "Appear", $\text{ReLU}(f_0 - f_1)$ is used in this branch. After the minus and ReLU operations, a conv. layer with kernel size 3×3 is applied on these two branches. Because it is no need to distinguish the semantic changes that may exists in a specific image (i.e., t_0 or t_1) and both two activation maps after ReLU identify there are objects only in one image, the "Appear" and "Disappear" branches share one conv. layer rather than customizing two different conv. layers.

"Exchange" means objects in t_0 and t_1 are different. This change can be considered as a object in t_0 disappears and a new object appears in t_1. To this end, we define the exchange as

$$\text{Exchange} = \text{Disappear} + \text{Appear} \tag{1}$$

$$= \text{ReLU}(f_0 - f_1) + \text{ReLU}(f_1 - f_0) \tag{2}$$

$$= \max(f_0, f_1) - \min(f_0, f_1). \tag{3}$$

This formulation is reasonable due to the change mask of exchange type is according to more significant objects between t_0 and t_1 (cf. Fig. 1). Finally, a
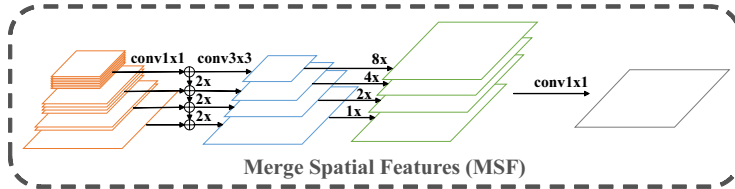
8

Figure 5: The detailed architecture of the MSF module. This module aims at merging the pyramid into a single feature map.

3×3 conv. layer is added after the activation map.

The under two branches directly use f_0 and f_1 to provide the object's semantic information. Objects (e.g. a car) can be detected by the extracting pyramid raw features. The upper three branches will lose this semantic information due to the minus operations. We find that it needs the under two branches to identify objects for detecting changes better.

At last, all branches' outputs are summarized into a single feature map. We force the shape of the output feature map is the same as that of f_0 and f_1 by properly setting the conv. layers' channels. So our MTF can be easily inserted between the backbone and the segmentation head.

With the multi-branch structure, our MTF enjoys sufficient expressivity. Concatenating and subtracting two features are two widely used fusion manners in the fields of change detection [31, 32, 2, 13]. The under two branches in MTF can model the concatenating fusion while the exchange branch computes the absolute difference between two features. So our MTF is more expressive than most previous fusion manners.

Merge spatial features (MSF): Fusing different spatial features can effectively deal with the multi-scaled change objects [2]. Different from previous works that designing complex architectures by themselves, we directly use FPN [33] and make minimal changes to merge spatial features. The FPN architecture has been verified by lots of works [34, 35]. Fig. 5 shows its detailed architecture. Given a pyramid with 4 spatial scales (i.e., 1/4, 1/8, 1/16 and 1/32), all feature maps are reduced to 256 channels by 1×1 conv. firstly. Then

the smaller feature maps are enlarged by $2\times$ bilinear upsampling and added in the lower feature map. 3×3 conv. are used to further process these feature maps. To construct the final single feature map, the upper 3 feature maps are enlarged by bilinear upsampling into the resolution of the lowest feature map. All feature maps are concatenated and reduced to 512 channels by 1×1 conv.

Semantic segmentation head: Once constructing the final feature map, we can adopt the existing semantic segmentation heads to predict the change mask. In this paper, FCN [20] and DeepLabv3 [22] are used. The FCN head first uses 3×3 conv. to reduce the number of channel by 4 times, and then uses 1×1 conv. to further reduce the channel to N classes. Finally, $4\times$ bilinear upsampling and softmax are used to generate the prediction mask at the original image resolution. Compared with FCN, DeepLabv3 is more advantageous that consists of ASPP [21] and can effectively capture multi-scale information.

3.1. Training

The change detection task requires the model to predict a change mask to indicate the change region. This can be considered as a semantic segmentation task with multi classes. Hence, the softmax cross-entropy loss function is used to train our network. However, the distribution of change/background is heavily imbalanced (e.g. $0.06 : 0.94$ in VL-CMU-CD task). Following previous works [2], we use the weighted cross-entropy loss to alleviate this problem. Assume there are total N classes from 0 to $N - 1$ (the 0-th class denotes the background), and the loss function on each pixel is defined as

$$\mathcal{L} = -\sum_{i=0}^{N-1} w_i y_i \log p_i \,, \tag{4}$$

where p_i and y_i are the prediction and ground-truth of the i-th class region, respectively. $w_i = \frac{\sum_{j=0}^{N-1} n_j - n_i}{(N-1)\sum_{j=0}^{N-1} n_j}$ is the balance weight, where n_i is the number of i-th class pixels. We use the training set to calculate these statistics. Finally, the loss averaged over pixels is used for optimization.

Typically, only two classes (i.e., change/background) are considered in the CD field. Recently, a dataset with multi-class is proposed [7]. Note that our

method can solve binary-class and multi-class CD problems with Eq. 4.

3.2. Analysis

"Appear" v.s. "Disappear": The concepts of "Appear" and "Disappear" are relative. As illustrated in Fig. 1, it can be considered as a car appears in the green box. On the other hand, it can also be considered as the ground disappears. Although the latter opinion seems weird (due to humans often focus on objects), these two opinions are both make sense for the change detection task. When it comes to deep neural networks, C-3PO can surprisingly deal with these two opinions. That means given a dataset which only contains "disappear" change, C-3PO can work well with the disappear branch or appear branch. More experimental details can be found in Section 4.1.

Symmetry: Symmetry is another important property of our framework. Obviously, the features generated by our MTF module will not change if we change the order of image pairs (i.e., exchange t_0 and t_1). This symmetry results from the sharing weights backbones and the symmetrical architecture of the MTF module. In MTF, we use the same conv. layer after the appear/disappear branches, and the same conv. layer after two info branches. Hence, exchanging image pairs' order will keep the output unchange. The symmetry implicitly augments training data by exchanging their orders. It is reasonable because exchanging the order should not affect the algorithm to identify changes in an image pair. So this symmetry property can alleviate the overfitting problem and boost performance. More experimental results can be found in Section 4.4.

Structure transfer: Note that our MTF module is versatile to apply in most semantic segmentation frameworks. That means given a semantic segmentation network, it can be used for solving CD tasks by inserting our MTF module. So we in fact propose a framework to generate CD networks. And the performance can be boosted by applying more powerful segmentation networks.

Parameter transfer: Without the MTF module, the remained parts can form into a semantic segmentation network. This network can be firstly trained on semantic segmentation tasks. Then, the parameters can be transferred to the

11

CD task. Note that it is difficult for previous CD networks to decompose into segmentation networks and be pretrained on the semantic segmentation task.

4. Experimental Results

In this section, first, we introduce datasets, implementation details, and evaluation metrics. Then, a series of ablation studies are conducted to study our method. Finally, we compare C-3PO with state-of-the-art methods.

VL-CMU-CD is proposed in [6]. This task *only* contains "disappear" semantic changes. Following previous works, we resize images into 512×512 resolution and split the dataset into training and testing. The training set consists of 933 image pairs, while the testing set consists of 429 pairs. Following previous works [14, 3, 31, 32], the training set is augmented to 3732 pairs by rotation. We train models on the training set and report evaluation metrics on the testing set.

PCD is proposed in [5] and three change types (i.e., "appear", "disappear" and "exchange") exist in it (cf. Fig. 1). PCD can be divided into two subsets (i.e., "GSV" and "TSUNAMI"), and each subset has 100 image pairs and hand-labeled change mask. The resolution of each image is 224×1024. Following previous works [3, 2], we crop the original images by sliding 56 pixels in width. Hence, 3000 paired patches with 224×224 resolution are generated. Then, these patches are augmented by rotate $0°$, $90°$, $180°$ and $270°$. In total, 12000 paired patches are generated. Finally, 5-fold cross-validation is adopted for training and testing. Specifically, each training set consists of 9600 paired patches with 224×224 resolution, while each testing set consists of 20 raw paired images with 224×1024 resolution. Following previous works, we reshape images to 256×1024 for testing.

ChangeSim is proposed in [7] and all three change types exist in it. The training and testing set consist of 13225 and 8212 pairs, respectively. Following [7], the images are firstly resized to 256×256, and then horizontal flipping and color jittering are used for the data augmentation. We simply resize images to

256×256 for testing.

Implementation details. To train the network, we use Adam [36] with an initial learning rate of 0.0001. And the learning rate is decayed by the cosine strategy. We use 4 V100 GPUs with batch size 16 (i.e., 4 images per GPU). For a fair comparison, the backbone is pretrained on ImageNet [37] while other parts are initialized randomly except experiments in Section 4.3. All networks are trained for 100 epochs. Note that *all* experiments use the same training strategy and there are *no* more hyperparameters to tune.

Evaluation metrics. Following previous works [3, 2], we evaluate the performance on VL-CMU-CD and PCD with F1-score. It is calculated upon Precision and Recall. In these two binary CD task, the change regions represent positive while the background (unchanged regions) represent negative. Given true positive (TP), false positive (FP) and false negative (FN), the F1-score is defined as

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \, , \tag{5}$$

where Precision $= \frac{TP}{TP+FP}$ and Recall $= \frac{TP}{TP+FN}$. In addition, some ground-truth masks are not accurate in these change detection tasks (cf. Fig. 6). Following the researches on noise label [38, 39], we evaluate the network after each epoch and report both best and last performance.

ChangeSim contains two CD tasks (i.e., binary and multi-class). Following [7], we evaluate the performance with mIoU, which is defined as

$$\text{mIoU} = \frac{1}{C} \sum_{i=1}^{C} \frac{P_i \cap G_i}{P_i \cup G_i} \, , \tag{6}$$

where C is the total number of category, and P_i and G_i denote the prediction and groundtruth masks for the i-th class.

4.1. Merge temporal features

To evaluate our insight that different change types should be learned separately, we conduct ablative experiments with using different branches in the MTF module.Fig. 7a shows F1-score on VL-CMU-CD for C-3PO with just one
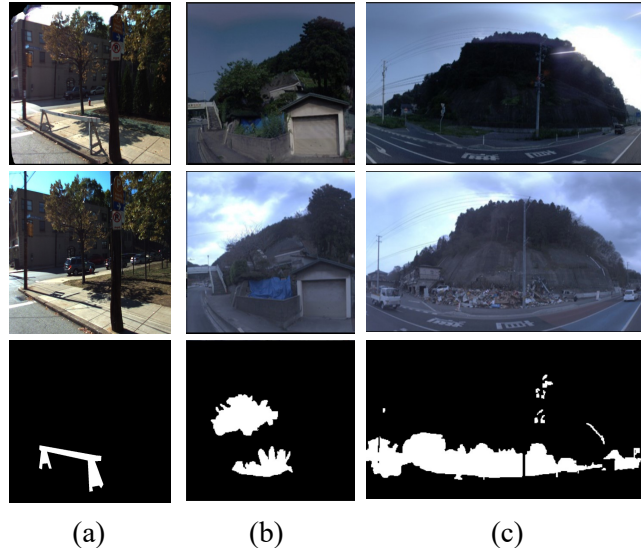
Figure 6: Examples of noisy label. Images in (a) are from VL-CMU-CD and the changed cars are not annotated. Examples in (b) and (c) are from PCD and the changed trees are not annotated well.

type branch after each epoch. Because this task only contains "disappear" changes, both appear and disappear branches perform well according to analysis in Section 3.2. The info branch loses the discrepant information to detect changes. And the exchange branch struggles with the dilemma that detecting changes according to objects or background (i.e., whether to make the object or background feature larger during training). These results demonstrate the manner of feature fusion is needed to suit the task.

GSV is different from VL-CMU-CD that it contains all three change types. Fig. 7b shows results. First, the exchange branch performs best among all three change branches, because three change types exist in this task. Second, because changes in GSV are heavily based on objects, C-3PO works well even with only the info branches. For example, as in Fig. 1, cars are changed with a high probability in image pairs. Third, it is unsuitable to only use the unidirectional change branch such as appear and disappear.

Fig. 8 shows the change masks predicted by C-3PO with different branches.
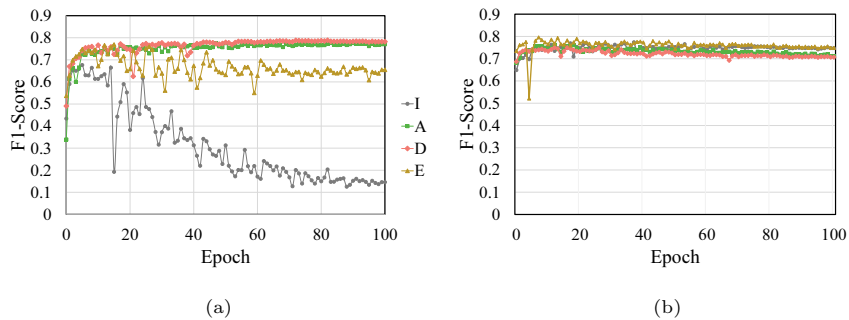
14

Figure 7: F1-score of C-3PO after each training epoch. (a) presents the results on VL-CMU-CD which only contains the "disappear" change. (b) shows the results on GSV which contains three change types. "I", "A", "D" and "E" denote the info, appear, disappear and exchange branches in Fig. 4, respectively.
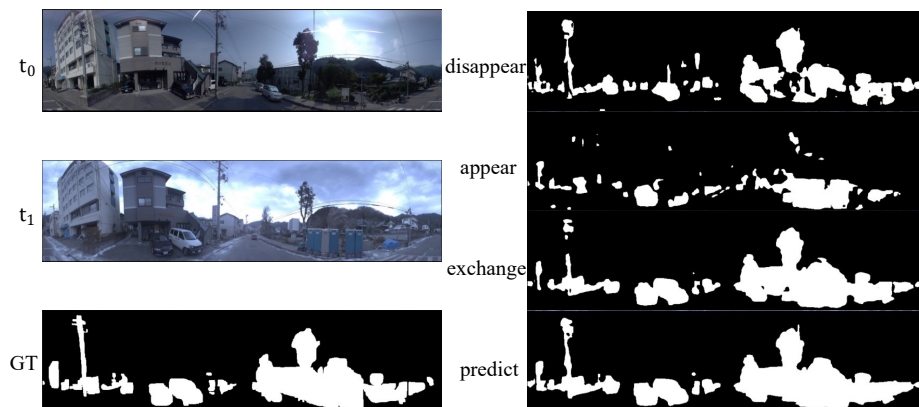

Figure 8: Visual illustration of change masks predicted by C-3PO with different branches. We train C-3PO with all three change branches, then use just one branch to identify changes. "disappear", "appear" and "exchange" denote three change branches, respectively. "predict" uses all branches and "GT" is the ground-truth change mask.

15

Table 1: F1-score (%) on benchmarks for C-3PO with using different branches in the MTF module. "I", "A", "D" and "E" denote the info, appear, disappear and exchange branches in Fig. 4, respectively.

Structure	VL-CMU-CD		GSV		TSUNAMI	
	last	best	last	best	last	best
I	14.6	67.7	75.2	76.9	83.7	84.6
A	77.1	77.7	73.0	76.3	84.1	85.0
D	**78.2**	**78.8**	73.4	76.2	84.2	85.1
E	65.5	77.0	**76.4**	**79.1**	**84.7**	**85.7**
I + A	**78.8**	**79.6**	75.4	77.6	84.6	85.4
I + D	78.3	79.4	75.9	77.9	84.9	**85.9**
I + E	59.4	74.6	**76.6**	**79.4**	**85.0**	85.9
I + A + E	77.7	**79.3**	76.7	78.9	84.9	85.9
I + D + E	**78.2**	79.2	76.8	78.9	84.9	86.0
I + A + D	57.0	77.2	**77.0**	**79.4**	**85.1**	**86.1**
I + A + D + E	53.7	76.5	**77.6**	**79.4**	84.7	85.4

Three change branches indeed extract different features to identify different change types. The prediction of "exchange" is close to the final prediction due to it combines appear and disappear (cf. Equation 1). So our MTF enjoys high interpretability and reveals the essential characteristic of CD.

Tab. 1 presents the results of C-3PO with different branch combinations. On VL-CMU-CD, comparing "I + A/D" and "A/D", we find adding object information is beneficial to detect change. "I + E" still works badly due to adding object information cannot solve the dilemma. And "I + A + D" achieves a similar performance to "I + E" because we define the exchange by Equation 1. Finally, "I+A/D" performs best among all combinations. Note that VL-CMU-CD only contains the disappeared changes. Models should have the right inductive bias for the unidirectional changes. As discussed in Sec. 3.2, both the appear and disappear branches have the inductive bias that makes C-3PO detect changes according to objects or backgrounds. When it comes to GSV

Table 2: F1-score (%) for C-3PO with different positions to merge temporal information and different pretrained models. $\odot_T$ and $\odot_S$ denotes temporal and spatial fusions, respectively. B and H denotes backbone and semantic segmentation head, respectively. ImageNet pretraining means B is pretrained on ImageNet, while COCO pretraining means B, $\odot_S$ and H are pretrained on COCO.

Structure	Pretrain	VL-CMU-CD		GSV		TSUNAMI	
		last	best	last	best	last	best
$\odot_T \to B \to \odot_S \to H$	ImageNet	69.1	70.6	43.7	53.2	63.1	67.8
	COCO	70.2	72.1	47.4	52.5	63.6	67.2
$B \to \odot_T \to \odot_S \to H$	ImageNet	77.6	79.4	77.6	**79.4**	84.7	85.4
	COCO	**78.9**	**79.9**	77.8	78.8	**84.8**	85.5
$B \to \odot_S \to \odot_T \to H$	ImageNet	76.4	76.7	75.3	77.7	84.7	**85.7**
	COCO	78.3	78.8	75.9	77.7	84.6	85.4
$B \to \odot_S \to H \to \odot_T$	ImageNet	7.4	12.6	55.9	61.9	73.3	78.0
	COCO	67.3	72.6	60.2	64.1	74.7	77.7

and TSUNAMI, these two datasets contain all change types. Hence, the model should be equipped with all change branches, especially the exchange branch for the bidirectional changes. Considering the difference among these benchmarks, in the rest of this paper we use "I + D" on VL-CMU-CD, and "I + A + D + E" on GSV, TSUNAMI, and ChangeSim. Note that these structures are not the best combinations in Tab. 1. We choose the model's structure according to the training set rather than the performance on the testing set. From training sets, VL-CMU-CD only contains disappeared changes, while other datasets have all change types. Our choices achieve comparable performance with the best combinations (less than 1% F1-score). We believe these choices are reasonable and give sound conclusions in the latter experiments.

4.2. Temporal fusion position

We study where to merge temporal information and Tab. 2 summarizes the experimental results. We placed the MTF module in four different positions. "$\odot_T \to B \to \odot_S \to H$" means combining image pairs by MTF directly.

Premature merging results in overfitting easily, because it does not utilize the backbone's ability to detect objects. Inserting MTF between the backbone and MSF achieves the best performance. And placing MTF after MSF also works well. These two settings utilize the backbone to detect objects and the head to predict change masks. Merging temporal information before spatial information can make the MSF module obtain more information for detecting changes. Finally, placing MTF after the head expects the head can predict the semantic mask of each image and the change mask can be predicted by simply combining two semantic masks. That is difficult due to only change masks are available to train the network.

4.3. Parameter transfer

We evaluate the performance of C-3PO with COCO [40] pretraining and Tab. 2 summarizes the results. As introduced in Section 3.2, B, $\odot_S$ and H can form into a general semantic segmentation network which can be pretrained on COCO. On VL-CMU-CD, C-3PO with COCO pretraining outperforms that with ImageNet pretraining by a significant margin. We find the boost is largest with $B \to \odot_S \to H \to \odot_T$ structure. With this structure, $B \to \odot_S \to H$ is pretrained, and it has the ability to semantically segment. Hence, it can work well that adding $\odot_T$ to directly merge two semantic maps. On PCD, the training set is sufficient to train the segmentation head.

4.4. Symmetry

The symmetry of C-3PO is due to the sharing weights used in the backbone, the info branches and the appear/disappear branches. We conduct experiments for C-3PO without sharing specific parts and Tab. 3 summarizes results. The symmetry matters significantly on GSV and sharing weights in all three parts achieves the best performance. Without the symmetry, it is easy for the network to overfit the individual information in two input images (e.g., the weather in each image). Using a symmetrical framework can make the network concentrate on the changes in the image pair. And it implicitly augments the image pair by exchanging their order which further alleviates the overfitting problem.

Table 3: F1-score (%) for C-3PO with sharing specific parts. "B", "I" and "C" denote the backbone, the info branches and the appear/disappear branches, respectively. "✓" denotes sharing weights. Note that we use "I+D" on VL-CMU-CD, so it does not need to consider the symmetry for appear/disappear branches in this model. We evaluate networks on the first cross-validation fold.

| Share | | | VL-CMU-CD | | GSV | | TSUNAMI | |
B	I	C	last	best	last	best	last	best
×	×	×	79.5	80.2	72.2	76.3	88.2	89.0
×	✓	✓	79.4	80.2	72.4	76.5	88.3	88.9
✓	×	×	-	-	73.0	77.0	88.7	89.2
✓	✓	×	-	-	74.8	77.7	88.5	89.1
✓	×	✓	**79.9**	**80.6**	74.7	77.4	88.2	89.1
✓	✓	✓	79.5	80.2	**76.5**	**79.3**	**88.8**	**89.4**

Table 4: F1-score (%) for C-3PO with/without weighted loss. "✓" denotes using balance weight in Eq. 4. We evaluate networks on the first cross-validation fold.

| weighted loss | VL-CMU-CD | | GSV | | TSUNAMI | |
	last	best	last	best	last	best
	77.4	79.3	**76.8**	**79.7**	88.3	89.1
✓	**79.5**	**80.2**	76.5	79.3	**88.8**	**89.4**

4.5. Weighted loss

Following previous works [3, 14, 2], we adopt the weighted cross-entropy loss to alleviate the imbalance problem. Tab. 4 makes a comparison for the loss without weighting. This balanced weight matters significantly on VL-CMU-CD, because its change/background distribution is about 0.06 : 0.94, more imbalanced than that of PCD (0.29 : 0.71). Overall, as argued by previous works, we also appreciate using weighted cross-entropy loss on these tasks.

4.6. More configurations

To demonstrate the generalization of our paradigm, we evaluate C-3PO with different configurations of the backbone, MSF and semantic segmentation head.

Table 5: F1-score (%) for C-3PO with different configurations of the backbone, MSF and the head.

Backbone	MSF	Head	VL-CMU-CD		GSV		TSUNAMI	
			last	best	last	best	last	best
ResNet-18	1	FCN	73.4	73.6	71.1	73.6	82.1	82.5
ResNet-18	2	FCN	77.9	78.3	75.7	77.8	84.1	84.9
ResNet-18	3	FCN	78.1	79.2	76.9	78.5	**85.1**	**85.7**
ResNet-18	4	FCN	**79.5**	**80.2**	**77.2**	**79.0**	85.0	**85.7**
ResNet-18	1	DeepLabv3	72.7	73.1	71.7	74.0	81.7	82.3
ResNet-18	2	DeepLabv3	**79.0**	79.1	76.6	78.1	84.3	84.8
ResNet-18	3	DeepLabv3	**79.0**	79.3	77.3	79.2	**84.9**	**85.6**
ResNet-18	4	DeepLabv3	77.6	**79.4**	**77.6**	**79.4**	84.7	85.4
MobileNetV2	4	DeepLabv3	77.8	79.3	**76.9**	78.8	84.9	85.8
ResNet-50	4	DeepLabv3	77.6	79.9	76.5	**80.0**	85.1	86.2
VGG-16	4	DeepLabv3	**79.6**	80.0	76.8	79.5	**85.6**	**86.5**

Tab. 5 summarizes the results. First, the number of MSF denotes how many feature maps are used in the MSF module. "1" means only using the feature map with spatial scale 1/32. The F1-score gets higher with adding more low-level feature maps. The low-level feature maps contain more details for C-3PO to segment more accurately. Second, we try FCN and DeepLabv3 as the semantic segmentation head. Both FCN and DeepLabv3 perform well on these benchmarks. Third, different backbones are adopted. The backbone with a larger capacity results in better performance.

4.7. Different viewpoints

In the case of street scene change detection, the viewpoints of two input images may differ slightly. We perform the perspective transformation of the input images to study this scenario. Fig. 9 presents visualization results. In each sub-figure, the top two images are the images captured at t_0 and t_1, respectively. The bottom two masks are the ground truth and the model's prediction, respectively. We transform one image with keeping another unchanged. Note that
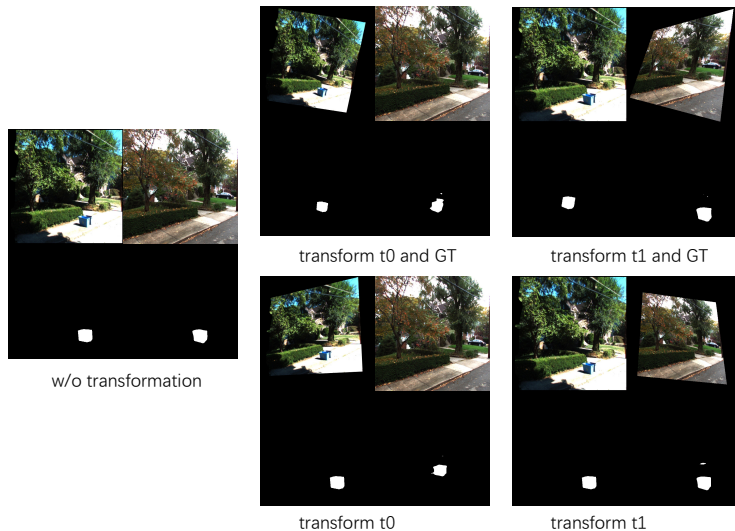
20

Figure 9: Visualization of C-3PO on testing images with perspective transformations.

Table 6: Results for C-3PO on VL-CMU-CD w.r.t. different object sizes.

	Small	Medium	Large	All
Ratio (%)	$0 \sim 2$	$2 \sim 6$	$6 \sim$	$0 \sim$
#Data	133	157	139	429
F1-score (%)	76.1	80.3	81.4	79.3

all predictions are from the same model trained without this transformation augmentation. First, to evaluate the accuracy, the crucial problem is how to define the groundtruth (GT) change mask, i.e., choosing which viewpoint for it. We find that it easily results in misalignment of the model's prediction and the GT mask. Our model predicts according to the viewpoint of t_0, because the disappeared object exists in this image. And our model is robust that detecting this object in all cases. Overall, we suggest adopting techniques from multi-view geometry to align two images first to alleviate the perspective problem, and therefore avoid the dilemma of how to define GT masks.
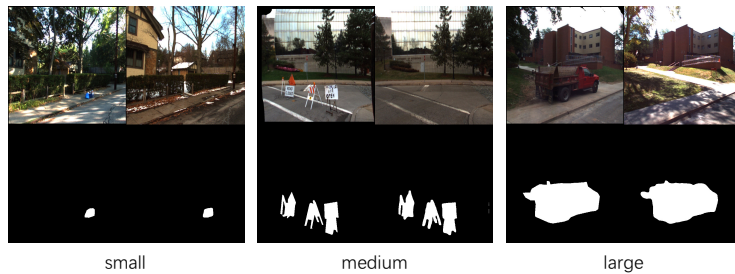
Figure 10: Visualization of C-3PO on testing images with different object sizes.

Table 7: F1-score (%) for C-3PO and previous methods on VL-CMU-CD.

Method	Backbone	VL-CMU-CD	
		last	best
FC-EF [31]	U-Net	43.2	44.6
FC-Siam-diff [31]	U-Net	65.1	65.3
FC-Siam-conc [31]	U-Net	65.6	65.6
DR-TANet [3]	ResNet-18	72.5	75.1
CSCDNet [14]	ResNet-18	75.4	76.6
C-3PO	ResNet-18	77.6	**79.4**
HPCFNet [2]	VGG-16	75.2	
C-3PO	VGG-16	79.6	**80.0**

4.8. Different object sizes

To study the performance w.r.t. the changing object size, we split the testing set of VL-CMU-CD into three folds. Given the binary change masks, we calculate the ratio for the changing area over the whole area. According to this ratio, "Small", "Medium", and "Large" testing sets are obtained (cf. Tab. 6) with comparable data amounts. F1-scores are evaluated on each testing set. As shown in Tab. 6, it is more challenging to detect small objects. Fig. 10 visualizes results with different sizes. Overall, our model detects all objects successfully.

Table 8: F1-score (%) for C-3PO and previous methods on PCD.

Method	Backbone	GSV		TSUNAMI		Average	
		last	best	last	best	last	best
FC-EF [31]	U-Net	63.3	64.7	77.1	77.7	70.2	71.2
FC-Siam-diff [31]	U-Net	64.0	66.2	78.6	79.5	71.3	72.8
FC-Siam-conc [31]	U-Net	69.8	70.4	81.2	81.6	75.5	76.0
DR-TANet [3]	ResNet-18	72.0	74.3	83.0	84.5	77.5	79.4
CSCDNet [14]	ResNet-18	71.1	75.0	83.2	84.8	77.1	79.9
C-3PO	ResNet-18	77.6	**79.4**	84.7	**85.4**	81.2	**82.4**
HPCFNet [2]	VGG-16	77.6		86.8		82.2	
C-3PO	VGG-16	76.8	**79.5**	85.6	86.5	81.2	**83.0**

4.9. Compared with state-of-the-art

Finally, we compare C-3PO with state-of-the-art approaches. Tab. 7 and 8 summarize the results. For a fair comparison, C-3PO *only* utilizes the ImageNet pretrained backbones as previous works. DeepLabv3 is adopted to be head in C-3PO. We rerun the code of state-of-the-arts supplied by authors and report the F1-score of the last and best epoch. Note that HPCFNet [2] does not release the code, and it is difficult to reproduce their method. So we directly cite their numbers. On VL-CMU-CD, C-3PO outperforms CSCDNet and HPCFNet by 2.8% and 4.8%, respectively. That is a significant improvement compared with previous works. With the FCN head, the performance can be further boosted by 0.8%. On the PCD benchmark, C-3PO outperforms CSCDNet and HPCFNet by 2.5% and 0.8%, respectively.

Finally, we evaluate C-3PO on the ChangeSim benchmark. Tab. 9 summarizes the results. The ChangeSim benchmark contains two tasks. The multi-class setting requires the model to classify different change types, while the binary setting expects the model to generate the binary change mask. In the multi-class setting, "new" and "missing" can be roughly considered as "appear" and "disappear" in our paper, respectively. And all other change types ("change", "rotated", and "replaced") are considered as "exchange" in our model. We sim-

Table 9: The mIoU (%) performance on the ChangeSim benchmark for C-3PO and previous methods. "S", "C", "N", "M", "Ro" and "Re" denote "static", "change", "new", "missing", "rotated" and "replaced", respectively.

Method	Binary			Multi-class					
	S	C	mIoU	S	N	M	Ro	Re	mIoU
ChangeNet [32]	73.3	17.6	45.4	80.6	9.1	6.9	11.6	6.6	23.0
CSCDNet [14]	87.3	22.9	55.1	90.2	12.4	6.0	17.5	7.9	26.8
C-3PO	90.4	28.8	**59.6**	92.6	13.3	8.0	16.9	8.0	**27.8**

ply set different numbers of final output channels according to different settings. At last, C-3PO outperforms other methods on both two tasks by a significant margin. Although ours improves the state-of-the-art results on this benchmark, it is still challenging and needs more effort to overcome the difficulties. Different from PCD and VL-CMU-CD, ChangeSim contains more change types, i.e., "rotated". To imporve the performance on ChangeSim, a more sophisticated MTF module should be proposed. However, this paper focuses on the general model for all change detection datasets. We leave the sophisticated model for only ChangeSim as the future work.

Overall, our C-3PO outperforms state-of-the-art by a significant margin. And due to its simplicity, we believe it can be served as a new baseline in this field and inspire more researches.

4.10. Visualization

We visualize the predicted masks of C-3PO. Figs. 11, 12, 13, and 14 illustrate the results of VL-CMU-CD, GSV, TSUNAMI, and ChangeSim, respectively. In each sub-figure, the top two images are the images captured at t_0 and t_1, respectively. The bottom two masks are the ground truth and the model's prediction, respectively.

Figure 11: Visualization of C-3PO on the testing set of VL-CMU-CD. The last row shows some failure cases.
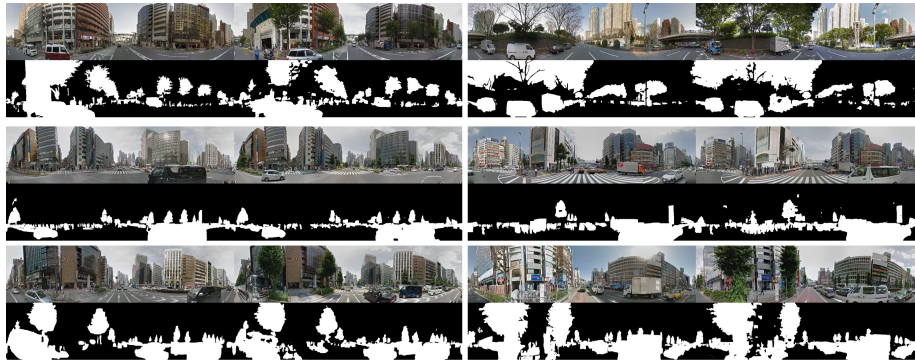

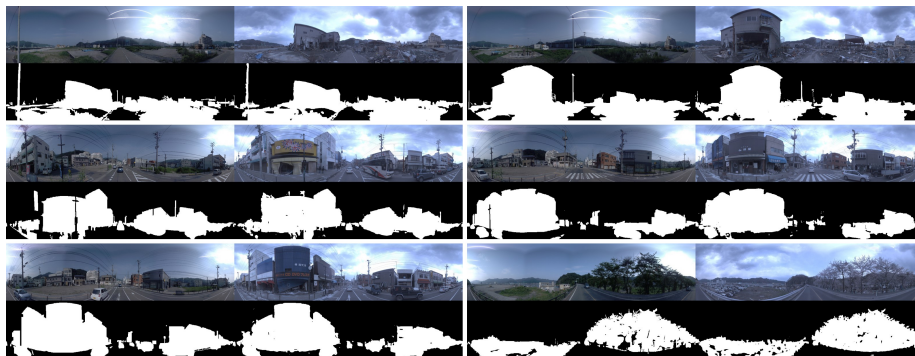Figure 12: Visualization of C-3PO on the testing set of GSV.

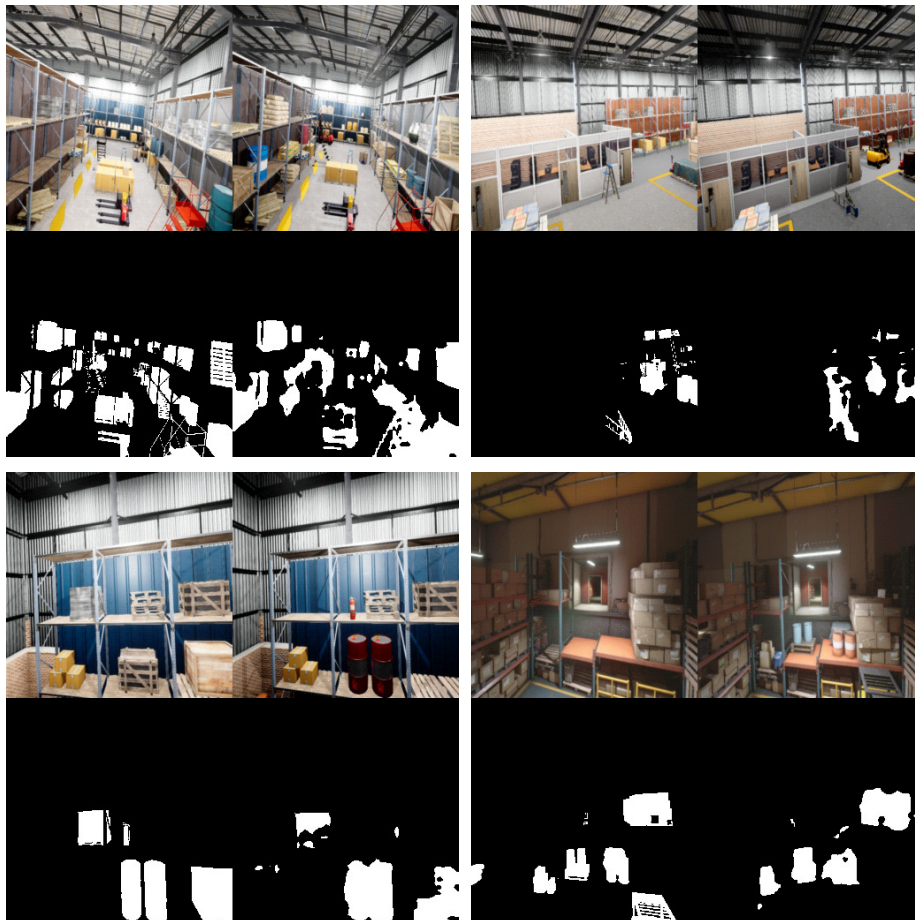Figure 13: Visualization of C-3PO on the testing set of TSUNAMI.

Figure 14: Visualization of C-3PO on the testing set of ChangeSim.

5. Conclusions and Future Work

To solve the change detection problem, we proposed a new paradigm that reduces CD to semantic segmentation. Our framework decouples the CD parts and the segmentation parts. Directly applying the mainstream semantic segmentation networks help us relieve from the general segmentation problems in the CD task. And we only need to study how to fuse features for change detection. We proposed the MTF module to achieve this target. MTF is designed based on our insight of CD, that is the fusion features needs to contain the information of three possible change types. Finally, we proposed a C-3PO network for change detection. C-3PO is simple but effective. And it achieves state-of-the-art performance without bells and whistles.

With our paradigm, applying a more powerful semantic segmentation network is a promising way to further boost the performance. And how to identify the specific change object is an important problem in real-world applications. We leave these as future works.

References

[1] H. Wang, Y. Zhang, J. Wu, Versatile, full-spectrum, and swift network sampling for model generation, Pattern Recognition 129 (2022) 108729.

[2] Y. Lei, D. Peng, P. Zhang, Q. Ke, H. Li, Hierarchical paired channel fusion network for street scene change detection, IEEE TIP 30 (2021) 55–67.

[3] S. Chen, K. Yang, R. Stiefelhagen, DR-TANet: Dynamic receptive temporal attention network for street scene change detection, in: IEEE IV, 2021, pp. 502–509.

[4] H. Chen, Z. Shi, A spatial-temporal attention-based method and a new dataset for remote sensing image change detection, Remote Sensing 12 (10) (2020) 1662.

[5] K. Sakurada, T. Okatani, Change detection from a street image pair using CNN features and superpixel segmentation, in: BMVC, Vol. 61, 2015, pp. 1–12.

[6] P. F. Alcantarilla, S. Stent, G. Ros, R. Arroyo, R. Gherardi, Street-view change detection with deconvolutional networks, Autonomous Robots 42 (7) (2018) 1301–1322.

[7] J.-M. Park, J.-H. Jang, S.-M. Yoo, S.-K. Lee, U.-H. Kim, J.-H. Kim, ChangeSim: Towards end-to-end online scene change detection in industrial indoor environments, in: IROS, 2021, pp. 1–8.

[8] H. Zheng, M. Gong, T. Liu, F. Jiang, T. Zhan, D. Lu, M. Zhang, HFA-Net: High frequency attention siamese network for building change detection in VHR remote sensing images, Pattern Recognition 129 (2022) 108717.

[9] Y. Sun, L. Lei, D. Guan, J. Wu, G. Kuang, Iterative structure transformation and conditional random field based method for unsupervised multimodal change detection, Pattern Recognition 131 (2022) 108845.

[10] J. Liu, W. Xuan, Y. Gan, Y. Zhan, J. Liu, B. Du, An end-to-end supervised domain adaptation framework for cross-domain change detection, Pattern Recognition 132 (2022) 108960.

[11] R. J. Radke, S. Andra, O. Al-Kofahi, B. Roysam, Image change detection algorithms: a systematic survey, IEEE TIP 14 (3) (2005) 294–307.

[12] K. Sakurada, W. Wang, N. Kawaguchi, R. Nakamura, Dense optical flow based change detection network robust to difference of camera viewpoints, arXiv preprint arXiv:1712.02941.

[13] E. Guo, X. Fu, J. Zhu, M. Deng, Y. Liu, Q. Zhu, H. Li, Learning to measure change: Fully convolutional siamese metric networks for scene change detection, arXiv preprint arXiv:1810.09111.

[14] K. Sakurada, M. Shibuya, W. Wang, Weakly supervised silhouette-based semantic scene change detection, in: ICRA, 2020, pp. 6861–6867.

[15] R. M. Karp, Reducibility among combinatorial problems, in: Complexity of computer computations, Springer, 1972, pp. 85–103.

[16] Q. Yang, Y. Zhang, W. Dai, S. J. Pan, Transfer learning, Cambridge University Press, 2020.

[17] R. Girshick, J. Donahue, T. Darrell, J. Malik, Region-based convolutional networks for accurate object detection and segmentation, IEEE TPAMI 38 (1) (2015) 142–158.

[18] Z. Tian, C. Shen, H. Chen, T. He, FCOS: Fully convolutional one-stage object detection, in: ICCV, 2019, pp. 9627–9636.

[19] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, in: ICLR, 2021, pp. 1–21.

[20] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: CVPR, 2015, pp. 3431–3440.

[21] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs, IEEE TPAMI 40 (4) (2017) 834–848.

[22] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, arXiv preprint arXiv:1706.05587.

[23] Q. Zhou, X. Wu, S. Zhang, B. Kang, Z. Ge, L. J. Latecki, Contextual ensemble network for semantic segmentation, Pattern Recognition 122 (2022) 108290.

[24] Y. Zhang, X. Sun, J. Dong, C. Chen, Q. Lv, GPNet: gated pyramid network for semantic segmentation, Pattern Recognition 115 (2021) 107940.

[25] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, H. Lu, Dual attention network for scene segmentation, in: CVPR, 2019, pp. 3146–3154.

[26] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, W. Liu, CCNet: Criss-cross attention for semantic segmentation, in: ICCV, 2019, pp. 603–612.

[27] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr, et al., Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers, in: CVPR, 2021, pp. 6881–6890.

[28] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: ICLR, 2015, pp. 1–14.

[29] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, 2016, pp. 770–778.

[30] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, MobileNetV2: Inverted residuals and linear bottlenecks, in: CVPR, 2018, pp. 4510–4520.

[31] R. C. Daudt, B. Le Saux, A. Boulch, Fully convolutional siamese networks for change detection, in: ICIP, 2018, pp. 4063–4067.

[32] A. Varghese, J. Gubbi, A. Ramaswamy, P. Balamuralidhar, ChangeNet: A deep learning architecture for visual change detection, in: ECCV Workshops, 2018, pp. 1–16.

[33] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: ICCV, 2017, pp. 2117–2125.

[34] A. Kirillov, R. Girshick, K. He, P. Dollár, Panoptic feature pyramid networks, in: CVPR, 2019, pp. 6399–6408.

[35] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: ICCV, 2017, pp. 2961–2969.

[36] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: ICLR, 2015, pp. 1–15.

[37] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge, IJCV 115 (3) (2015) 211–252.

[38] K. Yi, J. Wu, Probabilistic end-to-end noise correction for learning with noisy labels, in: CVPR, 2019, pp. 7017–7025.

[39] J. Li, R. Socher, S. C. Hoi, DivideMix: Learning with noisy labels as semi-supervised learning, in: ICLR, 2020, pp. 1–14.

[40] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common objects in context, in: ECCV, Vol. 8693 of LNCS, Springer, 2014, pp. 740–755.