Analysis of adapter in attention of change detection Vision Transformer

Ryunosuke Hamada[1,2], Tsubasa Minematsu[1,3], Cheng Tang[1,4], and Atsushi Shimada[1,5]

[1] Kyushu University, 744 Motooka, Nishi-ku, Fukuoka 819-0395, Japan
[2] hamada.ryunosuke.769@s.kyushu-u.ac.jp
[3] minematsu.tsubasa.659@m.kyushu-u.ac.jp
[4] tang@limu.ait.kyushu-u.ac.jp
[5] atsushi@ait.kyushu-u.ac.jp

Abstract. Vision Transformer (ViT) contributes to accurate change detection with robustness to background changes. However, retraining ViT requires a large amount of computation to adapt to unlearned scenes. This study investigates the addition of learnable parameters into change detection ViT to reduce the computational complexity of retraining. We introduce MLP as an adapter as an addition to the attention output and the residual connection of the change detection ViT and apply LoRA method to the change detection ViT. We evaluate the retraining of additional parameter models for various background changes and analyze proper setting of additional parameters to adapt the target scenes. Introducing MLP and LoRA to change detection ViT improves the accuracy for the target scenes without competition between two additional parameter methods.

Keywords: Change detection · Vision Transformer · Adapter

1 Introduction

Accurate change detection methods are needed for surveillance systems using security cameras. Background subtraction methods can be used for the change detection task, which extracts regions where there is a scene that does not exist regularly. The background is defined as the scene that is regularly observed in the image. Background subtraction detects the foreground by removing the background from the input image. One of the challenges of background subtraction is background changes such as dynamic background, shadow, and illumination change.

Recently, Vision Transformer (ViT) [7] has demonstrated highly accurate performance in several vision tasks. Change detection methods using ViT have also been studied. Wang et al. [24] proposed TransCD using ViT for change detection. TransCD inputs two images into a ViT and computes the feature representation of each image. It detects changes by taking the difference between two feature representations. In addition to being robust to background changes

that occur with fixed cameras, such as dynamic backgrounds and illumination changes, TransCD does not detect changes where different types of backgrounds are compared, such as those caused by PTZ camera scenes.

Change detection using ViT achieved high accuracy on trained datasets. However, the pre-trained change detection ViT can demonstrate lower performance on untrained scenes. Fine-tuning is often used to improve the accuracy on untrained scenes by retraining the model with a small number of untrained scene images. However, fine-tuning all parameters of ViT is computationally expensive because of the large dimension size. As a method to reduce the amount of calculation during fine-tuning ViT, adapter [19] was proposed. Adapter is a small number of parameters added to the model. We can handle target datasets with a low amount of calculation by tuning only the adapter for the target datasets because of its small size. ViT with adapters was effective in image classification tasks.

When adding an adapter for change detection ViT, it is necessary to consider the appropriate processing block in ViT for adding additional parameters. The knowledge that the adapter stores by retraining depends on the location of adapter, and affects the performance of the adapter model. For instance, Adaptformer [3], which introduced an adapter into the Multi-Layer Perceptron (MLP) block, enabled better image recognition than that of Visual Prompt Tuning [12], which introduced additional parameters into input tokens. When training a change detection model with an adapter, it is undesirable that the adapter only acquires knowledge about a specific object in the training dataset.

To keep the ability for change detection in ViT with adapters, we analyze the way to apply an adapter to acquire knowledge about changes in the training dataset. We introduce additional learnable parameters to attention, the residual connection, and attention weights in the change detection ViT and evaluate the contribution of each introduction method. We introduce MLP-based adapter for attention and the residual connection, and also introduce the existing additional parameter method to change detection ViT; LoRA [11]. We analyze the appropriate additional parameter method for change detection ViT by comparing the results of retraining with untrained scenes and changing the dimension size of additional parameters.

2 Related Works

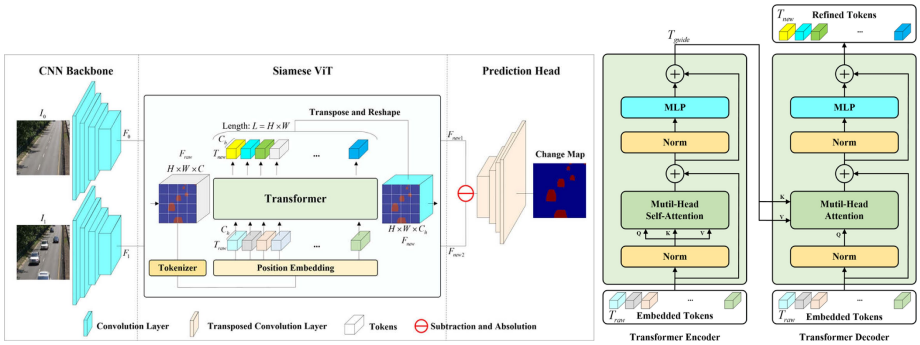
Various background modeling have been proposed for handling background changes such as illumination changes and dynamic backgrounds. Conventional background subtraction methods represent the background that is stationary in the input image with a background model created using the median [17] and statistics [15, 20] of pixels in the background image, and detect regions that deviate from the background as changes. Thereafter, background feature extraction using convolutional neural networks (CNNs) was proposed in [6]. Because the background subtraction method using CNNs enabled highly accurate change detection, various methods have been studied [1]. However, CNNs have problems

such that their locality, which depends on the size of the receptive field, limits the range of images that can be referenced to obtain a feature representation. As the model for computing global feature representation from the entire image, ViT [7] was proposed and enabled highly accurate image classification. Following the emergence of ViT, their applications to change detection have been studied. Wang et al. [24] proposed TransCD as a model to introduce ViT to change detection through background subtraction. It inputs two images into ViT and computes the feature representation of each image. Chen et al. [2] proposed the change detection model of ViT that inputs the concatenated token sequences of each input image to the ViT encoder to model the temporal and spatial features between images. Inspired by the success of the Swin Transformer [16] in image classification, Zhang et al. [25] proposed SwinUNet for change detection. TransBlast [18] incorporates the inductive bias obtained from the CNN into the Transformer-based model training. Self-supervised learning using the extended loss function by subspace learning contributed to robust background/foreground separation even with training using limited labeled data. GIBS-Net [5] introduced global information of input images calculated by ViT into BSUV-Net [21] and improved the performance for unseen scenes. GIBS-Net also shows that it is possible to perform change detection with good accuracy even when we reduce the number of layers in GIBS-Net and computational complexity.

ViT provides highly accurate results for each task. However, it requires enormous computational complexity for fine-tuning because it has a huge number of parameters. Additionally, changing the parameters of a trained model may result in forgetting learned knowledge. In response to these issues, Rebuffi et al. proposed an adapter [19], which is a small number of parameters added to the model. The model fine-tunes the adapter module only without changing the parameters of the trained model. It is possible to improve the accuracy with low computation complexity by fine-tuning the adapter for untrained scenes and retaining pre-trained knowledge. Methods such as Adapter that add parameters for fine-tuning inside a model have been applied to various deep-learning tasks. Studies have been conducted on which modules of the model are appropriate to add parameters to. For image classification, AdaptFormer [3] introduced an adapter to the MLP ViT block, making it possible to store knowledge for a long token sequence of input. In the field of natural language processing, by adding an adapter module in two places after the attention and FeedForward blocks in Bert [14], it has become possible to handle multiple tasks with fewer calculations [10]. The knowledge and functionality of the adapter model depend on the location and shape of the adapter. Depending on the role we intend to add to a model, we need to verify where it is appropriate to introduce an adapter.

2.1 The baseline change detection ViT

We selected TransCD [24] as a baseline change detection ViT. The reason is that it is a highly accurate change detection model with ViT and its internal processing is not changed from the first proposed ViT [7]. TransCD computes the feature representations of two images; the background image and the input



(a) TransCD **(b)** SViT

Fig. 1: Overview of TransCD [24] and the SViT in TransCD

image, using a CNN and Siamese Vision Transformer (SViT). Thereafter it computes the difference between the obtained feature representations. Fig. 1a shows the overview of the TransCD model.

First, each image is input to a CNN, and the features of the images are extracted from the feature map, which is converted into input tokens by dividing the map into patches. Positional encoding is added to input tokens, which is the positional information of the patch within the image. Next, each input token sequence, T^1_{raw} and T^2_{raw} is input to the SViT, two ViTs that share weights. The SViT models the features of each image from the feature map while acquiring the global relationship between each token and computes the two feature maps.

Fig. 1b shows the overview of the SViT. In the encoder, an attention block computes the Multi-Head Self-Attention (MSA) of encoder input Z_{l-1} passed through Layer-Normalization (LN) and adds the residual connection (Eq. (2)). The MLP block computes the MLP of the attention block output Z_{l-1} passed through LN and adds the residual connection (Eq. (3)). Eqs. (2) and (3) are repeated k times and we obtain the output T_{guide}. In the decoder, the attention block computes the Multi-Head Attention (MA) of encoder input Y_{l-1} passed through LN and encoder output T_{guide}, and adds the residual connection (Eq. (6)). The MLP block computes Y_l similar to the encoder (Eq. (7)). Eqs. (6) and (7) are repeated k times and we obtain output T_{new}. Finally, TransCD computes the feature difference of two feature maps, T^1_{new} and T^2_{new}, and converts it into a change map in the prediction head.

`Encoder`

$$Z_0 = T_{\text{raw}} \tag{1}$$

$$Z'_l = \text{MSA}(\text{LN}(Z_{l-1})) + Z_{l-1}, \tag{2}$$

$$Z_l = \text{MLP}(\text{LN}(Z'_l)) + Z'_l, \tag{3}$$

$$T_{\text{guide}} = \text{LN}(Z_{k-1}) \tag{4}$$

`Decoder`

$$Y_0 = (T_{\text{guide}}, T_{\text{raw}}) \tag{5}$$

$$Y_l' = \text{MA}(T_{\text{guide}}, \text{LN}(Y_{l-1})) + Y_{l-1} \tag{6}$$

$$Y_l = \text{MLP}(\text{LN}(Y_l')) + Y_l' \tag{7}$$

$$T_{\text{new}} = \text{LN}(Y_{k-1}) \tag{8}$$

, where $l = 1, 2, ..., k-1$ is the number of layers.

3 Introducing the additional parameter to a change detection ViT

3.1 Analysis of ViT in change detection

We analyze the role of the internal processing of TransCD to gain insights into the appropriate places to introduce an adapter. A previous study on TransCD [24] revealed that TransCD can detect changes with high accuracy owing to the representation of image features. Therefore, in this analysis, we focused on the attention block in the SViT decoder, where TransCD computes the feature representation. The attention block comprises two main types of processes, the MA layer and residual connection. We analyze the role that these two processes play in change detection. We amplified the outputs of the attention layer or residual connection in TransCD by applying an appropriate scalar. Next, we input the untrained dataset to the amplified-TransCD. We compared the differences in change detection results between amplified TransCD and baseline TransCD. We estimated the amplified function has a role in dealing with the differences.

We used a TransCD pre-trained with Change Detection.Net 2014 dataset (CDNet-2014) [8] as the baseline model. We selected People-and-Foliage in SBM-Net [13] as untrained scenes. Background and Input in Fig. 2 show the background and input images in People-and-Foliage. People in the input image are the detection targets, and plants and a car in the background image are background objects. Baseline in Fig. 2 shows the result of change detection by the baseline TransCD. The result images show foreground pixels as white regions. The baseline model failed to detect some of the foreground human regions.

After that, we amplified the output of attention and residual connection at various rates and observed the change detection results. In Eq. (6), we multiplied MA by a scalar value to amplify the output of attention. The value implies the amplification rate of attention. We also multiplied Y_{l-1} by a scalar value to amplify the residual connection. We used various scalar ranges from 1.0 to 5.0.

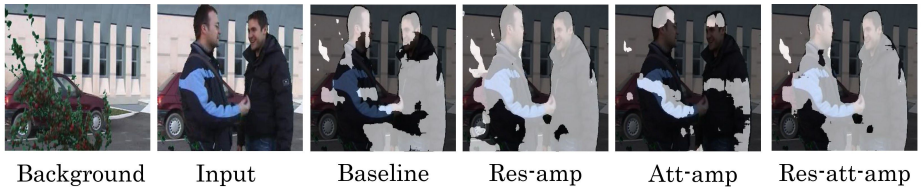


Fig. 2: Results of change detection with amplification

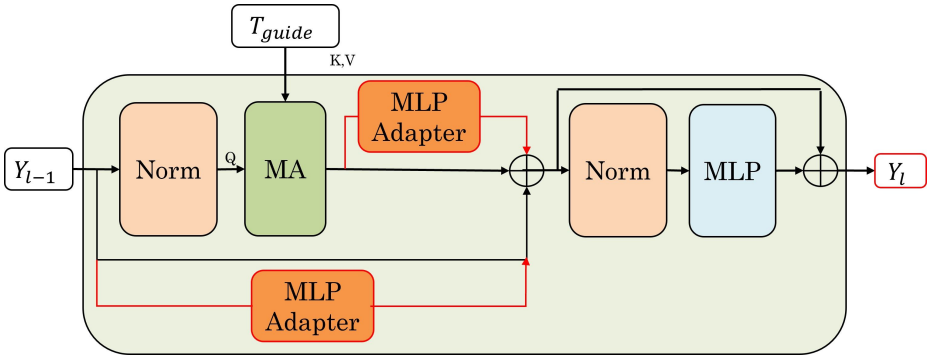


Fig. 3: The model of TransCD Decoder with MLP-Adapter

Res-amp, Att-amp, and Res-att-amp in Fig. 2 show the results of change detection by the model with various amplifications. Res-amp is the result of the model amplifying residual connection by 1.5. In Res-amp, the accuracy of foreground detection improved compared to that of the baseline model. This indicates that amplification of the residual connection is effective in suppressing defects in foreground detection. However, a new false detection of the plants occurred at the left of the image. Att-amp is the result of the model amplifying the attention by 1.5. Att-amp detected fewer foregrounds than the baseline model, but it suppressed the false detection of the plants. This result shows that amplifying attention effectively suppresses false detection of the background. Res-att-amp in Fig. 2 shows the results of change detection by the model amplifying attention and the residual connection by 1.5. Compared to Res-amp, the result had no false detection of the background caused by amplification of the residual connection. From these results, attention suppresses the false detection of background changes and the residual connection improves the foreground detection. Therefore, we expected that introducing learnable parameters to attention and the residual connection would contribute to adapting to the target scenes.

3.2 The additional parameters for change detection ViT

We introduce the learnable parameters for retraining to various functions of ViT and evaluate the effectiveness of each introduction method. We propose 2 types

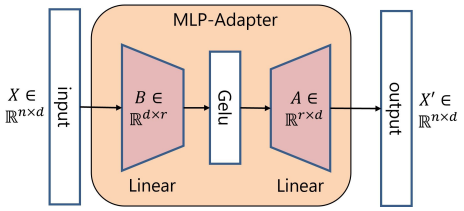


Fig. 4: The model of MLP-Adapter

of introduction; MLP blocks in attention output and the residual connection, and LoRA layer in attention weights.

First, we propose to introduce MLP block as adapter to TransCD. We call the MLP block MLP-Adapter. We introduce MLP-Adapter for two functions; Multi-head Attention (MA) and the residual connection. Fig. 3 shows the model of TransCD Decoder with MLP-Adapters. Eq. (9) shows the process in TransCD Decoder with two MLP-Adapters. MLP-Adapter for MA inputs the output of MA and executes the MLP process. It adds its output to the output of MA. MLP-Adatper for the residual connection inputs the input of l th Decoder and calculates the MLP process. It adds its output to the residual connection.

$$Y_l' = Y_l'' + \text{MLP-Adapter}(Y_l'') + Y_{l-1} + \text{MLP-Adapter}(Y_{l-1}) \tag{9}$$

, where $Y_l'' = \text{MA}(T_{\text{guide}}, \text{LN}(Y_{l-1}))$.

MLP-Adapter consists of three step processes. Fig. 4 shows the detail of MLP-Adapter and Eq. (10) shows the calculation in MLP-Adapter. MLP-Adapter has the internal dimension size r as a hyperparameter. First, it processes the input $X \in \mathbb{R}^{n\times d}$, where n is the number of patches and d is the hidden dimension size, by the first linear layer and resizes the hidden dimension from d to r by multiplying $B \in \mathbb{R}^{d\times r}$. Second, the activation function Gelu [9] processes the output of the first linear layer. Third, the second linear layer inputs the output of the activation function and outputs d dimension matrix $X' \in \mathbb{R}^{n\times d}$ the same size as the input of MLP-Adapter by multiplying $A \in \mathbb{R}^{r\times d}$.

$$\text{MLP-Adapter}(X) := \text{Gelu}(XB)A \tag{10}$$

Second, we apply LoRA to change detection ViT. LoRA introduces LoRA layer to the weight of the model. LoRA layer consists in two matrixes; one is the matrix initialized to a normal distribution and the other is the zero matrix. LoRA can learn additional knowledge about the target domain. We introduce LoRA layer to attention weights of change detection ViT and evaluate the effectiveness of LoRA in the change detection tasks.

4 Experiments

First, we evaluate the effectiveness of various additional parameter models for change detection ViT. Second, we show the experiments to analyze the suitable

hyperparameter of MLP-Adapter. Third, we introduce scalar parameter adapter to change detection ViT and analyze the propriety of introducing adapter to attention and the residual connection.

As the baseline, we used the TransCD pre-trained with CDNet-2014 [23]. It includes SViT with four encoder layers and four decoder layers. The input of the baseline model is a token sequence generated by splitting an image into 16×16 patches.

Every model was retrained and evaluated for each dataset independently. The training and evaluation datasets comprised the target datasets, and they are made to be different images. Training images are from the first half of the dataset, whereas evaluation images are from the second half of the dataset.

4.1 Experiments of MLP-Adapter

We evaluated tuning various additional parameter models by untrained scenes. We constructed three additional parameter models of TransCD; MLP-Adapter, LoRA, and MLP+LoRA. We analyzed the effectiveness of these models by comparing each other and the fine-tuned baseline model. In MLP-Adapter, the internal dimensions for both MA and the residual connection are four. We selected the best combination of internal dimensions because we make fair comparisons with LoRA. We also show the analysis of suitable combination of internal dimensions at Sec. 4.2. In LoRA, we introduced four-dimensional LoRA layers to attention key, query, and value weights. In MLP+LoRA, we introduced both two MLP-Adapters and LoRA to TransCD. We set the internal dimensions of MLP-Adapter to four and the rank of LoRA to four; They are the same setting as MLP-Adapter and LoRA. We executed retraining for 10 training datasets; Lasiesta I_IL_02 [4], LIMU Light switch on/off in Indoor Scene [6], and BMC Real [22] Video001 to Video009 except for Video003. We excluded Video003 due to its small number of images. We set the number of epochs to 500, and the learning rate to 0.0002. We set dropout in tuning parameters in each experiment setting. We calculated Precision, Recall, and F1 score of each dataset in each method.

Tab. 1 summarizes the accuracy of retraining by each method for each dataset. Fig. 5 shows the change detection results of each method. First, we compared the results of MLP-Adapter with ones of LoRA. MLP-Adapter resulted in higher accuracy in three datasets, especially in Video001. The average F1 score of MLP-Adapter was Almost same as that of LoRA; 0.0296 lower than LoRA. We consider MLP-Adapter is an effective additional parameter model as LoRA from the above results. However, MLP-Adapter failed training in Video004. We should consider that MLP-Adapter has some scenes where it is not effective.

Second, we analyzed MLP+LoRA by comparing other methods. The average F1 of MLP+LoRA was the highest in additional parameter models. MLP+LoRA resulted in higher F1 than MLP-Adapter in seven datasets and higher F1 than

[6] https://limu.ait.kyushu-u.ac.jp/dataset/en/

Table 1: Results of retraining for various datasets by each method

	MLP-Apdater			LoRA		
Dataset	F1	Precision	Recall	F1	Precision	Recall
Lasiesta	**0.9288**	0.9573	0.9019	0.9228	0.9393	0.9069
LightSwitch(LIMU)	0.2273	0.1464	0.5079	0.2305	0.1358	0.7611
Video001(BMC Real)	**0.6884**	0.7851	0.6130	0.5060	0.4790	0.5363
Video002(BMC Real)	0.5435	0.8270	0.4047	0.5648	0.8610	0.4203
Video004(BMC Real)	0.1664	0.9246	0.0914	0.4734	0.8980	0.3215
Video005(BMC Real)	0.1637	0.0911	0.8044	0.1697	0.0948	0.8066
Video006(BMC Real)	0.6996	0.8891	0.5767	0.7446	0.9129	0.6287
Video007(BMC Real)	0.4911	0.4382	0.5585	0.4890	0.4060	0.6148
Video008(BMC Real)	0.7605	0.7885	0.7344	0.7934	0.7959	0.7910
Video009(BMC Real)	0.6360	0.8878	0.4955	0.7072	0.8469	0.6070
Average	0.5305	0.6735	0.5688	0.5601	0.6370	0.6394
	MLP+LoRA			Fine-Tuning		
Dataset	F1	Precision	Recall	F1	Precision	Recall
Lasiesta	0.9227	0.9667	0.8826	0.9243	0.9655	0.8865
LightSwitch(LIMU)	**0.6221**	0.8205	0.5009	0.4133	0.3205	0.5820
Video001(BMC Real)	0.6713	0.7087	0.6376	0.6301	0.6491	0.6121
Video002(BMC Real)	0.5630	0.8400	0.4234	**0.6828**	0.8222	0.5838
Video004(BMC Real)	0.3038	0.9441	0.1810	**0.3061**	0.8930	0.1847
Video005(BMC Real)	0.1016	0.0543	0.7931	**0.3500**	0.2409	0.6399
Video006(BMC Real)	0.7242	0.9086	0.6020	**0.8061**	0.8960	0.7326
Video007(BMC Real)	0.5313	0.5818	0.4889	**0.6051**	0.7130	0.5255
Video008(BMC Real)	**0.8073**	0.8057	0.8090	0.7917	0.7940	0.7894
Video009(BMC Real)	0.6848	0.8473	0.5747	**0.7402**	0.8420	0.6603
Average	0.5932	0.7478	0.5893	**0.6250**	0.7136	0.6197

Table 2: The number of tuned parameters of each method

	MLP-Adapter	LoRA	MLP+LoRA	Fine-tuning
Number of parameter	18464	24576	43040	11158211

LoRA in four datasets. MLP+LoRA improved accuracy in Video004 where MLP-Adapter failed training and got the highest F1 in LightSwitch. MLP+LoRA did not reduce accuracy significantly in any datasets. These results show that MLP-Adpater and LoRA do not compete with each other. Therefore, introducing both MLP-Adapter and LoRA contributes to the improvement of accuracy of retraining.

Tab. 2 shows the number of tuned parameters in each method. The amount of parameters in MLP-Adapter was about 0.17% of those of Fine-tuning. Even in the larger additional parameter model MLP+LoRA, the number of tuned parameters was less than 0.4% of that of Fine-tuning. Therefore, introducing MLP and LoRA to change detection ViT contributes to the reduction of computational complexity while improving the accuracy of change detection ViT.

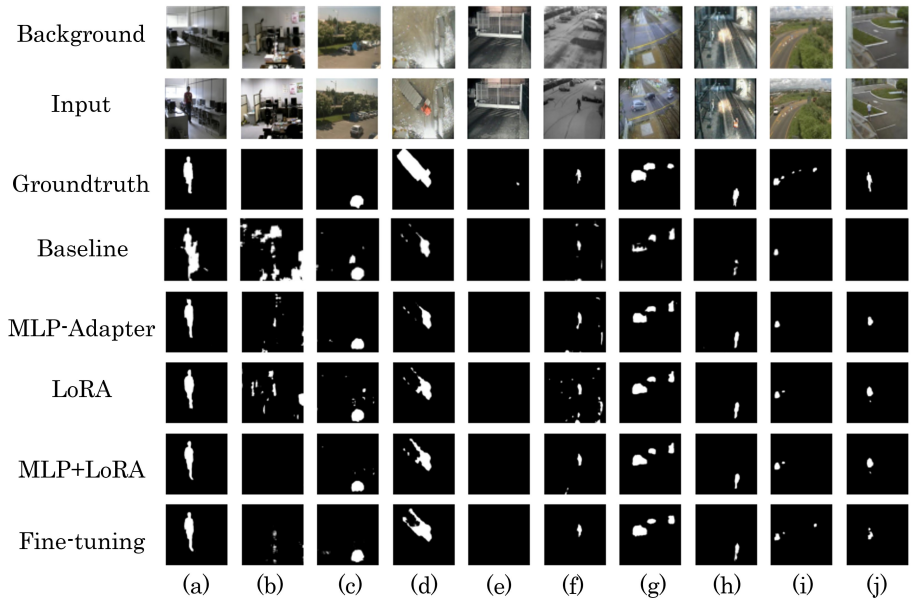
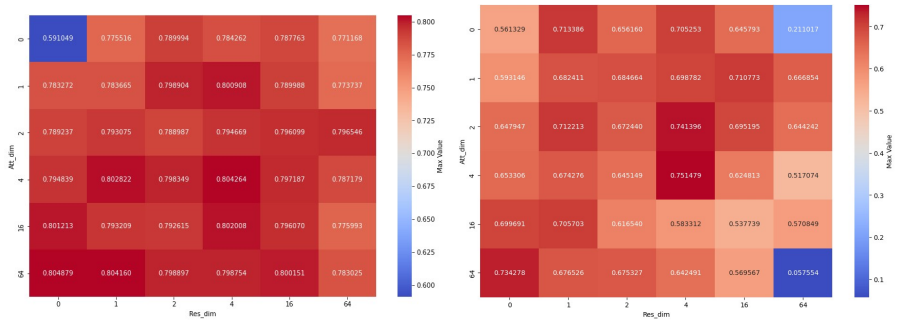


Fig. 5: The results of retraining by various methods
(a)Lasiesta, (b)LIMU LightSwitch, (c)Video001, (d)Video002, (e)Video004, (f)Video005, (g)Video006, (h)Video007, (i)Video008, (j)Video009
Baseline means the result by baseline TransCD pre-trained by CDNet-2014

4.2 Analysis of internal dimension of MLP-Adapter

MLP-Adapter has an internal dimension size as a hyperparameter. In this section, we analyze the suitable dimension size of two MLP-Adapters; for MA and the residual connection. We selected 1, 2, 4, 16, and 64 as the candidate of internal dimension size. These internal dimensions are smaller than the input dimension of MLP-Adapter. We intended to store effective knowledge about the target scene in smaller parameters such as LoRA [11]. We executed retraining various internal dimension size MLP-Adapter for some datasets and evaluated the accuracy for target scenes. We retrained MLP-Adapter models with two different settings for the number of training images. We selected BMC Real Video008 as the training dataset. It includes simple dynamic background; swaying plants. We selected it because the baseline model cannot detect changes accurately despite the simplicity of background change. First, we retrained MLP-Adapter models with many training images as the experiments in Sec. 4.1; the first half of the dataset. Second, we retrained MLP-Adapter models with a few images as few-shot learning. We selected two training images from each dataset. Two training images contain one image and another different background scene image. We set learning late 0.0002 and retrained in 100 epochs. We also constructed one-MLP-Adapter models to analyze the effect of introduced MLP-Adapter. One-



(a) Retraining with many images **(b)** Retraining with two images

Fig. 6: Heatmap of F1 score for BMC real Video008

MLP-Adapter model has one MLP-Adapter either for MA or for the residual connection.

Fig. 6 shows the heatmap of the F1 score by retraining various internal dimension MLP-Adapters with BMC Real Video008. The vertical axis means the internal dimension of MLP-Adapter for MA, and the horizontal axis means that of MLP-Adapter for the residual connection. 0 dimension in each axis means introducing no MLP-Adapter to TransCD; for example, the (0,4) model has only one MLP-Adapter for the residual connection, not one for MA. (0,0) means the result of baseline TransCD. The heatmap scale of Fig. 6a is different from that of Fig. 6b.

Fig. 6a shows the result of retraining with many images. All MLP-Adapter models with every combination of dimensions improved the F1 score from the baseline model. There was no significant difference in F1 between the different dimension sizes; The gap between the best F1 (64,0) and the worst F1 (0,64) was less than 0.035. Fig. 6b shows the result of retraining with two images. MLP-Adapter with combination of small-dimension sizes such as (4,4) and (2,4) had higher F1 scores than the combination of large-dimension sizes such as (16,16) and (64,64); (64,64) model failed to improve accuracy from the baseline model.

From these results, when retraining with many images, MLP-Adapter with every dimension size demonstrated better performance than the baseline. In contrast, when retraining with two images, MLP-Adapter with the small-dimension size improved the performance stably, while some of the MLP-Adapter with large-dimension size failed to improve accuracy. In addition, retraining small-dimension size of MLP-Adapter needs smaller computation than large-dimension size of MLP-Adapter. Therefore, the small dimension MLP-Adapter is suitable for retraining. However, the smallest dimension size is not always the best combination; the best combination in retraining with two images was (4,4), not such as (1,1). Therefore, we need to select an appropriate combination of two MLP-Adapters depending on the target scenes and retraining method.

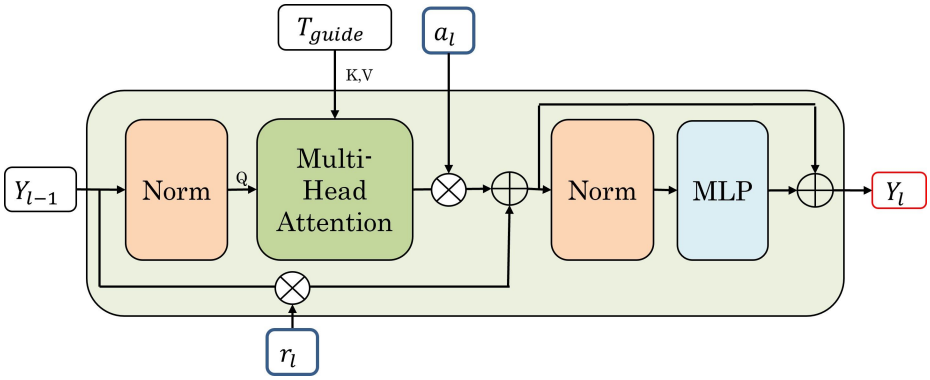


Fig. 7: Model of TransCD decoder with scalar adapters

4.3 Introducing scalar parameter adapter to change detection ViT

In this section, we analyze the contribution of adapter for MA and the residual connection. We introduced scalar parameters as the smallest adapter to MA and the residual connection. We confirmed whether the introduction scalar adapter to MA and the residual connection can get the knowledge of the target scene. Fig. 7 shows the decoder with two scalar parameter adapters. We introduced a scalar parameter adapter a_l after MA and another scalar parameter adapter r_l to the residual connection. We multiplied the output of MA by a_l and multiplied the residual connection r_l. These processes mean introducing one dimension MLP-Adapter in Fig. 4 like Eq. (11).

$$\text{Adapter}(X) := cX \tag{11}$$

In this experiment, we retrained the scalar parameter adapter for untrained illumination change scenes. We also verified the difference in detection accuracy depending on the number and content of untrained datasets. We selected two datasets in terms of the difference in strength of illumination change. We used Lasiesta I_IL_02 and LIMU Camera Parameter Changes in Indoor Scene [7] as untrained datasets. Lasiesta includes weak illumination changes depending on the presence and absence of sunshades. LIMU CameraParameter includes strong illumination changes depending on the light switch. We used the following four types of datasets as concerns the number of datasets and their content, 100-imgs, 2-imgs, Dark, and Bright. 100-imgs has one hundred images containing both dark and bright scenes. 2-imgs has one dark scene and one bright scene. Dark is one dark scene. Bright is one bright scene. Tuning with 100-imgs implies adjusting with enough training images as conventional fine-tuning. We used 2-imgs as a few-shot learning, and Dark and Bright as one-shot learning. We set the learning rate to 0.1, and the number of epochs to 20. Each evaluation dataset contained

[7] https://limu.ait.kyushu-u.ac.jp/dataset/en/

Table 3: Accuracy of retraining scalar adapter and fine-tuning with various setting of dataset

		Laciesta			LIMU CameraParameter		
Method	Datasets	F1	Precision	Recall	F1	Precision	Recall
baseline	-	0.5703	0.4632	0.7418	0.112	0.062	0.607
Fine-tuning	100-imgs	**0.9412**	0.9556	0.9271	**0.889**	0.897	0.881
Fine-tuning	2-imgs	0.9100	0.9301	0.8907	0.297	0.483	0.214
Fine-tuning	Dark	0.7359	0.6284	0.8877	0.180	0.103	0.713
Fine-tuning	Brignt	0.8866	0.9298	0.8472	0.387	0.482	0.323
Scalar-tuning	100-imgs	0.7643	0.8493	0.6947	0.227	0.217	0.238
Scalar-tuning	2-imgs	**0.7768**	0.9623	0.6513	0.327	0.491	0.246
Scalar-tuning	Dark	0.7119	0.7891	0.6485	0.134	0.105	0.185
Scalar-tuning	Bright	0.7726	0.9561	0.6482	**0.402**	0.700	0.282

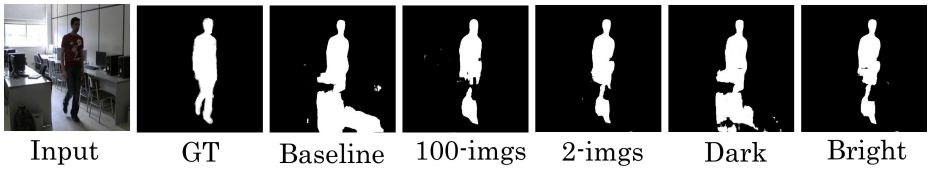


Fig. 8: Results of change detection to Lasiesta by retraining scalar adapters

images before and after the illumination change. We also did fine-tuning baseline TransCD as the comparing method.

Tab. 3 summarizes the accuracy of change detection to Lasiesta and LIMU CameraParamter by fine-tuning and by tuning scalar adapter with the four datasets. In the results of Lasiesta, tuning scalar adapter with both training datasets recorded a higher F1 score than the baseline model. Scalar-tuning with 2-imgs had the best accuracy in Scalar-tuning, followed by Scalar-tuning with Bright which achieved an almost similar accuracy. Scalar-tuning with Dark resulted in an approximately 0.06 lower F1 score than the other Scalar-tuning. Fig. 8 shows the detection results of each Scalar-tuning. Scalar-tuning except for Dark scene did not detect illumination changes and shadows on the floor.

In the results of LIMU CameraParamter, Scalar-tuning except for with Dark improved the F1 score by more than 0.1. Scalar-tuning with Dark exhibited little improvement in the F1 score. Compared to fine-tuning, Scalar-tuning with 100-imgs was significantly less accurate than fine-tuning. However, Scalar-tuning with 2-imgs and Bright had a better F1 score than that of fine-tuning. For all datasets, the accuracy of the scalar adapter models was inferior to fine-tuning. Comparing the result of 100-imgs and that of Bright in LIMU CameraParameter, training with multiple datasets reduced the accuracy of the scalar parameter adapter model. Training with 1 scene could improve accuracy for the training scene. Fig. 9 shows the result of change detection to LIMU CameraParameter. The baseline detected the brightness of bright input mistakenly. However,

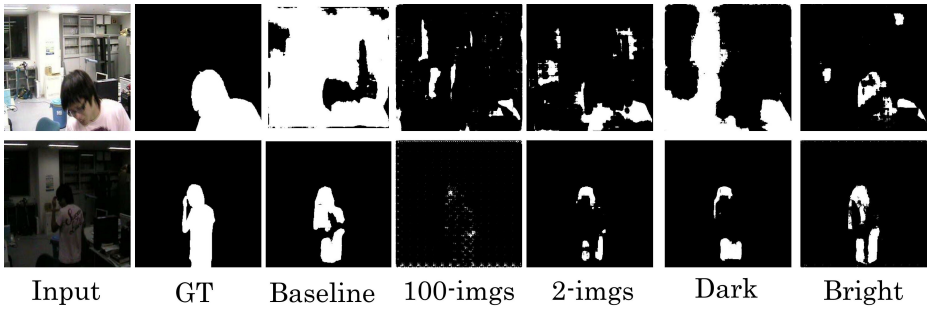


Fig. 9: Results of change detection to LIMU CameraParameter by retraining scalar adapters

training with Bright suppressed to detect the brightness. From the results, a scalar adapter can get the knowledge of one illumination change and improve the change detection for the learned scene. Bright dataset has only one bright scene, so the scalar adapter may get the knowledge of the bright scene without competition by dark scenes.

5 Conclusion

This study analyzed the introduction of the various learnable parameters to change detection ViT. MLP-Adapter for MA and the residual connection and LoRA effectively improve accuracy for untrained scenes by tuning additional parameters, and the combination of MLP-Adapter and LoRA is available. The small internal dimension size is suitable for MLP-Adapter. From the experiments with scalar parameters, we showed that the introduction of adapter to MA and the residual connection is reasonable. Adapter needs more than two dimension sizes of parameters to adapt multiple scenes before and after background scenes. In the future, we will construct an efficient method for selecting and introducing appropriate additional parameters for the target scene. In this study, we used pre-trained TransCD and tuned additional parameters with the pre-trained TransCD. We should evaluate training TransCD from scratch to the target scene and the limits of the methods of additional parameters. In addition, we compared MLP-Adapter to LoRA and Fine-tuning only. To gain deeper insights, we further evaluate the difference between our method and other previous adapter models.

Acknowledgements

This work was supported by JST CREST Grant Number JPMJCR22D1, JSPS KAKENHI Grant Number JP22H00551, and JST, PRESTO Grant Number JPMJPR236A, Japan.

References

1. Bouwmans, T., Javed, S., Sultana, M., Jung, S.K.: Deep neural network concepts for background subtraction:a systematic review and comparative evaluation. Neural Networks **117**, 8–66 (2019). https://doi.org/https://doi.org/10.1016/j.neunet.2019.04.024, https://www.sciencedirect.com/science/article/pii/S0893608019301303
2. Chen, H., Qi, Z., Shi, Z.: Remote sensing image change detection with transformers. IEEE Transactions on Geoscience and Remote Sensing **60**, 1–14 (2022). https://doi.org/10.1109/TGRS.2021.3095166
3. Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., Luo, P.: Adaptformer: Adapting vision transformers for scalable visual recognition. Advances in Neural Information Processing Systems **35**, 16664–16678 (2022)
4. Cuevas, C., Yáñez, E.M., García, N.: Labeled dataset for integral evaluation of moving object detection algorithms: Lasiesta. Computer Vision and Image Understanding **152**, 103–117 (2016)
5. Cui, H., Lv, Z., Yuan, T., Feng, C., Shan, X.: Gibs-net: Unseen video background subtraction with global information. In: 2023 IEEE 11th Joint International Information Technology and Artificial Intelligence Conference (ITAIC). vol. 11, pp. 125–130 (2023). https://doi.org/10.1109/ITAIC58329.2023.10408949
6. Culibrk, D., Marques, O., Socek, D., Kalva, H., Furht, B.: A neural network approach to bayesian background modeling for video object segmentation. In: VISAPP (2006)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
8. Goyette, N., Jodoin, P.M., Porikli, F., Konrad, J., Ishwar, P.: Changedetection.net: A new change detection benchmark dataset. In: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. pp. 1–8 (2012). https://doi.org/10.1109/CVPRW.2012.6238919
9. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016)
10. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for nlp. In: International Conference on Machine Learning. pp. 2790–2799. PMLR (2019)
11. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
12. Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. In: European Conference on Computer Vision. pp. 709–727 (2022)
13. Jodoin, P.M., Maddalena, L., Petrosino, A., Wang, Y.: Extensive benchmark and survey of modeling methods for scene background initialization. IEEE Transactions on Image Processing **26**(11), 5244–5256 (2017). https://doi.org/10.1109/TIP.2017.2728181
14. Kenton, J.D.M.W.C., Toutanova, L.K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT. pp. 4171–4186 (2019)

15. Lin, H.H., Liu, T.L., Chuang, J.H.: A probabilistic svm approach for background scene initialization. In: Proceedings. International Conference on Image Processing. vol. 3, pp. 893–896 vol.3 (2002). https://doi.org/10.1109/ICIP.2002.1039116
16. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021)
17. McFarlane, N.J., Schofield, C.P.: Segmentation and tracking of piglets in images. Machine vision and applications **8**, 187–193 (1995)
18. Osman, I., Abdelpakey, M., Shehata, M.S.: Transblast: Self-supervised learning using augmented subspace with transformer for background/foreground separation. In: 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW). pp. 215–224 (2021). https://doi.org/10.1109/ICCVW54120.2021.00029
19. Rebuffi, S.A., Bilen, H., Vedaldi, A.: Learning multiple visual domains with residual adapters. Advances in neural information processing systems **30** (2017)
20. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149). vol. 2, pp. 246–252 Vol. 2 (1999). https://doi.org/10.1109/CVPR.1999.784637
21. Tezcan, O., Ishwar, P., Konrad, J.: Bsuv-net: A fully-convolutional neural network for background subtraction of unseen videos. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2774–2783 (2020)
22. Vacavant, A., Chateau, T., Wilhelm, A., Lequièvre, L.: A benchmark dataset for outdoor foreground/background extraction. In: Park, J.I., Kim, J. (eds.) Computer Vision - ACCV 2012 Workshops. pp. 291–300. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
23. Wang, Y., Jodoin, P.M., Porikli, F., Konrad, J., Benezeth, Y., Ishwar, P.: Cdnet 2014: An expanded change detection benchmark dataset. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 393–400 (2014). https://doi.org/10.1109/CVPRW.2014.126
24. Wang, Z., Zhang, Y., Luo, L., Wang, N.: Transcd: scene change detection via transformer-based architecture. Opt. Express **29**(25), 41409–41427 (Dec 2021). https://doi.org/10.1364/OE.440720, https://opg.optica.org/oe/abstract.cfm?URI=oe-29-25-41409
25. Zhang, C., Wang, L., Cheng, S., Li, Y.: Swinsunet: Pure transformer network for remote sensing image change detection. IEEE Transactions on Geoscience and Remote Sensing **60**, 1–13 (2022). https://doi.org/10.1109/TGRS.2022.3160007