TinyCD: a (not so) Deep Learning Model for Change Detection

Andrea Codegoni
Dipartimento di Matematica "F. Casorati"
University of Pavia
andrea.codegoni01@ateneopv.it

Gabriele Lombardi, Alessandro Ferrari
ARGO Vision
Milano
{gabriele.lombardi,alessandro.ferrari}@argo.vision

Abstract

The aim of change detection (CD) is to detect changes occurred in the same area by comparing two images of that place taken at different times. The challenging part of the CD is to keep track of the changes the user wants to highlight, such as new buildings, and to ignore changes due to external factors such as environmental, lighting condition, fog or seasonal changes. Recent developments in the field of deep learning enabled researchers to achieve outstanding performance in this area. In particular, different mechanisms of space-time attention allowed to exploit the spatial features that are extracted from the models and to correlate them also in a temporal way by exploiting both the available images. The downside is that the models have become increasingly complex and large, often unfeasible for edge applications. These are limitations when the models must be applied to the industrial field or in applications requiring real-time performances. In this work we propose a novel model, called TinyCD, demonstrating to be both lightweight and effective, able to achieve performances comparable or even superior to the current state of the art with 13-150X fewer parameters. In our approach we have exploited the importance of low-level features to compare images. To do this, we use only few backbone blocks. This strategy allow us to keep the number of network parameters low. To compose the features extracted from the two images, we introduce a novel, economical in terms of parameters, mixing block capable of cross correlating features in both space and time domains. Finally, to fully exploit the information contained in the computed features, we define the PW-MLP block able to perform a pixel wise classification. Source code, models and results are available here: `https://github.com/AndreaCodegoni/Tiny_model_4_CD`

***K*eywords** Change Detection (CD) · Remote Sensing (RS) · Convolutional Neural Network (CNN)

1 Introduction

In the Remote Sensing community, Change Detection (from now denoted with CD) is one of the main research topics. The main purpose of CD is to construct a model able to identify changes occurred in a scene between two different times. To this aim, a CD model compares two co-registered images acquired at times t_1 and t_2 [1]. Once the relevant changes have been identified, such as urban expansion, deforestation or post disaster damage assessment [2–7], the challenge is to let the CD model ignore other irrelevant changes. Examples of irrelevant changes are, but not limited to, light conditions, shadows, and seasonal variations.

Thanks to the increasing number of available high resolution aerial images datasets, such as [2,6,7], data driven methods like deep Convolutional Neural Networks (CNN) found successful applicability [8]. The well known ability of deep CNNs to extract complex and relevant features from images is the key factor for their early promising results [9]. In the CD scenario, complex features are important, but are not sufficient to accomplish the task. To highlight the occurred changes it is in fact crucial to model the spatio-temporal dependencies between the two images. Plain CNNs have as drawback a limited receptive field caused by the use of fixed kernels in convolutions. To overcome this issue, recent works has focused their attention to enlarging the receptive fields by using different kernel types [10], or adding attention mechanisms [2, 6, 11, 12, 12–14]. However, most of them failed to explicitly relate data in the temporal domain, since attention mechanisms are applied separately on the two images. The self attention mechanism adopted in [2, 14] shows promising results relating images spatio-temporally, but do that in a very computationally inefficient way. More recently, Transformers have been introduced in CD because of their receptive fields spatially covering the whole image [15, 16]. Notice that, by applying multi-headed attention layers in the decoder part of the network, the receptive field covers the temporal domain too.

The CD field finds applicability also outside the remote sensing world. Our work is inspired by several industrial needs that we are facing. In that field one additional constraint is to keep low the complexity of the model, thus reducing machine response times. Unfortunately, the majority of State-Of-The-Art (SOTA) models proposed in literature are millions-parameters-sized. In industrial scenarios, using such big models is not always possible. The first issue with those models is related to the training phase. Training-time is clearly affected by the size of the model, thus making model tuning and Hyper-Parameters-Optimization (HPO) times incompatible with the resources available in medium-small companies. Moreover, the bigger the network, the bigger the datasets needed to prevent overfitting the data. The dataset size is a common issue in industrial applications where collecting data is a time-consuming and costly task. Finally, big networks require dedicated hardware both for training and inference. This is in contrast with production requirements and project budgets.

The main purpose of our work is to develop a neural network having comparable performances with respect to the SOTA CD models, but requiring a lower computational complexity.

The majors contributions of our work are the following:

- We explore the effectiveness of using low-level features in the problem of comparing images. This approach allowed us to validate our intuition that in this context the low-level features are sufficiently expressive. Moreover, this allowed us to significantly limit the number of parameters of our model.
- We introduce a novel strategy to mix the features between the two images. To this aim, we leverage the grouped convolutions and a specific type of concatenation. The mixing block operates a spatio-temporal cross-correlation between the pixels of the two input images.
- A novel block called MAMB is introduced to produce skip connections. More precisely, we define an attention mechanism that uses pixel-wise information to retrieve a skip connection mask on a per-resolution basis. The obtained masks are exploited in the up sampling phase to refine the carried information.
- To face the problem of per-pixel classification, in the last block we use the obtained channel wise information to generate the final mask by classifying each pixel separately.

Our architecture exploits the information contained in the channels of the feature vectors generated by the backbone. For this reason, it can effectively exploit low level features such that a relative small backbone can be adopted. Being the backbone[1] the most time-consuming and parameter-demanding component in the architecture, maintaining it small allows us to achieve our goal. In particular, this allows us to maintain the total number of parameters below 0.3 millions.

Finally, we compare the quality of the model with SOTA architectures, and we demonstrate that it has performances comparable if not even superior to other SOTA models in the CD field. We have extensively tested our model on public and proprietary datasets. In order to validate and make reproducible our results, in this paper we highlight the results obtained in the field of aerial images on public datasets. Similar results in terms of efficiency and effectiveness have been found in non-public datasets, in application fields other than that which is the subject of this paper.

The paper is organized as follows: in Chapter 2 we present some related works; in Chapter 3 we describe our proposed model; in Chapter 4 we report the results of our model on two public available datasets, and in Chapter 5 we highlight some future research directions.

2 Related works

Deep learning models, and in particular CNN, have been applied with great success in image comparison tasks [17–19], in pixel-level image classification [20–22], and they represent the SOTA in many other Computer Vision tasks [23].

Models in the context of the CD must manage two inputs: one image acquired at time t_1, and another one acquired at time t_2. The correct use of these two inputs and the features extracted from them are extremely important for the well behavior of the CD model. To the best of our knowledge, the first work that applied CNN to the CD problem is [9]. In this work the authors propose two different approaches. In the first case they use a U-Net [21] type network with

[1] Notice that the backbone is evaluated twice in Siamese architectures.

the Early Fusion Strategy (FC-EF), i.e. they concatenate the two images taken at times t_1 and t_2, and then they feed the U-Net with the concatenated tensor. In the second case they use a Siamese U-Net type network [18, 22, 24] where the two images are processed separately, and subsequently the features are fused in two different ways: concatenation (FC-Siam-conc) and subtraction (FC-Siam-diff). These fused features are then used as skip connections in the decoder. After this seminal work, an entire research line investigated both the Early Fusion Strategy [3, 12, 25, 26], and the Feature Fusion Strategy [2, 6, 10, 11, 13, 14, 27–32].

To take full advantage of the large amount of spatial information, deeper CNNs such as ResNet [33] or VGG16 [34] have been used [2, 6, 10, 14] in order to extract spatial information and group them in a hierarchical way. Although, standard convolution has a fixed receptive field that limits the capacity of modelling the context of the image. To face this issue, atrous convolutions [35] have been experimented [10]: they are able to enlarge the receptive field of the single convolutional kernel without increasing the number of parameters. To definitively overcome the problem of fixed receptive field, attention mechanisms, in the form of spatial attention [6, 11, 12], channel wise attention [6, 11–13], and also self-attention [2, 14], have been introduced. In [6], the attention mechanisms are used in the decoder part: the channel wise attention is used to re-weight each pixel after the fusion with the skip connections, while the spatial attention is adopted to spatially re-weight the pixels containing misleading information due to the up sampling step. To further exploit the interconnection between spatial and channel information, in [11] a dual attention module is introduced. The co-attention module introduced in [13] tries to leverage correlation between features extracted from both images. Also in [13] a co-layer aggregation and a pyramid structure is used to make full use of the features extracted at each level and with different receptive fields. In [2], the non-local self attention introduced in [36], is used. This mechanism consists in stacking the features extracted from a Siamese backbone to apply them both a basic spatial attention mechanism, and a pyramidal attention mechanism. Since these two attention blocks are applied to stacked t_1 and t_2 features, these are correlated in a non-local spatio-temporal way.

Finally, we mention that the global attention mechanisms introduced with Transformers [37, 38] have also been applied to the CD problem. In [15] the authors employ a modified ResNet18 as Siamese backbone to extract features. Then, to better justify the use of Transformer blocks, they follow a parallelism between the natural language processing field, and the image processing one, by introducing the semantic tokens. Roughly speaking, semantic tokens are the pixels of the last feature tensor extracted by the backbone. The authors use this concept to illustrate that concatenating single pixels and then processing them with a transformer encoder-decoder, a pair of features tensors can be obtained that incorporates both global spatial information, and global temporal information. On the other side, in [16] the authors replace the CNN backbone with a transformer in order to exploit the global information contained in the images right from the start. In this model, the temporal aggregation is done only in the final multilayer perceptron decoder.

Our work is inspired by [15]. In fact, we believe that the information contained in each single pixel at different resolutions, i.e. the semantic tokens, are essential for the final classification, and also they are more important than the global context provided by spatial attention. Moreover, our intuition is that, since we can compare two images, we can use only low level features to highlight the changes occurred in time. To these extents, we designed a Siamese U-Net type network where the backbone is represented by the first 4 blocks of EfficientNetb4 [39]. To better fuse the information contained in channels in a spatio-temporal way, we introduce a mixing strategy that forces the network to fuse and compare features in a semantically consistent way between the features extracted at times t_1 and t_2. Finally, to exploit all the information contained in each pixel/semantic token, we heavily applied multi layer perceptrons (MLP) on each pixel/semantic token. MLP are used to create spatial attention mask skip connections in the U-Net structure. Moreover, they have been applied in the classification block to classify each pixel.

3 Proposed model

In this section we describe and motivate the structure of our model. We use a Siamese U-Net like model consisting of 4 main components:

- Siamese encoders constituted by a pre-trained backbone (see Section 3.2).
- Mix and Attention Mask Block (MAMB) and bottleneck mixing block to compose backbone results (see Section 3.3).
- Up-sample decoder to refine low resolution results incorporating higher resolution data from the skip connections (see Section 3.4).
- Pixel level classifier (see Section 3.5)

In what follows, we denote with $X \in \mathbb{R}^{(C \times H \times W)}$ the *reference* tensor (image at time t_1) and with $Y \in \mathbb{R}^{(C \times H \times W)}$ the *comparison* tensor (image at time t_2). C is the number of channels, H is the height and W the width of the tensors.

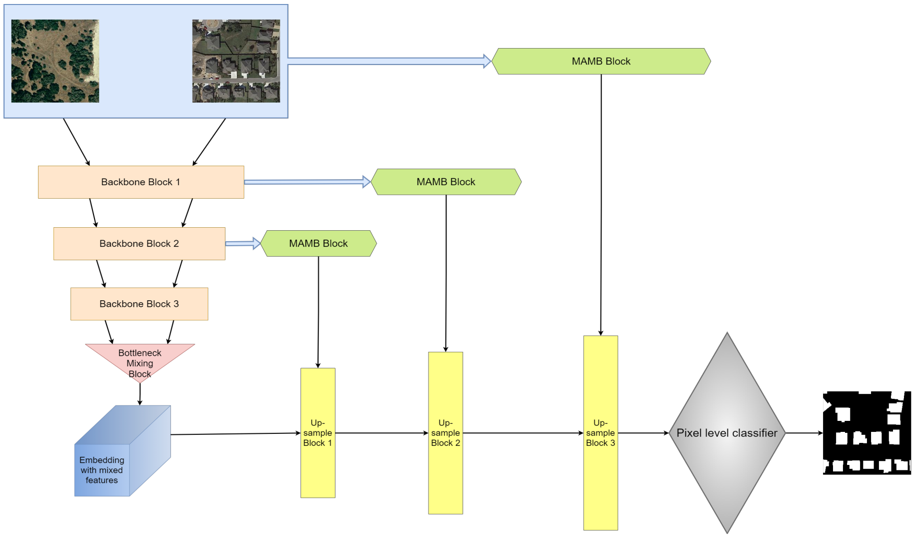

Figure 1: Siamese U-Net like architecture including MAMB.

We omit the batch dimension for the ease of notation. We denote with Conv the convolution operator, with PReLU the Parametric Rectified Linear Unit [40], with $\mathrm{IN2d}$ the Instance Normalization [41], and with $\mathrm{Sigmoid}$ the Sigmoid function.

3.1 Model overview

Indicating with f_k the composition of the backbone blocks up to the k^{th} one, the high-level features $X_k = f_k(X)$ and $Y_k = f_k(Y)$ are extracted from each level k of the backbone. These features are used both to compute the resulting output at each level of the U-Net encoder, and to estimate the attention masks. The last backbone block produces the embeddings X_e and Y_e representing the bottleneck inputs.

Every backbone intermediate output pair (X_k, Y_k) is processed by means of the MAMB producing spatial attention masks M_k. These masks are used as skip-connections and composed in the decoder. The last mixed tensor is obtained by composing (X_e, Y_e).

The decoder consists of a series of up-layers, one for each block of the backbone. Each up-layer increases the spatial dimensions of the tensor received from the previous layer to reach the same resolution of the corresponding skip-connection. Furthermore, the up sampled tensors and the skip-connections are composed to generate the next layer inputs. This composition is the attention mask application to the features obtained from the previous layer.

Finally, the last block of our model classifies each pixel of the obtained tensor through a Pixel-Wise Multi-Layer Perceptron (PW-MLP). The PW-MLP associates to each pixel the probability that it belongs to the anomaly class. Applying a threshold to this tensor we obtain the binary mask of changes.

In the following subsections we describe each component separately.

3.2 Siamese encoders with pre-trained backbone

The purpose of the Siamese encoder is to extract simultaneously features from both images in a semantic coherent way. In deep neural networks, training the first layers of the model is sometimes difficult due to the well-known phenomenon of vanishing gradients [42, 43]. To overcome this problem, several tricks have been introduced such as the residual connections of ResNet [33] or the skip connections of U-Net [21]. However, training deep backbones remains a difficult, time-consuming, or even impossible task to accomplish if the dataset is too small.

For these reasons, pre-trained backbones are often preferred, even in CD problems [2, 6, 10, 14, 15]. The disadvantage of this approach is that the backbones are not always trained on images that are similar to the ones we are dealing with. However, CNN backbones work by layering information. Low-level features, such as lines, black/white spots, points, edges, can be considered general-purpose being common to all images.

In our intuition, in the faced task the comparison between two images acquired at times t_1 and t_2 can be accomplished by using just the low-level features extracted from the first few layers of a pre-trained backbone.

We therefore decided to use one EfficientNet backbone [39] pre-trained on the ImageNet dataset [44]. We allowed the training phase to tune all the backbone parameters during the update of all the other network parameters. We have chosen the EfficientNet backbone family due to both its efficacy and its efficiency. Moreover, the resolution reduction in the first EfficientNet layers is sufficiently slow in order to create skip connections of different spatial dimensions.

3.3 Mix and Attention Mask Block (MAMB) and bottleneck mixing block

The purpose of this block is to merge the features (X_k, Y_k) extracted from one of the blocks of the Siamese encoder. It creates a mask M_k that is then used as skip connection to refine the information obtained during the up sampling phase.

The mask we create can also be understood as a pixel-level attention mechanism. The idea of pixel-wise attention has been already studied in [45]. Here we specifically designed a pixel-wise attention mechanism exploiting both spatial and temporal information.

The MAMB can be divided into two sub-blocks: the Mixing block (see Section 3.3.1), and the Pixel level mask generator (see Section 3.3.2).

3.3.1 Mixing block

As the name suggests, in this sub-block we compose the features generated by the k^{th} backbone block (X_k, Y_k). To this aim, we observe that the features X_k and Y_k, share both the same shape $C_k,\ H_k,\ W_k$, and the same arrangement in terms of features. This means that the features in channel c of X_k have the same semantic meaning with respect to the corresponding features in channel c of Y_k, being the Siamese encoder weights shared. In view of this observation, we decided to concatenate the tensors X_k and Y_k in the tensor $Z_k \in \mathbb{R}^{2C_k \times H_k \times W_k}$ using the following rule:

$$Z_k^c := \begin{cases} X_k^{c/2} & \text{if } c \text{ is even} \\ Y_k^{(c-1)/2} & \text{otherwise} \end{cases} \qquad \forall c \in \{0, 1..., 2C_k - 1\}. \tag{1}$$

To mix the features coming from X_k and Y_k both spatially and temporally, we used a group convolution. By choosing the number of groups equal to C_k we obtain C_k kernels of depth 2 which process the tensor Z_k in pairs of channels. These kernels perform at the same time both spatial and temporal convolution using the cross-correlation between semantically similar features.

The new tensor $Z'_k \in \mathbb{R}^{C_k \times H_k \times W_k}$ is defined as:

$$Z'_k = \text{Mix}(X_k, Y_k) := \text{PReLU}\left[\text{IN2d}\left[\text{Conv}(Z_k,\ ch_{in} = 2C_k,\ ch_{out} = C_k,\ groups = C_k)\right]\right]. \tag{2}$$

An illustration of our concatenation strategy, and the following grouped convolution, is reported in Figure 2.

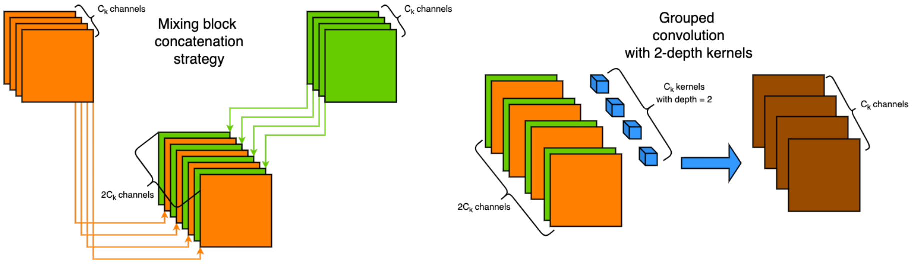
C_k channels
Mixing block concatenation strategy
C_k channels
$2C_k$ channels
Grouped convolution with 2-depth kernels
C_k kernels with depth = 2
C_k channels
$2C_k$ channels

Figure 2: Visual representation of our concatenation strategy (1) and the grouped convolution (2).

3.3.2 Pixel-level mask generator

Fixing the spatial coordinates of a single pixel, the C_k values in the tensor Z'_k contain spatial information related to both times t_1 and t_2. Our idea is to use the PW-MLP in order to process this information and generate a score that acts as a spatio-temporal attention.

To this aim, the PW-MLP is designed to produce a mask tensor $M_k \in \mathbb{R}^{H \times W}$.

3.3.3 PW-MLP

To implement a pixel-wise Multi-Layer Perceptron, that is an MLP working on all the channels of one single pixel at a time, we use 1×1 convolutions. The MLP is composed by N blocks each containing one 1×1 convolution and one activation function. As activation, we used the PReLU, being this able to propagate gradients also on the negative side of the real axis. The last convolution contains just one filter, thus producing a tensor M_k with dimensions $1,\ H_k,\ W_k$.

The use of 1×1 convolutions to implement an MLP is not a new idea. In [46] this strategy has been used to substitute layers such as convolutions with small, trainable, networks. As pointed out in [46], we have very poor prior information on the latent concepts in pixel vectors. Hence, we have decided to use this universal function approximator to separate different semantic concepts.

3.3.4 The bottleneck mixing block

We applied the tensor mixing strategy reported in Section 3.3.1 to compute the bottleneck of the U-Net like network. More precisely, we compute: $U_e = Z'_e = \mathrm{Mix}(X_e, Y_e)$.

U_e represents the output of the encoder and the input to be processed by the decoder. U_e contains the higher level features extracted by the backbone and correlated both spatially and temporarily. Given that, our intuition is that U_e contains enough information in order to classify each pixel at the bottleneck resolution.

3.4 Up-sample decoder with skip connections

The general $k-$th decoder block takes as input the tensor U_{k+1} of shape $C_{k+1}, H_{k+1}, W_{k+1}$ and a mask M_k of shape $1, H_k, W_k$ generated by one MAMB. Firstly, an up sampling operation is performed in order to transform U_{k+1} so that its shape matches the one of M_k. We call the up sampled tensor U'_k. Then, we define U_k with

$$U_k = \mathrm{Up}(U_{k+1}, M_k) := \mathrm{PReLU}\left[\mathrm{IN2d}\left[\mathrm{Conv}(U'_k \odot M_k)\right]\right],$$

where we have denoted with the symbol $\odot$ the Hadamard product. This represents the skip connection attention mechanism at the pixel level.

As we already mentioned in Section 3.3.4, U_e contains enough information to classify each pixel at its spatial resolution. By multiplying the mask M_k we are re-weighting each pixel in order to alleviate the misleading information generated by up sampling.

Notice that, in this Up block we employ the depth wise separable convolution [47, 48].

3.5 Pixel level classifier

Finally, since the change detection problem is a binary classification problem, we decided to use as last layer a PW-MLP with output classes $\{0, 1\}$ representing respectively normal and change pixels. With respect to what reported in Section 3.3.3, in this case we used as the last activation layer a $\mathrm{Sigmoid}$ function instead of the PReLU, thus enforcing the result of the network to contain values in $[0, 1]$. In this case, the PW-MLP is used as a non-linear classifier which separates pixels in normal or changed.

4 Experiment Settings and Results

In this section we present the settings used in our experiments, and we report the achieved results.

4.1 Datasets

As already stated in Section 1, we cannot share the dataset related to our industrial application. Moreover, in order to fairly evaluate our model, and to compare it with other works in the CD field, we used the following public and widely adopted aerial images building datasets: LEVIR-CD [2] and WHU-CD [7][2]. Notice that the task defined by these datasets is particularly close to the faced industrial one, that is the driver of our research work. In these two datasets the model has to track some specific patters, those corresponding to buildings, and carefully segments the eventually occurred changes.

LEVIR-CD contains 637 pairs of high resolution aerial images. Starting from these images, patch pairs of size 256×256 each have been extracted. After that, the pair instances have been partitioned accordingly to the authors'

[2] Both the adopted dataset have been obtained from `https://github.com/wgcban/SemiCD` in an already pre-processed version.

original indications. This step produced 7120, 1024, and 2048 pair instances for the train, validation, and test dataset respectively.

WHU-CD contains just one pair of images having resolution 32507×15354 as a crop of a wider geographic area[3]. Following [49], the images have been split in non overlapping patches with resolution 256×256. After that, a randomly partitioning of the dataset have been performed obtaining 5947, 743, and 744 pairs for train, validation, and test respectively.

4.2 Implementation details

We implemented our model using PyTorch [50] and we trained it on an NVIDIA GeForce RTX 2060 6GB GPU. As described in Section 3.2, we selected the first four blocks of the EfficientNet version $b4$ backbone pretrained on the ImageNet dataset. All other weights of the model have been initialized randomly[4].

As optimizer we adopted AdamW [51], tuning its hyperparameters with the package Neural Network Intelligence (NNI) [52]. More precisely, we tuned the learning-rate, the weight decay, and the usage of the optimizer *amsgrad* variant. Due to computational resource limitations, no other hyperparameters have been tuned. We have not experimented any network architecture search technique (NAS). To dynamically adjust the learning rate during the training, we opted for the cosine annealing strategy as described in [53], but avoiding the warm restart.

Since aerial images are spatially registered, we applied as data augmentation both Random Flip and Random Rotation simultaneously to the reference/comparison images and their associated ground-truth mask. Moreover, Gaussian Blur and Random Brightness/Contrast changes have been applied independently on the reference and the comparison images respectively. [5]

Finally, due to the limited GPU memory capacity and computational power, we fixed the batch size to 8, and trained for just 100 epochs.

4.3 Loss function and evaluation metrics

Since the CD problem is a binary classification problem, we used the Binary Cross Entropy (BCE) loss function, namely:

$$\mathcal{L}(G, P) := -\frac{1}{|H| \cdot |W|} \sum_{h \in H, w \in W} g_{h,w} \log(p_{h,w}) + (1 - g_{h,w}) \log(1 - p_{h,w}),$$

where we denoted G the ground truth mask, P the model prediction, and with a little abuse of notation, H and W the set of indices relative to height and width and whose cardinalities are $|H|$ and $|W|$ respectively.

To evaluate the performances achieved by our model, we calculated the following indicators with respect to the change class:

$$\begin{aligned}
&\textbf{Precision} && Pr := \frac{TP}{TP + FP}, \\
&\textbf{Recall} && Rc := \frac{TP}{TP + FN}, \\
&\textbf{F1 score} && F1 := \frac{1}{Pr^{-1} + Rc^{-1}}, \\
&\textbf{Intersection over Union} && IoU := \frac{TP}{FN + FP + TP}, \\
&\textbf{Overall Accuracy} && OA := \frac{TP + TN}{FN + FP + TP + TN},
\end{aligned}$$

where TP, TN, FP, FN are computed on the change class and represent the true positives, true negatives, false positives, and false negatives respectively. To retrieve the change mask we applied a 0.5 threshold to the output mask.

[3] The whole dataset depicts the city of Christchurch, in New Zealand. The crop, aimed to be used in CD tasks, is a sub-area acquired in two different times.

[4] To make our results reproducible, we fixed the random seed at the beginning of each experiment.

[5] To achieve all the adopted augmentations, we used the Albumentation library [54].

4.4 Comparison with state-of-the-art

To demonstrate the effectiveness of our approach, we compared our results with those reported in [15, 16]. As baseline, we used the three models present in [9]. Moreover, to compare our model with other works adopting both spatial and channel attention mechanisms, we dealt with [2, 6, 11, 31]. Finally, given the success achieved by Transformers applied to the computer vision field, we also compared the results obtained in [15, 16].

Table 1: Performance metrics on the LEVIR-CD dataset. To improve results readability, we adopted a color ranking convention representing the First, Second, and Third results. The metrics are reported in percentage.

LEVIR-CD

Model	Pr	Rc	F1	IoU	OA
FC-EF [9]	86.91	80.17	83.40	71.53	98.39
FC-Siam-diff [9]	89.53	83.31	86.31	75.92	98.67
FC-Siam-conc [9]	91.99	76.77	83.69	71.96	98.49
DTCDSCN [11]	88.53	86.83	87.67	78.05	98.77
STANet [2]	83.81	91.00	87.26	77.40	98.66
IFNet [6]	94.02	82.93	88.13	78.77	98.87
SNUNet [31]	89.18	87.17	88.16	78.83	98.82
BIT [15]	89.24	89.37	89.31	80.68	98.92
Changeformer [16]	92.05	88.80	90.40	82.48	99.04
Ours	92.68	89.47	91.05	83.57	99.10

Table 2: Performance metrics on the WHU-CD dataset. To improve results readability, we adopted a color ranking convention representing the First, Second, and Third results. The metrics are reported in percentage.

WHU-CD

Model	Pr	Rc	F1	IoU	OA
FC-EF [9]	71.63	67.25	69.37	53.11	97.61
FC-Siam-diff [9]	47.33	77.66	58.81	41.66	95.63
FC-Siam-conc [9]	60.88	73.58	66.63	49.95	97.04
DTCDSCN [11]	63.92	82.30	71.95	56.19	97.42
STANet [2]	79.37	85.50	82.32	69.95	98.52
IFNet [6]	96.91	73.19	83.40	71.52	98.83
SNUNet [31]	85.60	81.49	83.50	71.67	98.71
BIT [15]	86.64	81.48	83.98	72.39	98.75
Ours	92.22	90.74	91.48	84.30	99.32

4.5 Discussion of results

The results reported in Table 1 and Table 2 show the superior performance of our model on these two building change detection datasets. The baseline models FC-Siam-diff and FC-Siam-conc [9] are probably the architectures most similar to ours. With respect to these two baseline models, we increased the F1 score by 4.73 points on LEVIR-CD, and by more than 20 points on the WHU-CD. With respect to the best model we found in the literature [16], the performance increment on the LEVIR-CD dataset is smaller. However, as we can see from Table 3, our model is 146.50 times smaller.

Table 3: Parameters, complexity and performance comparison. The metrics are reported in percentage, parameters in millions (M) and complexity in GigaFLOPs (G).

Model	Parameters (M)	Parameters ratio	FLOPs (G)	LEVIR-CD F1	WHU-CD F1
DTCDSCN [11]	41.07	146.67	7.21	87.67	71.95
STANet [2]	16.93	60.46	6.58	87.26	82.32
IFNet [6]	50.71	181.10	41.18	88.13	83.40
SNUNet [31]	12.03	42.96	27.44	88.16	83.50
BIT [15]	3.55	12.67	4.35	89.31	83.98
Changeformer [16]	41.02	146.50	N.D.	90.40	N.D.
Ours	0.28	1	1.54	91.04	91.48

In view of these results, we can conclude that our model, despite the lower complexity and the lower number of employed parameters, is very effective on the buildings CD task. Moreover, having not used any global attention mechanism, we have a confirmation of our intuitions: in the faced CD task, low level information is sufficient to reach high-quality results. Also, the information contained in each single pixel at different resolutions is very rich and can be exploited to effectively classify changes.

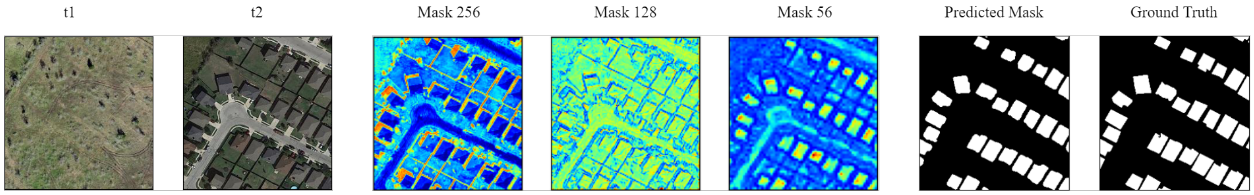

Figure 3: Visualization of the intermediate masks at different resolutions and the final binary mask for one example image pair.

In Figure 3 we reported an example of the intermediate masks that our model creates in skip-connections. As can be seen, the intermediate resolution masks have captured some semantics of the scene partially classifying the pixels' content. Hence, as can be seen comparing the ground truth binary mask (GT) with that generated by our model, the final result is very sharp and detailed. Moreover, unlike usual skip connections, our mechanism is lighter in terms of parameters, it does not impose major limitations on the number of used features, and finally it makes the role of skip connections more interpretable.

To quantitatively confirm the usefulness of skip connections, we trained a model without them and compared the achieved results in Table 4. As can be seen, all the metrics confirm the beneficial effects of skip connections in the model. Since the Recall index increased, we deduce that the skip connection model lowered the number of false negatives by better segmenting the various changes between the two images.

In Table 5 we compare our bottleneck mixing block described in Section 3.3.1 with other possible and simple feature fusion block. We replace the mixing block Section 3.3.1 with:

- Subtraction;
- Concatenation + Depth wise separable convolution;
- Concatenation + Convolution.

Our mixing strategy is a generalization of the pixel-wise subtraction[6]. However, our mixing block Section 3.3.1 is fully trainable with the spirit of feature re-use [55]. Also, concatenation with depth-wise separable convolution and standard convolution can be seen as generalizations of subtraction, but the number of parameters to achieve these generalizations is much bigger than ours. Generally speaking, the number of parameters needed by our mixing block is $c(2 \cdot k_h \cdot k_w)$,

[6] In fact, if we initialize all of our 2-depth kernels with the "central" weights to 1 and -1 and all the rest to 0, we have the standard subtraction.

Table 4: Performance comparison between the model with/without skip connections on both datasets LEVIR-CD and WHU-CD.

LEVIR-CD					
Model type	Pr	Rc	F1	IoU	OA
No Skip	92.35	88.50	90.38	82.45	99.04
Skip	92.68	89.47	91.05	83.57	99.10
WHU-CD					
Model type	Pr	Rc	F1	IoU	OA
No Skip	90.56	89.77	90.16	82.09	99.22
Skip	92.22	90.74	91.48	84.30	99.32

where c is the number of channels, k_h, k_w are the convolutional kernel sizes. The parentheses are highlighting the size of each kernel and the number of kernels. By comparison, a convolution working on the concatenated feature tensors requires $c(2c\cdot k_h\cdot k_w)$ parameters. Even with a more cheap depth wise separable convolution we have $2c(\cdot k_h\cdot k_w)+c(2c)$ parameters, that means $2c^2$ more than our proposed strategy. In addition to the savings on the number of parameters, from Table 5 we see that our strategy appears to be the one that achieves the best performance. This shows that our feature fusion strategy is effective and also very efficient and cheap in terms of number of parameters.

Table 5: Performance comparison between the model with our mixing strategy and a simple concatenation and convolution on both datasets LEVIR-CD and WHU-CD.

LEVIR-CD						
Model type	Pr	Rc	F1	IoU	OA	Param. tot.
Subtraction	92.36	89.11	90.70	82.99	99.06	284063
Depth. Sep.	92.81	89.01	90.87	83.27	99.08	291513
Concat + conv	92.90	88.23	90.81	83.18	99.08	340568
Mix + grouped	92.68	89.47	91.05	83.57	99.10	285128
WHU-CD						
Model type	Pr	Rc	F1	IoU	OA	Param. tot.
Subtraction	90.32	87.95	89.12	80.38	99.14	284063
Depth. Sep.	91.33	89.05	90.18	82.11	99.23	291513
Concat + conv	91.31	90.66	90.98	83.46	99.28	340568
Mix + grouped	92.22	90.74	91.48	84.30	99.32	285128

In Figure 4 a visual/qualitative comparison between the masks created by our model and those created by BIT [15] on the LEVIR-CD test dataset is reported. Generally speaking, both models perform well and we end up our analysis by conjecturing that the performance difference reported in Table 1 and Table 2 are more related to missing or hallucinated change regions, than region quality issues. Nevertheless, we can find some examples were there are significant differences between the ground truth mask (GT) and that created by the two models. In Figure 4 it is interesting to note that there are examples where both models fail similarly in the same region, despite the two models are based on very different approaches (local versus global).

5 Conclusions and future works

Guided by industrial needs, we proposed a tiny convolutional change-detection Siamese U-Net like model.

Our model exploits low-level features by comparing and classifying them to obtain a binary map of detected changes. We propose a mixing block, introducing the ability to compare/compose features on both spatial and temporal domains.

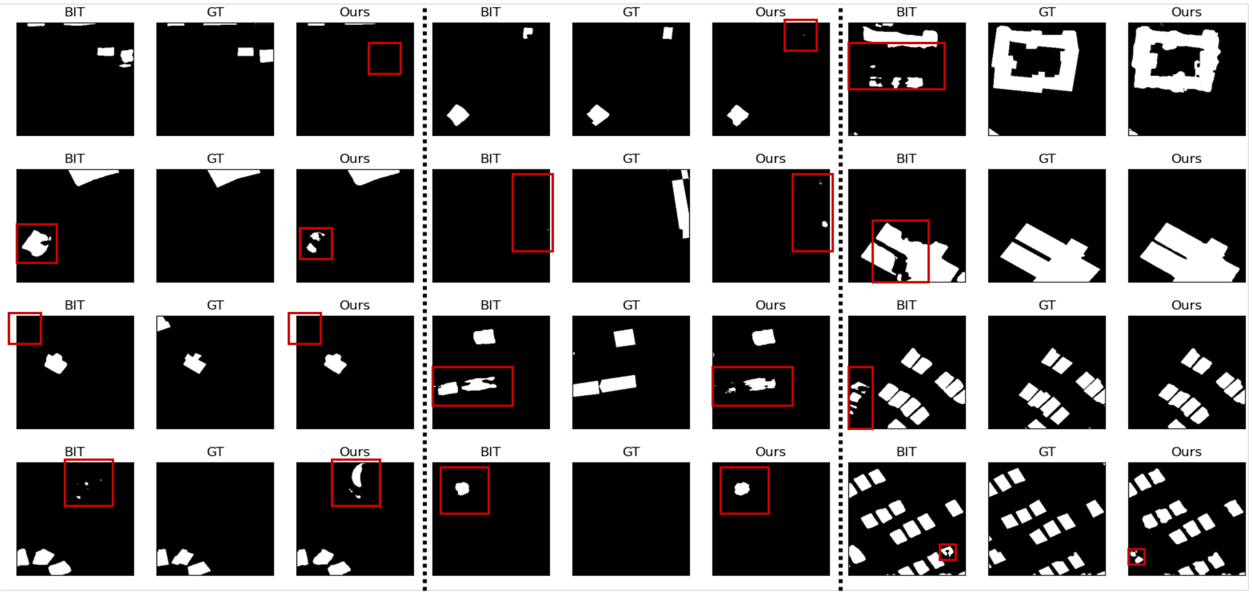

Figure 4: Visual comparison between our model and BIT. We highlighted with red bounding boxes those regions containing significant differences between the ground-truth and the generated masks.

The proposed PW-MLP block, shows a great ability in extracting features useful to classify occurred changes on a per-pixel basis. The composition of these proposed blocks, here referred as MAMB, show the ability to estimate masks useful to enrich features used in the U-Net decoder part. We have shown that an effective way to generate the output mask is to process low-level backbone features in a PW-MLP block, effectively facing the change-detection task as a per-pixel classification problem.

We tested our model on public change-detection datasets containing aerial images acquired in two different times. Furthermore, we compared the achieved results with state-of-the-art models proposed in the change-detection literature. Our tests demonstrated that our model performs comparably or better than the current state-of-the-art models, remaining at the same time the smaller and faster one.

The ideas employed in this work can be also applied to other fields. For this reason, we will investigate the application of MAMB and PW-MLP blocks to tasks such as anomaly-detection, surveillance and semantic segmentation.

Acknowledgments

The authors want to thank the whole Argo Vision team, professor Stefano Gualandi, Gabriele Loli and Gennaro Auricchio for the useful discussion and comments. We also want to thank all those who have provided their codes in an accessible and reproducible way. The scholarship of Andrea Codegoni is founded by SeaVision s.r.l..

References

[1] A. Singh, “Review article digital change detection techniques using remotely-sensed data,” *International journal of remote sensing*, vol. 10, no. 6, pp. 989–1003, 1989.

[2] H. Chen and Z. Shi, “A spatial-temporal attention-based method and a new dataset for remote sensing image change detection,” *Remote Sensing*, vol. 12, no. 10, p. 1662, 2020.

[3] P. P. De Bem, O. A. de Carvalho Junior, R. Fontes Guimarães, and R. A. Trancoso Gomes, “Change detection of deforestation in the brazilian amazon using landsat data and convolutional neural networks,” *Remote Sensing*, vol. 12, no. 6, p. 901, 2020.

[4] A. Viña, F. R. Echavarria, and D. C. Rundquist, “Satellite change detection analysis of deforestation rates and patterns along the colombia–ecuador border,” *AMBIO: A Journal of the Human Environment*, vol. 33, no. 3, pp. 118–125, 2004.

[5] J. Z. Xu, W. Lu, Z. Li, P. Khaitan, and V. Zaytseva, “Building damage detection in satellite imagery using convolutional neural networks,” *arXiv preprint arXiv:1910.06444*, 2019.

[6] C. Zhang, P. Yue, D. Tapete, L. Jiang, B. Shangguan, L. Huang, and G. Liu, “A deeply supervised image fusion network for change detection in high resolution bi-temporal remote sensing images,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 166, pp. 183–200, 2020.

[7] S. Ji, S. Wei, and M. Lu, “Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 1, pp. 574–586, 2018.

[8] L. Khelifi and M. Mignotte, “Deep learning for change detection in remote sensing images: Comprehensive review and meta-analysis,” *IEEE Access*, vol. 8, pp. 126 385–126 400, 2020.

[9] R. C. Daudt, B. Le Saux, and A. Boulch, “Fully convolutional siamese networks for change detection,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 4063–4067.

[10] M. Zhang, G. Xu, K. Chen, M. Yan, and X. Sun, “Triplet-based semantic relation learning for aerial remote sensing image change detection,” *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 2, pp. 266–270, 2018.

[11] Y. Liu, C. Pang, Z. Zhan, X. Zhang, and X. Yang, “Building change detection for remote sensing images using a dual-task constrained deep siamese convolutional network model,” *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 5, pp. 811–815, 2020.

[12] D. Peng, Y. Zhang, and H. Guan, “End-to-end change detection for high resolution satellite images using improved unet++,” *Remote Sensing*, vol. 11, no. 11, p. 1382, 2019.

[13] H. Jiang, X. Hu, K. Li, J. Zhang, J. Gong, and M. Zhang, “Pga-siamnet: Pyramid feature-based attention-guided siamese network for remote sensing orthoimagery building change detection,” *Remote Sensing*, vol. 12, no. 3, p. 484, 2020.

[14] J. Chen, Z. Yuan, J. Peng, L. Chen, H. Huang, J. Zhu, Y. Liu, and H. Li, “Dasnet: Dual attentive fully convolutional siamese networks for change detection in high-resolution satellite images,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 1194–1206, 2020.

[15] H. Chen, Z. Qi, and Z. Shi, “Remote sensing image change detection with transformers,” *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[16] W. G. C. Bandara and V. M. Patel, “A transformer-based siamese network for change detection,” *arXiv preprint arXiv:2201.01293*, 2022.

[17] S. Chopra, R. Hadsell, and Y. LeCun, “Learning a similarity metric discriminatively, with application to face verification,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 1. IEEE, 2005, pp. 539–546.

[18] S. Zagoruyko and N. Komodakis, “Learning to compare image patches via convolutional neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4353–4361.

[19] S. Stent, R. Gherardi, B. Stenger, and R. Cipolla, “Detecting change for multi-view, long-term surface inspection.” in *BMVC*, 2015, pp. 127–1.

[20] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[21] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[22] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, “Fully-convolutional siamese networks for object tracking,” in *European conference on computer vision*. Springer, 2016, pp. 850–865.

[23] A. Ioannidou, E. Chatzilari, S. Nikolopoulos, and I. Kompatsiaris, “Deep learning advances in computer vision with 3d data: A survey,” *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–38, 2017.

[24] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, “Signature verification using a" siamese" time delay neural network,” *Advances in neural information processing systems*, vol. 6, 1993.

[25] M. Lebedev, Y. V. Vizilter, O. Vygolov, V. Knyaz, and A. Y. Rubis, “Change detection in remote sensing images using conditional adversarial networks.” *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 42, no. 2, 2018.

[26] W. Zhao, X. Chen, X. Ge, and J. Chen, “Using adversarial network for multiple change detection in bitemporal remote sensing imagery,” *IEEE Geoscience and Remote Sensing Letters*, 2020.

[27] X. Peng, R. Zhong, Z. Li, and Q. Li, “Optical remote sensing image change detection based on attention mechanism and image difference,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 9, pp. 7296–7307, 2020.

[28] T. Bao, C. Fu, T. Fang, and H. Huo, “Ppcnet: A combined patch-level and pixel-level end-to-end deep network for high-resolution remote sensing image change detection,” *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 10, pp. 1797–1801, 2020.

[29] B. Hou, Q. Liu, H. Wang, and Y. Wang, “From w-net to cdgan: Bitemporal change detection via deep learning techniques,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 3, pp. 1790–1802, 2019.

[30] Y. Zhan, K. Fu, M. Yan, X. Sun, H. Wang, and X. Qiu, “Change detection based on deep siamese convolutional network for optical aerial images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 10, pp. 1845–1849, 2017.

[31] B. Fang, L. Pan, and R. Kou, “Dual learning-based siamese framework for change detection using bi-temporal vhr optical remote sensing images,” *Remote Sensing*, vol. 11, no. 11, p. 1292, 2019.

[32] H. Chen, W. Li, and Z. Shi, “Adversarial instance augmentation for building change detection in remote sensing images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2021.

[33] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[34] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[35] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[36] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[37] B. Wu, C. Xu, X. Dai, A. Wan, P. Zhang, Z. Yan, M. Tomizuka, J. Gonzalez, K. Keutzer, and P. Vajda, “Visual transformers: Token-based image representation and processing for computer vision,” *arXiv preprint arXiv:2006.03677*, 2020.

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[39] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[40] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[41] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Instance normalization: The missing ingredient for fast stylization,” *arXiv preprint arXiv:1607.08022*, 2016.

[42] S. Hochreiter, “Gradient flow in recurrent nets: the difficulty of learning long-term dependencies,” *A field guide to dynamical recurrent neural networks*, 2001.

[43] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[44] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[45] N. Liu, J. Han, and M.-H. Yang, “Picanet: Learning pixel-wise contextual attention for saliency detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3089–3098.

[46] M. Lin, Q. Chen, and S. Yan, “Network in network,” *arXiv preprint arXiv:1312.4400*, 2013.

[47] L. Sifre and S. Mallat, “Rigid-motion scattering for texture classification,” *arXiv preprint arXiv:1403.1687*, 2014.

[48] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[49] W. G. C. Bandara and V. M. Patel, “Revisiting consistency regularization for semi-supervised change detection in remote sensing images,” *arXiv preprint arXiv:2204.08454*, 2022.

[50] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” *Advances in neural information processing systems*, vol. 32, 2019.

[51] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” *arXiv preprint arXiv:1711.05101*, 2017.

[52] Microsoft, “Neural Network Intelligence,” 2021. [Online]. Available: https://github.com/microsoft/nni

[53] I. Loshchilov and F. Hutter, “Sgdr: Stochastic gradient descent with warm restarts,” *arXiv preprint arXiv:1608.03983*, 2016.

[54] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, “Albumentations: fast and flexible image augmentations,” *Information*, vol. 11, no. 2, p. 125, 2020.

[55] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.