TINYCD: a (not so) deep learning model for change detection

Andrea Codegoni1 · Gabriele Lombardi2 · Alessandro Ferrari2

Received: 16 August 2022 / Accepted: 24 November 2022 / Published online: 18 December 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract
In this paper, we present a lightweight and effective change detection model, called TinyCD. This model has been designed to be faster and smaller than current state-of-the-art change detection models due to industrial needs. Despite being from 13 to 140 times smaller than the compared change detection models, and exposing at least a third of the computational complexity, our model outperforms the current state-of-the-art models by at least 1% on both F1-score and IoU on the LEVIR-CD dataset, and more than 8% on the WHU-CD dataset. To reach these results, TinyCD uses a Siamese U-Net architecture exploiting low-level features in a globally temporal and locally spatial way. In addition, it adopts a new strategy to mix features in the space-time domain both to merge the embeddings obtained from the Siamese backbones, and, coupled with an MLP block, it forms a novel space-semantic attention mechanism, the Mix and Attention Mask Block (MAMB).

Keywords Change detection (CD) · Remote sensing (RS) · Convolutional neural network (CNN) · Siamese neural networks (SNN)

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 identify changes occurred in a scene between two different times. To this aim, a CD model compares two co-registered images $I_1$ and $I_2$ acquired at times $t_1$ and $t_2$ [1–3]. Once the relevant changes have been identified, such as urban expansion, deforestation, or post-disaster damage assessment [4–10], the challenge is to let the CD model ignore other irrelevant changes. Examples of irrelevant changes are, but not limited to, lighting conditions, shadows, and seasonal variations.

Thanks to the increasing number of available high resolution aerial images datasets, such as [4, 8, 9], data-driven methods like deep Convolutional Neural Networks (CNN) found successful applicability [11]. The well-known ability of deep CNNs to extract complex and relevant features from images is the key factor for their early promising results [12]. In the CD scenario, complex features are important, but are not sufficient to accomplish the task. To detect the occurred changes, it is in fact crucial to model the spatiotemporal dependencies between the two images. Unfortunately, plain CNNs have a limited receptive field due to the usage of fixed kernels in convolutions. To overcome this issue, recent works focused their attention to enlarging the receptive field by employing different kernel types [13], or by adding attention mechanisms [4, 8, 14, 15, 15–17]. 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 [4, 17] shows promising results relating images in the spatiotemporal domain. More recently, Transformers have been introduced in CD because of their receptive fields spatially covering the whole image [18, 19]. Notice that, by applying multi-headed attention layers in the decoder part of the network, the receptive field covers the temporal domain too.
Unfortunately, the resulting models are computationally very inefficient. The CD field finds applicability also outside the remote sensing world. As an example, in [10, 20] two models are discussed in order to be used on drones or other autonomous vehicles to implement smart city monitoring functions. In our case, the change detection model has been developed for an industrial application. In our application field, the need for real-time performances adds a model complexity constraint. Unfortunately, the majority of State-Of-The-Art (SOTA) models are millions-parameters-sized, so that their applicability is not possible. Another issue with those big models is related to the training time clearly affected by the size of the model. With large models, the Hyper-Parameters-Optimization (HPO) task requires resources that are usually not available to medium-small companies. Moreover, big networks require dedicated hardware also at inference time. This is in contrast with production requirements and project budgets. The search for models having both small size and performances comparable to the current SOTA can be considered an open problem.

A possible strategy to cope with both model size and complexity that, to the best of our knowledge, has not been studied in the literature, is to use low-level features to compare the two images under examination. Another underestimated aspect, in our opinion, is that a Siamese type backbone produces two tensors containing channels underestimated aspect, in our opinion, is that a Siamese compare the two images under examination. Another studied in the literature, is to use low-level features to complex the resulting models are computationally very inefficient. The CD field finds applicability also outside the remote sensing world. As an example, in [10, 20] two models are discussed in order to be used on drones or other autonomous vehicles to implement smart city monitoring functions. In our case, the change detection model has been developed for an industrial application. In our application field, the need for real-time performances adds a model complexity constraint. Unfortunately, the majority of State-Of-The-Art (SOTA) models are millions-parameters-sized, so that their applicability is not possible. Another issue with those big models is related to the training time clearly affected by the size of the model. With large models, the Hyper-Parameters-Optimization (HPO) task requires resources that are usually not available to medium-small companies. Moreover, big networks require dedicated hardware also at inference time. This is in contrast with production requirements and project budgets. The search for models having both small size and performances comparable to the current SOTA can be considered an open problem.

A possible strategy to cope with both model size and complexity that, to the best of our knowledge, has not been studied in the literature, is to use low-level features to compare the two images under examination. Another underestimated aspect, in our opinion, is that a Siamese type backbone produces two tensors containing channels arranged semantically in the same order. This observation could be used to design strategies for merging features more efficiently.

The main purpose of our work is to investigate the aforementioned issues developing a neural network that requires lower computational complexity with respect to the SOTA CD models reaching at the same time comparable performances.

The major contributions of our work are the following:

- We explore the effectiveness of using low-level features in the problem of comparing images. The results validate our intuition that in this context the low-level features are sufficiently expressive. Moreover, this allowed us to significantly limit the number of model parameters.
- We introduce a novel strategy to mix the features between the two images. This strategy allows the computation of a spatiotemporal correlation between the input images keeping a low computational complexity.
- A fast attention mechanism is introduced with a block called MAMB. It uses features localized in space to compute attention masks needed in the up-sampling phase to refine the low-resolution results.
- We propose to use a pixel-wise classifier to generate the final mask. In our tests, this proved to be very effective.

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 relatively small backbone can be adopted. Being the backbone the most time-consuming and parameter-demanding component in the architecture, maintaining it as small as possible allows us to achieve our goal. In particular, this allows us to maintain the total number of parameters below 300,000.

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 the one faced in this paper.

The paper is organized as follows. In Sect. 2 we present some related works. In Sect. 3 we describe our proposed model. In Sect. 4 we report both the results of our model on two publicly available datasets and the ablation study. In Sect. 5 a final discussion of our model is presented and then in Sect. 6 we examine limitation of our work. Finally, in Sect. 7 we draw conclusions, and we highlight some future research directions.

2 Related works

2.1 Early deep neural network works on CD

Deep learning models, and in particular CNNs, have been applied with great success in image comparison tasks [21–23], in pixel-level image classification [24–26], and they represent the SOTA in many other Computer Vision fields [27].

Models in the context of the CD must manage two inputs: one image $I_1$ acquired at time $t_1$, and another one $I_2$ 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. One of the first works using deep learning techniques to the field of CD is [28]. This work highlights how deep neural networks, in particular Deep Belief Networks obtained by stratifying Restricted Boltzmann machines, are a very valid tool to compare and highlight the changes between the two images

---

1 Notice that the backbone is evaluated twice in Siamese architectures.
under examination. To the best of our knowledge, the first work that applies CNNs to the CD problem is [12]. In this work the authors propose two different approaches. In the first case they use a U-Net [25] type network with the Early Fusion Strategy (FC-EF), i.e., they concatenate the images I_1 and I_2, and then they feed the U-Net with the resulting tensor. In the second case they investigate the Feature Fusion Strategy. To this aim, they employ a Siamese U-Net type network [22, 26, 29] 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 [5, 15, 30, 31], and the Feature Fusion Strategy [4, 8, 13, 14, 16, 17, 32–37].

To take full advantage of the large amount of spatial information, deeper CNNs such as ResNet [38] or VGG16 [39] have been used [4, 8, 13, 17] in order to extract spatial information and group them in a hierarchical way. Unfortunately, standard convolution has a fixed receptive field that limits the capacity of modelling the context of the image. To face this issue, atrous convolutions [40] have been experimented [13]; they are able to enlarge the receptive field of convolutional kernels without increasing the number of parameters.

2.2 Attention-based CNN

To definitively overcome the problem of fixed receptive field, attention mechanisms, in the forms of spatial attention [8, 14, 15], channel-wise attention [8, 14–16], and also self-attention [4, 17], have been introduced. In [8], 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 [14] a dual attention module has been introduced. The co-attention module introduced in [16] tries to leverage correlation between features extracted from both images. Also in [16] 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 [4], the non-local self-attention introduced in [41], have been applied to CD. This mechanism consists in stacking the features extracted from a Siamese backbone to then apply both a basic spatial attention mechanism, and a pyramidal attention mechanism. Since these two attention blocks are applied to stacked features obtained from I_1 and I_2, these are correlated in a non-local spatiotemporal way. Another interesting approach is the one presented in [42]. In this paper, the authors decided to combine CNNs with Object Image Analysis (OBIA) to mitigate the limited receptive field problem. In a first preprocessing phase, they segment the image and extract the patches containing objects to be compared. Subsequently, the extracted patches are compared using a CNN which then works on small patches containing more specific and detailed information.

In [20], the authors propose a temporal attention mechanism. They exploit the features extracted from I_2 to generate a query matrix which is then compared with the features extracted from I_1. This mechanism is made dynamic by reducing the receptive field as tensors’ spatial dimension diminishes. Finally, the authors also use attention mechanisms capable of emphasizing some horizontal and vertical dependencies of recurring objects in their scenario.

2.3 Transformers in CD

Finally, the global attention mechanisms introduced with Transformers [43, 44] have also been applied to the CD problem. In [18] 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 [19] 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.

2.4 Relations between our work and existing models

Our work is inspired by [18]. As reported in Sect. 2.3, in that work the authors introduce the concept of semantic tokens, which are basically single pixels of the tensor obtained by the backbone. Then, they use a transformer in order to process these tokens and extract global spatiotemporal information. In agreement with [18], we believe that the information contained in the pixel/semantic token is crucial to obtain a good result. However, we prefer to apply channel-wise local feature comparison, limiting
the semantic complexity and aggregation of adopted features to the first few backbone layers; while in [18] the comparison is global, being it obtained by means of transformers. Moreover, we adopt multilayer perceptrons (MLPs) to compute both the spatial attention maps and the final mask, actually facing the problem as a pixel-wise classification one. Recently, MLP blocks have received great attention in computer vision [45–51]. These architectures divide the images into patches and then process the patches with MLP blocks. Different structures of MLP blocks have been proposed to incorporate as much spatial information as possible. For example, in [46, 51] a spatial shift operator is applied in order to obtain information from different axial directions. The CycleMLP block proposed in [45] follows a similar idea but instead of applying the spatial shift operator to the features’ tensor, it composes several MLP steps capable of mimicking the shift. A more refined version of these concepts is proposed in [47] where the authors employ a block which dynamically learns the spatial offset used in CycleMLP. In our model, the MLP blocks work exclusively along the channel dimension to both compute the spatiotemporal attention maps, and produce the final pixel-wise classification (Fig. 1).

3 Proposed model

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

- Siamese encoders constituted by a pre-trained backbone (see Sect. 3.2).
- Mix and Attention Mask Block (MAMB) and bottleneck mixing block to compose backbone results (see Sect. 3.3).
- Up-sample decoder to refine low resolution results incorporating higher resolution data from the skip connections (see Sect. 3.4).
- Pixel-level classifier (see Sect. 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. We omit the batch dimension for the ease of notation. We denote with Conv the convolution operator, with PReLU the Parametric Rectified Linear Unit [52], with IN2d the Instance Normalization [53], and Sigmoid.

Fig. 1 Siamese U-Net architecture including MAMB

[Diagram of Siamese U-Net architecture including MAMB]
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 Multilayer 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 [54, 55]. To overcome this problem, several tricks have been introduced such as the residual connections of ResNet [38], or the skip connections of the U-Net [25]. 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 [4, 8, 13, 17, 18]. 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, and edges, can be considered general-purpose being common to all images.

In our intuition, in the faced task the comparison between two images $I_1$ and $I_2$ 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 [56] pre-trained on the ImageNet dataset [57]. We allowed the training phase to tune also the totality of the backbone parameters. Guided by experiments on our industrial dataset, the EfficientNet backbone family have been selected 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.

For completeness, in Appendix 1 we compare the performance of other backbones in order to show the generality of our approach.

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 [58]. 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 Sect. 3.3.1), and the Pixel-level mask generator (see Sect. 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 blocks $(X_k, Y_k)$. 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:

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

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:

\[
Z'_k = \text{Mix} (X_k, Y_k) := \text{PReLU} \left[ \text{IN2d} \left[ \text{Conv} (Z_k, \; \text{chin} = 2C_k, \; \text{chout} = C_k, \; \text{groups} = C_k) \right] \right].
\]

(2)

An illustration of our concatenation strategy, and the following grouped convolution, is reported in Fig. 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 spatiotemporal attention.

To this aim, the PW-MLP is designed to produce a mask tensor \( M_k \in \mathbb{R}^{H_k \times W_k} \).

3.3.3 PW-MLP

To implement a pixel-wise Multilayer Perceptron, that is an MLP working on all the channels of one single pixel at a time, we use \( 1 \times 1 \) convolutions. The MLP is composed by \( N \) blocks each containing one \( 1 \times 1 \) convolution and one activation function. As activation, we used the \( \text{PReLU} \), 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 \( 1 \times 1 \) convolutions to implement an MLP is not a new idea. In [59] this strategy has been used to substitute layers such as convolutions with small, trainable, networks. As pointed out in [59], 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 Sect. 3.3.1 to compute the bottleneck of the U-Net like network. More precisely, we compute: \( U_e = \text{Mix} (X_e, Y_e) \).

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

3.4 Up-sampling 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 \( 1, H_k, W_k \). 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

\[
U_k := \text{PReLU} \left[ \text{IN2d} \left[ \text{Conv} (U'_k \odot M_k) \right] \right],
\]

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 Sect. 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 [60, 61].

Fig. 2 Visual representation of our mixing strategy and the full MAMB block. In the inner dashed block we highlight the concatenation strategy, (1), the grouped convolution, (2). These two blocks when coupled with the PW-MLP form the MAMB block.
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 changed pixels. With respect to what reported in Sect. 3.3.3, in this case we used as the last activation layer a 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 nonlinear classifier which separates pixels in normal or changed class.

4 Experiment settings and results

In this section we presents the settings used in our experiments, the achieved results, and the performed ablation study.

4.1 Datasets

As already stated in Sect. 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 building images datasets: LEVIR-CD [4] and WHU-CD [9]. 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 patterns, 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’ original indications. This step produced 7120, 1024, and 2048 pair instances for the train, validation, and test datasets, respectively.

WHU-CD contains just one pair of images having resolution 32507 × 15354 as a crop of a wider geographic area. Following [62], 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 Loss function and evaluation metrics

As stated in Sect. 3.5, we cast the CD problem in a pixel-wise binary classification setting. In fact, the role of the final MLP block is to output the per-pixel change probability.

Since the reference mask is a binary mask (0 for unchanged pixels, 1 for changed pixels), and since we are comparing probabilities, one loss function that can be used is the Binary Cross Entropy (BCE). It is defined as:

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

where we denoted with \( G \) the ground-truth mask, with \( P \) the model prediction, and with \( H \) and \( W \) the set of indices relative to height and width.

Notice that the BCE loss function is widely used in other SOTA models such as [18, 19]. In contrast, other researchers implemented more sophisticated loss functions like the one presented in [4]. We decided to use the simpler BCE in order to attribute the improvement in performances to the model and not to an ad hoc built-in loss function. For completeness, in Appendix 2 we report other experiments conducted using other widely adopted loss functions.

To evaluate the performances achieved by our model, we calculated the Precision (PR), Recall (RC), \( F1 \)-score (F1), Intersection over Union (IoU) and Overall Accuracy (OA) with respect to the change class, as defined below:

\[
Pr = \frac{TP}{TP + FP}, \\
Rc = \frac{TP}{TP + FN}, \\
F1 = \frac{Pr + Rc}{1 + \frac{1}{Pr} + \frac{1}{Rc}}, \\
IoU = \frac{FN + TP}{FN + FP + TP + TN}, \\
OA = \frac{FN + TP}{FN + FP + TP + TN},
\]

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.

4.3 Implementation details

We implemented our model using PyTorch [63] and we trained it on an NVIDIA GeForce RTX 2060 6GB GPU. As described in Sect. 3.2, we selected the first four blocks of the EfficientNet version b4 backbone pre-trained on the...
ImageNet dataset. All other weights of the model have been initialized randomly.\textsuperscript{4}

As optimizer, we adopted AdamW \cite{31}. To optimize its hyperparameters, i.e., learning rate, weight decay and \textit{amsgrad} variant, and also to verify the robustness of our model with respect to the choice of these parameters, we firstly run a Hyper-Parameters Optimization (HPO) task for each dataset using the package Neural Network Intelligence (NNI) \cite{65}. After this, we fixed the learning rate equal to 3 \times 10^{-3}, and the weight decay equal to 9 \times 10^{-3}, for the LEVIR-CD dataset. Moreover, we fixed the learning rate equal to 2 \times 10^{-3}, and the weight decay equal to 8 \times 10^{-3}, for the WHU-CD dataset. For both datasets, \textit{amsgrad} have been set to \textit{False}. An example of the HPO procedure is reported in Appendix 2. Due to computational resource limitations, no other hyperparameters have been tuned. We have not experimented network architecture search techniques (NAS).

To dynamically adjust the learning rate during the training, we adopted the cosine annealing strategy as described in \cite{66}, but avoiding the warm restart.

Since aerial images are spatially registered, we applied the geometric data augmentation operators simultaneously to the reference/comparison images and their associated ground-truth mask. Also, non-geometric augmentations are applied independently on the reference and the comparison images.

The applied geometric augmentations are Random Flip on both X and Y axes, and Random Rotation with free degree. Moreover, the applied non-geometric augmentations are Gaussian Blur and Random Brightness/Contrast change. To achieve all the adopted augmentations, we used the Albumentations library \cite{67}.

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

### 4.4 Comparison with SOTA models

To demonstrate the effectiveness of our approach, we compared our results with those reported in \cite{18,19}. As baseline, we used the three models presented in \cite{12}. Moreover, to compare our model with other works adopting both spatial and channel attention mechanisms, we dealt with \cite{4,8,14,36}. Finally, given the success achieved by Transformers applied to the computer vision field, we also compared our results with those obtained in \cite{18,19}.

\textsuperscript{4} To make our results reproducible, we fixed the random seed at the beginning of each experiment.

### Table 1 Performance metrics on the LEVIR-CD dataset.

| Model          | Pr  | Rc  | F1  | IoU  | OA  |
|----------------|-----|-----|-----|------|-----|
| FC-EF \cite{12}| 86.91 | 80.17 | 83.40 | 71.53 | 98.39 |
| FC-Siam-diff \cite{12}| 89.53 | 83.31 | 86.31 | 75.92 | 98.67 |
| FC-Siam-conc \cite{12}| 91.99 | 76.77 | 83.69 | 71.96 | 98.49 |
| DTCDSCN \cite{14}| 88.53 | 86.83 | 87.67 | 78.05 | 98.77 |
| STANet \cite{4}| 83.81 | 91.00 | 87.26 | 77.40 | 98.66 |
| IFNet \cite{8}| 94.02 | 82.93 | 88.13 | 78.77 | 98.87 |
| SNUNet \cite{36}| 89.18 | 87.17 | 88.16 | 78.83 | 98.82 |
| BIT \cite{18}| 89.24 | 89.37 | 89.31 | 80.68 | 98.92 |
| Changeformer \cite{19}| 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.

| Model          | Pr  | Rc  | F1  | IoU  | OA  |
|----------------|-----|-----|-----|------|-----|
| FC-EF \cite{12}| 71.63 | 67.25 | 69.37 | 53.11 | 97.61 |
| FC-Siam-diff \cite{12}| 47.33 | 77.66 | 58.81 | 41.66 | 95.63 |
| FC-Siam-conc \cite{12}| 60.88 | 73.58 | 66.63 | 49.95 | 97.04 |
| DTCDSCN \cite{14}| 63.92 | 82.30 | 71.95 | 56.19 | 97.42 |
| STANet \cite{4}| 79.37 | 85.50 | 82.32 | 69.95 | 98.52 |
| IFNet \cite{8}| 96.91 | 73.19 | 83.40 | 71.52 | 98.83 |
| SNUNet \cite{36}| 85.60 | 81.49 | 83.50 | 71.67 | 98.71 |
| BIT \cite{18}| 86.64 | 81.48 | 83.98 | 72.39 | 98.75 |
| Ours           | 91.72 | 91.76 | 91.74 | 84.74 | 99.34 |

The results reported in Tables 1 and 2 show the superior performance of our model on the LEVIR-CD and WHU-CD building change detection datasets.

The baseline models FC-Siam-diff and FC-Siam-conc \cite{12} are 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 \cite{19}, our performance increment on the LEVIR-CD dataset is more limited. However, as we can see from Table 3, our model is 146.50 times smaller.

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.

In Fig. 3 a visual/qualitative comparison between the masks created by our model, and those created by BIT [18] 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 Tables 1 and 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 masks (GT) and those created by the two models. In Fig. 3 it is interesting to note that there are examples where both models fail similarly in the same regions, despite the two models being based on very different approaches (local versus global).

### 4.5 Ablation study

In this section we describe the adopted ablation study steps and the achieved results.

#### 4.5.1 Backbone dimension and final PW-MLP

The first ablation study we conducted concerns the size of the backbone and the use of the final MLP. Regarding the backbone size, we considered both the whole EfficientNet-b4 except the final classifier, and a sliced version of the

| Model          | Param (M) | Param ratio | FLOPs (G) | LEVIR-CD F1 | WHU-CD F1 |
|----------------|-----------|-------------|-----------|-------------|-----------|
| DTCDSCN [14]   | 41.07     | 146.67      | 7.21      | 87.67       | 71.95     |
| STANet [4]     | 16.93     | 60.46       | 6.58      | 87.26       | 82.32     |
| IFNet [8]      | 50.71     | 181.10      | 41.18     | 88.13       | 83.40     |
| SNUNet [36]    | 12.03     | 42.96       | 27.44     | 88.16       | 83.50     |
| BIT [18]       | 3.55      | 12.67       | 4.35      | 89.31       | 83.98     |
| Changeformer [19] | 41.02    | 146.50      | N.D       | 90.40       | N.D       |
| Ours           | 0.28      | 1           | 1.45      | 91.05       | 91.74     |
EfficientNet-b4 network including just the first 3 blocks. Moreover, to assess the effectiveness of the final classification PW-MLP block, we considered both the architecture including it, and the one that produces its output directly from the last up-sampling block by forcing it output just one channel. The results shown in Table 4 confirm our intuition on low-level features. In fact, our solution with the sliced backbone and final PW-MLP, turns out to be the one with the best performances on both datasets. Furthermore, we note that, to get the best performances, the backbone slicing and PW-MLP classifier must be coupled. In fact, on LEVIR-CD the backbone slicing only model shows poor performances, while the use of the PW-MLP classifier helps the full backbone architecture to improve the quality of the segmentations. In contrast, on the WHU-CD the architecture with sliced backbone and the PW-MLP classifier obtains better scores than the one with full backbone but without PW-MLP, remaining the performances of the latter still unsatisfactory and far from those obtained by our model.

4.5.2 Impact of skip connection with MAMB

To quantitatively confirm the usefulness of the skip connections, we trained a model without them and compared the achieved results in Table 5.

As can be seen, all the metrics confirm the beneficial effects of skip connections in the model. In Fig. 4 we reported an example of the intermediate masks that our model creates in the skip connections.

As can be seen, the masks created with the MAMB block at resolution 64 highlight the objects that must be tracked (red pixels). The intermediate masks at resolution 128 act more like an edge detector. Finally, the masks at resolution 256, obtained applying the MAMB block directly to the original images \( I_1 \) and \( I_2 \), distinguish between object classes like buildings and street (dark blue), vegetation (light green), and shadows (red). The ability to highlight shadows is very effective since it helps the model to detect objects and to refine their edges.

4.5.3 Comparison with other simple mixing strategy

In Table 6 we compare our mixing strategy, described in Sect. 3.3.1, with other widely used feature fusion blocks. We tested the following alternatives:

- subtraction, both in the bottleneck and in skip connections;
- concatenation + convolution, both in the bottleneck and in skip connections.

We selected these two alternatives since our mixing strategy can be seen as a generalization of the pixel-wise subtraction. However, our mixing block Sect. 3.3.1 is fully trainable with the spirit of feature re-use [68]. Moreover, concatenation + convolution can be seen as generalization of our mixing block. However, the number of trainable parameters to be tuned for this mixing block is much bigger than ours. More precisely, the number of parameters in our mixing block is \( c(2 \cdot k_h \cdot k_w) \), where \( c \) is the number of channels, \( k_h, k_w \) are the convolutional kernel sizes. By comparison, a convolution working on the concatenated feature tensors contains \( c(2c \cdot k_h \cdot k_w) \) parameters.

In Table 7 we reported the results of a more detailed study on mixing strategies. We alternated the use of subtraction/concatenation + convolution with our respective proposal to mix the features in the bottleneck/skip connections.

The obtained results confirm that our proposal can be considered an effective generalization of the subtraction, with little impact on the size and complexity of the model. On the other hand, the overhead introduced by the concatenation + convolution mixing strategy, seems to produce little differences in terms of performance.

4.5.4 Channel-wise MLP vs CycleMLP

As reported in Sect. 2.4, several MLP blocks have recently been studied with the intent of incorporating both spatial and channel-specific information. As previously described, we used the MLPs only along the channels in the final classifier, and coupled to our mixing strategy in the MAMB blocks to obtain space-time correlation. We then decided to deal with the CycleMLP block proposed in [45]. The results reported in Table 8 suggest the superiority of our proposed use of MLPs compared to that proposed in [45]. A heuristic explanation for these results can be the following: the MLP blocks proposed in [45] have shown to obtain excellent performances when they are used to construct a hierarchical architecture to generate pyramid features. This makes us think that the advantage of CycleMLPs may be more significant when the features are more refined than the low-level features we use.

---

5 We employed the sigmoid activation on this output.

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.

7 The parentheses are highlighting the size of each kernel and the number of kernels.
In CD problems, the model must be able to compare two input images, highlighting the changes that occurred. The model we presented in this work, TinyCD, exploits low-level features by comparing them and classifying pixels to obtain a binary map of detected changes. Despite the lower computational complexity and the reduced number of parameters, the results achieved place TinyCD among the CD SOTA models.

Among others, the adoption of low-level features also allows to reduce the depth of the included backbone part,
thus reducing the overall depth of the model. In this way, the resulting network is wider than deeper, and this allows to reduce the gradient confusion, improving the training efficiency [69]. Moreover, smaller models are easier and faster to train, allowing a more effective and affordable HPO phase. In industrial applications this is a desirable feature, since the model can be retrained according to the necessity of the costumers.

We have shown that an effective way to generate the output mask is to process low-level backbone features with a PW-MLP block, facing the change-detection task as a per-pixel classification problem. Moreover, the results of the ablation study reported in Table 4, show that low-level backbone features, and the final PW-MLP classifier, perform best when coupled.

The mixing strategy that we have introduced, and the MAMBs derived from it, leverage the structure of the Siamese networks. Indeed, the features extracted from the two branch of the network share the semantics associated to the single channels/filters. Our mixing strategy takes this as an advantage, efficiently comparing those corresponding features in a spatiotemporal manner. In the case of MAMBs, the low-level features are exploited to form the attention masks. These masks show how the low level features are able, at different levels, to help the model to pay attention to details of different meanings.

6 Limitations of our work

In all the experiments we performed, we observed our model learning some domain-specific patterns. Despite being this an advantage allowing the model to better deal with the faced task, this is also a limitation because it reduces the model’s ability to adapt to new scenarios by means of fine-tuning. Furthermore, in the two datasets taken into consideration, and also in our industrial case study, the images $I_1$ and $I_2$ are spatially registered. This allowed the successful usage of low-level features without assessing to global feature relationships. In different contexts, where the images undergo large spatial shifts, this local approach can show worse performances with respect to more global approaches like vision transformers [18, 19].
7 Conclusions and future works

Guided by industrial needs, we propose TinyCD, a convolutional change-detection Siamese U-Net-like model. TinyCD exploits low-level features by comparing and classifying them to obtain a binary map of detected changes. We investigate the ability of PW-MLP blocks in extracting features, and propose a new spatiotemporal features fusion strategy. The latter is more general than the subtraction between tensors, and it is also less expensive than convolution + concatenation with comparable performances, if not superior. Combining these two elements, we introduce MAMB, a mixing and attention mask block that we employ to create skip connection masks. This approach has the advantage of reducing the number of parameters and the computational complexity of the model, requirements that are fundamental for the industrial application that motivated our research.

We tested our model on public change-detection datasets containing aerial images acquired at two different times. Furthermore, we compared the achieved results with SOTA models proposed in the change-detection literature.

Table 9 Comparison of different backbones on LEVIR-CD dataset

| Backbone       | Precision | Recall  | F1-score | IoU   | Accuracy | Params    |
|----------------|-----------|---------|----------|-------|----------|-----------|
| mobilenetv2    | 90.95     | 86.43   | 88.63    | 79.59 | 98.87    | 38,798    |
| mobilenetv3large | 90.56    | 85.98   | 88.21    | 78.91 | 98.82    | 32,886    |
| resnet18       | 92.15     | 87.43   | 89.72    | 81.37 | 98.98    | 707,894   |
| efficientnetb0 | 92.18     | 87.96   | 90.02    | 81.85 | 99.00    | 79,480    |
| efficientnetb1 | 92.17     | 88.92   | 90.51    | 82.67 | 99.05    | 122,092   |
| efficientnetb2 | 92.13     | 89.26   | 90.68    | 82.94 | 99.06    | 148,040   |
| efficientnetb3 | 92.40     | 89.54   | 90.95    | 83.40 | 99.09    | 178,716   |
| efficientnetb4 | 92.68     | 89.47   | 91.05    | 83.57 | 99.10    | 285,128   |
| mnasnet13      | 91.95     | 88.17   | 90.02    | 81.86 | 99.00    | 97,262    |
| densenet121    | 92.13     | 87.97   | 90.00    | 81.83 | 99.00    | 1,364,790 |

Fig. 5 Different combination of parameters and their impact on the F1-score on the LEVIR-CD dataset

Fig. 6 Behavior of the final F1-score in the different experiments conducted to tune the hyperparameters. The drop in the F1-score is due to the use of IoU as loss function
Our tests demonstrated that our model performs comparably or better than the current SOTA models, remaining at the same time the smaller and faster one.

Notice that 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.

As a future work, solutions to the limitations presented in Sect. 6 will be investigated. We also left for future works an extensive study on how our low-level local approach can be beneficial for training and performances also in other areas besides CD. Moreover, in order to be able to extend our approach even in those contexts where global features play a fundamental role, we would like to explore multibranch models in which one branch works on local features and the other on global ones.

**Appendix 1 Backbones comparison**

In this Appendix we report the results obtained by varying the backbone adopted in the model. In each backbone we select only few initial blocks in order to work with features that are not very complex and not excessively aggregated from a spatial point of view. We decide to select the all the initial blocks up to the first having spatial resolution 32x32. Due to the different compositions of the considered networks, the final size of the models changes it the range starting from a minimum of 32k parameters up to 1.3 M.

As we can see from Table 9, the results obtained are stable from the performances point of view. The backbones of the EfficientNet family appear to be, in accordance with the experiments carried out on our proprietary dataset, those that achieve the best performances. However, the other backbone types also produce comparable results making our approach:

- robust with respect to the backbone used;
- flexible with respect to the required size and computational complexity.

In this comparison we have not considered Transformers-type backbones such as [70, 71]. The reason for this choice lies in the fact that the philosophy of the Transformers is a global philosophy, as opposed to the blocks we propose which are instead local. As mentioned in Sect. 7, an integration of these two philosophies will be the subject of future works.

**Appendix 2 Hyperparameters tuning**

In this Appendix we report the details about the hyperparameters tuning experiments. One of the advantages of using limited computational complexity models is being able to fine-tune hyperparameters using relatively few computational resources and in a reasonable time from an industrial point of view. In our experiments we tune the learning rate, the weight decay, and the usage of the `amsgrad` strategy. The framework used to run the experiments and optimize the hyperparameters is NNI [65].

Since we execute only 100 epochs per run, we chose a higher learning rate range \(10^{-3}, 4 \times 10^{-3}\), in order to explore whether a higher than standard learning rate leads to faster model convergence. As for the weight decay, we follow a conservative choice by setting the range between \(10^{-2}\) and \(8 \times 10^{-3}\). We also test other simple loss functions for model training such as Mean Square Error (MSE), Intersection over Union (IoU) and a combination of IoU and BCE.

In Fig. 5 we show the various combinations of hyperparameters explored in a batch of 30 experiments, and the relative performances on the LEVIR-CD validation set. Analyzing the results, we note that BCE and MSE, regardless of the other parameters, obtain superior performance compared to the IoU. In addition, the BCE + IoU combination, although better than IoU, also scores lower than the BCE and MSE. Regarding the other hyperparameters, as can be seen in particular from Fig. 6, our model obtains robust performances with respect to all the tested combinations. Finally, we note that in the conducted experiments, BCE has lower variance in terms of F1-score with respect to the choices of the other hyperparameters. This represents another motivation for us to chose BCE as loss function.

**Acknowledgements** 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. We want to thank all the anonymous reviewers for their comments and suggestions that help us to improve our work. The Ph.D. scholarship of Andrea Codegoni is founded by SEA Vision.

**Data availability** The datasets analyzed during the current study, and all the source code to reproduce our results, are available in the GitHub repository https://github.com/AndreaCodegoni/Tiny_model_4_CD.

**Declarations**

**Conflict of interest** The authors have no conflicts of interest to declare.

**References**

1. Singh A (1989) Review article digital change detection techniques using remotely-sensed data. Int J Remote Sens 10(6):989–1003
2. Shafique A, Cao G, Khan Z, Asad M, Aslam M (2022) Deep learning-based change detection in remote sensing images: a review. Remote Sens 14(4):871
41. Wang X, Girshick R, Gupta A, He K (2018) Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7794–7803
42. Liu T, Yang L, Lunga D (2021) Change detection using deep learning approach with object-based image analysis. Remote Sens Environ 256:112308
43. Wu B, Xu C, Dai X, Wan A, Zhang P, Yan Z, Tomizuka M, Gonzalez J, Keutzer K, Vajda P (2020) Visual transformers: token-based image representation and processing for computer vision. arXiv preprint arXiv:2006.03677
44. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. Adv Neural Inform Process Syst 30:6000–6010
45. Chen S, Xie E, Chongjian G, Chen R, Liang D, Luo P (2022) Cyclemlp: a mlp-like architecture for dense prediction. In: International conference on learning representations. Oral Presentation
46. Lian D, Yu Z, Sun X, Gao S (2022) As-mlp: an axial shifted mlp architecture for vision. In: International conference on learning representations. Poster presentation
47. Zhang J, Yang K, Ma C, Reiši S, Peng K, Stiefelhagen R (2022) Bending reality: distortion-aware transformers for adapting to panoramic semantic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 16917–16927
48. Tolstikhin IO, Houbis N, Kolesnikov A, Beyer L, Zhai X, Touvron H, Bojanowski P, Caron M, Cord M, El-Nouby A, Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L et al (2019) Pytorch: an imperative style, high-performance deep learning library. Adv Neural Inform Process Syst 32:8026–8037
49. Loshchilov I, Hutter F (2017) Decoupled weight decay regularization for semi-supervised change detection in remote sensing images. arXiv preprint arXiv:2204.08454
50. Microsoft: neural network intelligence (2021). https://github.com/microsoft/nni
51. Yu T, Li X, Cai Y, Sun M, Li P (2022) S2-mlp: spatial-shift mlp architecture for vision. In: Adv Neural Inform Process Syst 34:9204–9215
52. Yu T, Li X, Cai Y, Sun M, Li P (2022) S2-mlp: spatial-shift mlp architecture for vision. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp. 297–306
53. He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 1026–1034
54. Ulyanov D, Vedaldi A, Lempitsky V (2016) Instance normalization: the missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022
55. Hochreiter S, Bengio Y, Frasconi P, Schmidhuber J (2001) Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In: A field guide to dynamical recurrent neural networks, pp. 237–244
56. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural comput 9(8):1735–1780
57. Tan M, Le Q (2019) Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning, pp. 6105–6114
58. Liu N, Han J, Yang M-H (2018) Picanet: Learning pixel-wise contextual attention for saliency detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3089–3098 (2018)
59. Lin M, Chen Q, Yan S (2013) Network in network. In: International conference on learning representations. arXiv preprint arXiv:1312.4400
60. Sifre L, Mallat S (2014) Rigid-motion scattering for texture classification. Comput Sci 3559:501–515
61. Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1251–1258
62. Bandara WGC, Patel VM (2022) Revisiting consistency regularization for semi-supervised change detection in remote sensing images. arXiv preprint arXiv:2204.08454
63. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L et al (2019) Pytorch: an imperative style, high-performance deep learning library. Adv Neural Inform Process Syst 32:8026–8037
64. Bengio Y, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell 35(8):1798–1828
65. Sankaranaman KA, De S, Xu Z, Huang WR, Goldstein T (2020) The impact of neural network overparameterization on gradient confusion and stochastic gradient descent. In: International conference on machine learning, pp. 8469–8479
66. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 10012–10022
67. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houbis N (2021) An image is worth 16x16 words: transformers for image recognition at scale. In: International conference on learning representations. Oral presentation

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.