AYANet: A Gabor Wavelet-based and CNN-based Double Encoder for Building Change Detection in Remote Sensing

Priscilla Indira Osa, Josiane Zerubia, Zoltan Kato

AYANet: A Gabor Wavelet-based and CNN-based Double Encoder for Building Change Detection in Remote Sensing

Priscilla Indira Osa[1,2⋆][0000−0001−5815−1563], Josiane Zerubia[2][0000−0002−7444−085], and Zoltan Kato[3,4][0000−0002−9328−4254]

[1] University of Genoa, DITEN dept., Italy, priscilla.indira.osa@edu.unige.it
[2] Inria, Université Côte d'Azur, France
[3] University of Szeged, Institute of Informatics, Hungary, kato@inf.u-szeged.hu
[4] J. Selye University, Komarno, Slovakia

Abstract. The main challenge presents in bitemporal building change detection (BCD) in remote sensing (RS) is to detect the relevant changes that are related to the buildings, while ignoring changes induced by other types of land cover as well as varied environmental condition during the sensing process. In this paper, we propose a new BCD model with a double encoder architecture. The Gabor wavelet-based encoder which aims to highlight the characteristic of buildings on RS imagery i.e., the comparatively more regular and repetitive texture than other objects on RS images. This Gabor Encoder is used in addition to the convolutional-neural-network-based encoder that extracts other meaningful and high-level information from the images. Moreover, we also propose Feature Conjunction Module to efficiently combine the extracted features by characterizing possible types of changes. Comparative results with State-of-the-art models on 3 different BCD datasets (LEVIR-CD, S2Looking, and WHU-CD) confirm that the proposed model outperforms current BCD methods in producing a highly accurate change map of buildings. Our code is available on https://github.com/Ayana-Inria/AYANet.

Keywords: Gabor wavelet · Convolutional Neural Network · Building Change Detection · Remote Sensing.

1 Introduction

Change Detection aims to identify changes occurred in a scene between two different times, based on a pair of (geometrically) registered images acquired at pre and post-event. Some examples of the event that can cause the changes include urban expansion, deforestation, or natural disaster. The challenge is to recognize changes attributed to the event, while ignoring any other visual changes that

⋆ The first author performed the work while at Inria, Université Côte d'Azur, France. University of Genoa and Université Côte d'Azur are part of the Ulysseus Alliance (European University). https://ulysseus.eu/.

are unrelated to the event itself, which are generally due to lighting conditions, shadows, seasonal variations, or changes in other environmental conditions. An important special case of change detection is *Building Change Detection* (BCD), where the goal is to highlight changes only in buildings and ignore the irrelevant changes of other objects (*e.g.* vegetation) [22]. In remote sensing imagery, built-in regions typically have a distinctive repetitive visual pattern compared to other natural regions. Thus, such characteristics are important in identifying built-in areas as well as any changes related to buildings (either buildings that are demolished or newly constructed). The typical BCD task creates a change map that highlights appearance and disappearance of buildings, which can be used as the starting point of a broad range of applications, such as urban growth analysis [9], and disaster assessment and recovery [20].

Traditional change detection methods can be divided into pixel-based and object-based methods. While pixel-based methods rely on pixel-wise spectral value changes between bi-temporal images, object-based methods can incorporate both spectral and spatial (*e.g.* shape, texture) contextual information of images. The former approach is challenged by the limited spatial contextual information provided by a small neighborhood of pixels, while the latter one is subject to object segmentation errors and lacks the capability to include both local and global features which are crucial as local features preserve spatial details and global features provide a bigger context information to accurately recognize the semantic information of pixels.

Deep Convolutional Neural Networks (CNN) have demonstrated promising performance in addressing the complexities of the BCD task [1, 4, 13, 18, 26]. CNN is able to extract image features via spatial convolutions and hierarchical feature representations, which successfully combines local features by gradually increasing the effective receptive field of subsequent layers as it goes deeper in the network, creating a pyramid-like stack of features at multiple resolution. Recently, Transformer networks are becoming popular in BCD because of their efficiency in capturing the global context of the features. It can be incorporated in combination with CNN [3, 15], or it can also be used without feature extraction by CNNs [2]. Theoretically, both CNN and Transformer can learn texture features from the training image data [17, 19], assuming sufficiently many training data are available. However this is not the case in remote sensing imagery. While general purpose large datasets exist to train such networks, *e.g.* ImageNet [8] which contains around 14 million images, and JFT-3B [27] with approximately 3 billion images, open BCD datasets generally contain fewer images by several order of magnitude (less than hundreds of thousands). This is a serious constraint when more and more complex models are appearing with several million parameters to learn.

Models based on CNN, Transformer, or both, incorporate typical strategies such as metric-based learning [4], as well as integrating attention mechanisms [1, 4, 10, 18]. Indeed, attention-based approaches put weights on relevant features *e.g.* temporal attention which emphasizes the relation between the features of the bi-temporal images that accentuate the change [4]. However, considering

the typical size of BCD datasets, it is by far not evident that such complex networks can learn features effectively, especially Transformer which may fail to learn some specific features if the training data are not provided sufficiently [19]. On the other hand, other approaches are using fewer parameters to learn by reducing the complexity of the network [3, 7, 10, 15]. While all of the State-of-the-Art methods mentioned above, including attention-based, Transformer-based or CNN-based ones, perform well (see Section 3.3), none of them explicitly perform feature extractions that are characteristic to the particular texture properties of building in spite of its importance in differentiating buildings from other objects in the BCD task.

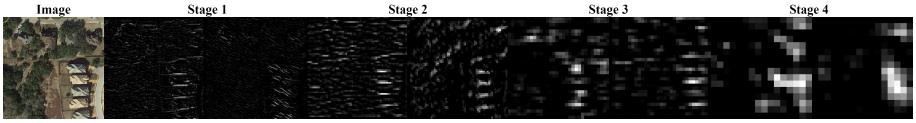

Fig. 1. Some examples of features extracted at different stages of the Gabor Encoder. The deeper the stage goes, the lower the resolution is. Notice that regions with buildings are clearly highlighted at each resolution, while other regions (in spite of being textured but without regularity) are supressed in the feature maps.

To address this issue, we propose AYANet which adopts a double-encoder feature extraction backbone that provides rich *texture* features in a Siamese network to extract multi-scale features from bi-temporal image pairs. At each resolution, feature differences are extracted and forwarded to a final decoder, which identifies building changes and provides the final change map. The main contributions of this paper are:

1. We integrate local feature extraction from a CNN-based encoder which is based on EfficientNet-B7 [23], and explore the advantages of a dedicated multi-scale texture feature encoder based on Gabor wavelets [11], in the form of a so called *double encoder* where CNN-extracted hierarchical features are augmented by features directly representing repetitive visual patterns at different scale and orientation. One can also interpret it as a kind of attention to highly regular textured regions. Fig. 1 illustrates some multi-scale features extracted by the proposed Gabor Encoder. We can observe that the extracted features highlight the textures of buildings that are located on the right side of the image. While a CNN can already extract general texture features from the input images, the intuition we have in mind when designing the Gabor Encoder is to ensure the encoder to extract the textures belong to the buildings by imposing Gabor filters when the network learns to update the convolutional filters which are integrated together as the building block of the encoder (more detail in Section 2.1).

2. Features from corresponding scale in the Siamese network are processed by a Feature Conjunction Module (FCM), which will characterize their dissimilarity for the decoder.

The quantitative and qualitative experimental results on standard datasets demonstrate the superiority of our method.

2 AYANet

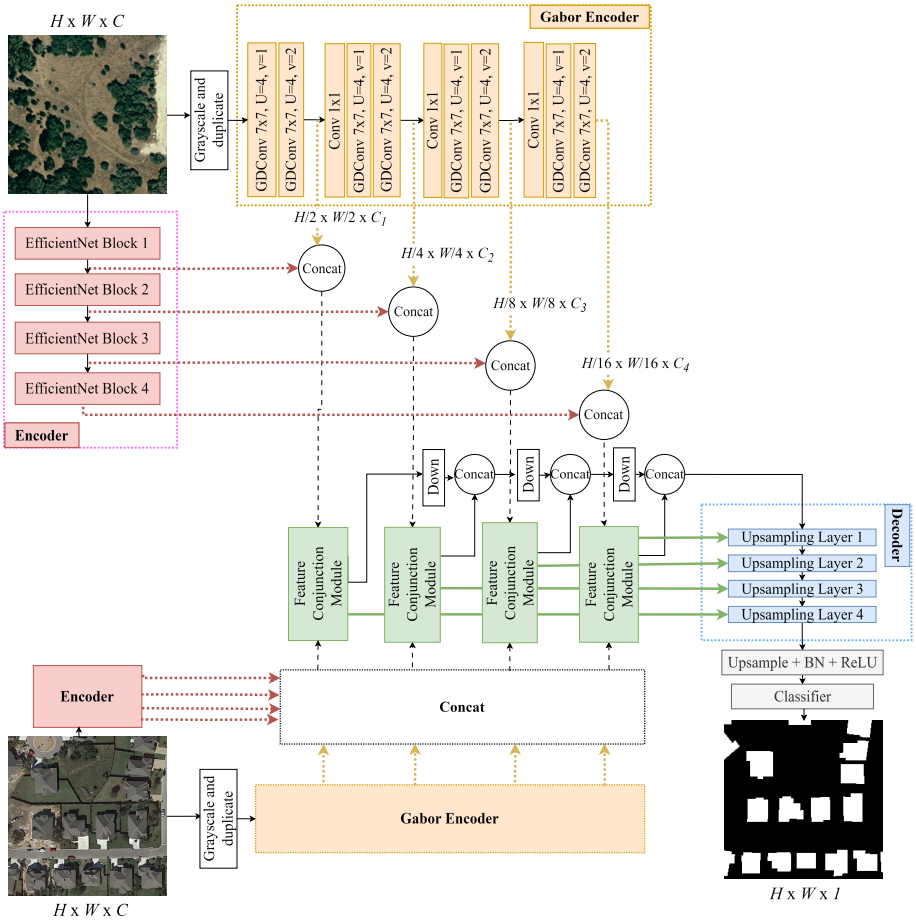

Fig. 2. The architecture of AYANet. The design follows the style of a Siamese network i.e., the same Encoder and Gabor Encoder are used to process the two input images.

The proposed model, shown in Fig. 2, is a Siamese network with three main components:

1. double encoder consisting of our *Gabor Encoder* and the EfficientNet-based general *Encoder*
2. *Feature Conjunction Module*s at 4 resolutions
3. the *Decoder* and *Classifier* which produces the final change map

A pair of pre-change and post-change images with input size $H \times W \times C$ (H, W, C refer to height, width, channels respectively), goes directly to the Encoder. The Encoder produces multi-scale features from each block with a size of $\frac{H}{2^i} \times \frac{W}{2^i} \times C_i$, where $i = \{1, 2, 3, 4\}$, and $C_{i+1} > C_i$. The same pair of input images is converted to grayscale and is duplicated to $H \times W \times C_1$ before being fed to the Gabor Encoder in order to accommodate the depthwise mechanism used in that block, which will be explained in detailed in Subsection 2.1. The Gabor Encoder extracts features at different scales at the same resolution as the features extracted by the Encoder. Features are then concatenated and being passed to the Feature Conjunction Module where pre-change and post-change features are combined such that feature changes are highlighted. These conjugated features are subsequently passed to the Decoder. The Decoder of AYANet utilizes the decoder part of [5], which comprises several upsampling layers. The operation includes a simple bilinear upsample followed by the sum of upsampled features, and the features coming directly from the FCM module at every stage. The binary change map is produced by classifying the features upsampled by transposed convolution layers.

2.1 Double Encoder

Feature extraction by the double encoder comprises two components. One CNN-based Encoder, which consists of the first 4 mobile inverted bottleneck blocks (MBConv) of EfficientNet-B7 [23]. The depthwise separable convolution implementation in the building block of EfficientNet allows deep feature extraction with less computational cost compared to architectures using regular convolution blocks. Moreover, the squeeze and excitation (SE) block [12] in MBConv will act as the channel attention mechanism in the Encoder which models the interdependencies among channels of the features. The other one is our Gabor Encoder which focuses on the repetitive visual patterns of the buildings.

Gabor Encoder. The main element of Gabor Encoder is inspired by Gabor Orientation Filters (GoFs) proposed in [17]. A GoF consists of a group of filters in which each of the filter is a learnable convolutional filter modulated by a Gabor filter [17]. Gabor filters [11] are biologically motivated as mammalians' vision system uses similar multiscale filters to extract texture information from retinal images. Gabor filters are represented by the following equation [16, 17, 25]:

$$G(u, v) = \frac{||\boldsymbol{k_{u,v}}||^2}{\sigma^2} e^{-(||\boldsymbol{k_{u,v}}||^2 ||\boldsymbol{z}||^2 / 2\sigma^2)} [e^{i\boldsymbol{k_{u,v}}\boldsymbol{z}} - e^{-\sigma^2/2}], \tag{1}$$

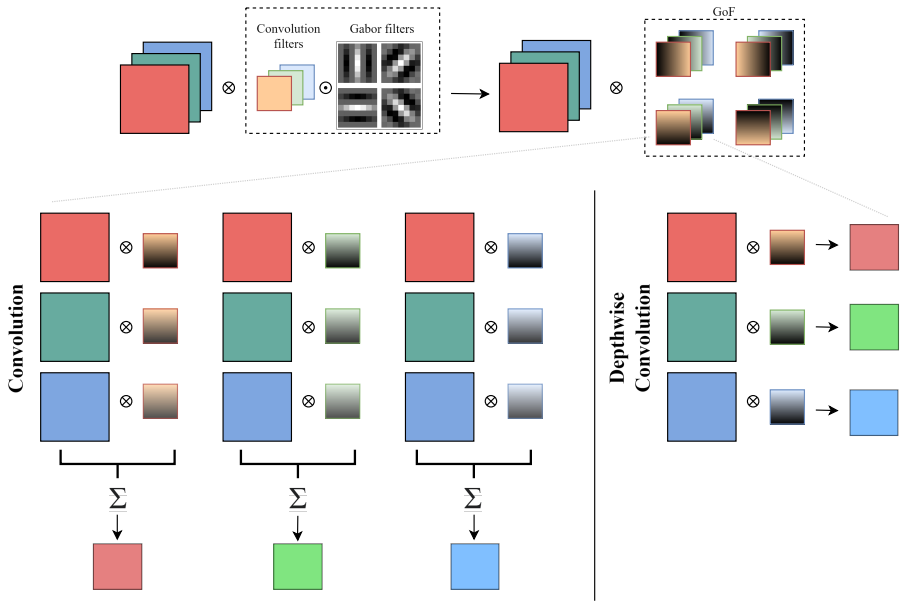

Fig. 3. The upper part of the image shows how GoF is obtained and the lower part indicates the difference on how the filters work between standard convolution (left) and depthwise convolution (right).

where $\boldsymbol{z} = (x, y)$, $\boldsymbol{k_{u,v}} = \begin{pmatrix} k_{jx} \\ k_{jy} \end{pmatrix} = \begin{pmatrix} k_v \cos k_u \\ k_v \sin k_u \end{pmatrix}$, frequency $k_v = (\pi/2)/\sqrt{2}^{(v-1)}$, orientation $k_u = u\frac{\pi}{U}$, and $\sigma = 2\pi$. The scale parameter $v = 1, ..., V$ controls the frequency of the filter in inverse proportion while the parameter $u = 0, ..., U-1$ determines the orientation of the filter.

Each filter in a GoF is a product of element-wise multiplication of a convolutional filter C_i of size $N \times K \times K$ with a Gabor filter $G(u, v)$ with size $K \times K$, orientation u, and scale v

$$C^v_{i,u} = C_i \odot G(u, v), \tag{2}$$

Thus, a GoF [17] comprises a group of filters with a scale v and a set of orientations U

$$\hat{C}^v_i = (C^v_{i,0}, ..., C^v_{i,U-1}), \tag{3}$$

The upper part of Fig. 3 illustrates the process to obtain a GoF. We intuitively interpret the integration of Gabor filters in the convolutional block, in some way, guides the parameter learning in the Gabor Encoder to be imposed by Gabor filters we set in the GoFs because the backpropagation process will take into account the Gabor filters in each block [17]. Additionally, we modified the original GoF by replacing standard convolution to depthwise convolution [6],

which changes the operation (assume stride=1 with padding) from

$$\hat{F}_{k,l,n} = \sum_{i,j,m} K_{i,j,m,n} F_{k+i-1,l+j-1,m}, \tag{4}$$

to

$$\hat{F}_{k,l,m} = \sum_{i,j} K_{i,j,m} F_{k+i-1,l+j-1,m} \tag{5}$$

where K is the filter, F is the input image or feature, and $\hat{F}$ is the output feature. (i, j) denotes the position of the cell indexed based on the kernel size, (k, l) defines the position of the cell indexed based on the output feature size, m increments until the number of input channel, and n is looped until the number of output channel. The lower part of Fig. 3 shows the difference between convolution operation on the left and depthwise convolution on the right. Depthwise convolution applies one kernel to each input channel, which provides the following benefits: 1) less computational cost and 2) filtering the features spatially without the mixing of channel-wise information. In order to implement this, we need to make sure that the number of input channels is the same as the number of filters or the output channels, which is the reason why we need to duplicate the input image to the number of filters of the first block in the Gabor Encoder.

Referring back to Fig. 2, The block of operations between the input image or input features and GoF with the depthwise convolution is called GDConv. Two blocks of GDConv with kernel size of 7×7 are responsible to produce the Gabor Encoder's output features at a particular resolution. These output features from each stage are then to be concatenated with the output features from the EfficientNet encoder at the same resolution. The orientation of GDConv was set to $U = 4$ to represent the horizontal, vertical, and diagonal orientations. The scale parameters were $v = 1$ and $v = 2$ for the first and second GDConv block respectively. A depthwise convolution with stride 2 is used in the second block to bring down the spatial resolution to half of the input size. In order to adjust the channel size, we implement Convolution 1×1 before every first block of each stage except for the first stage where the channel adjustment is handled by duplicating the image channel. Every second block of each stage is also followed by Batch Normalization and ReLU activation function.

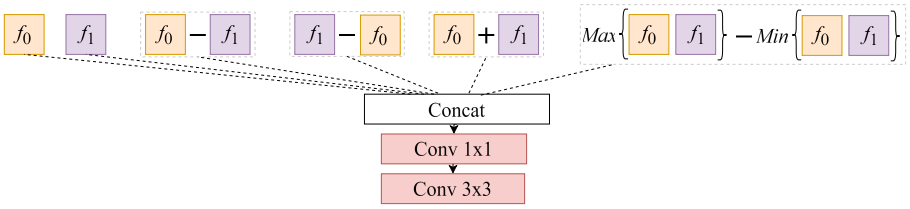

Fig. 4. The structure of Feature Conjuction Module.

2.2 Feature Conjunction Module

The extracted multi-resolution features concatenated from both encoders are processed by the Feature Conjunction Module (FCM). As shown in Fig. 4, we treat pre-change feature f_0 and post-change feature f_1 with several operations similar to [24], in order to explicitly represent the behavior of bi-temporal changes. Referenced from [24], $f_0 - f_1$ and $f_1 - f_0$ define the appearance and disapperance of the object, while $Max(f_0, f_1) - Min(f_0, f_1)$ intends to capture exchanging objects. We additionally add $f_0 + f_1$ to highlight the changed objects from the unchanged ones. All of the products of these operations are concatenated together with the original features f_0 and f_1, then they undergo a 1×1 convolution which will learn the important channels related to the changes and reduce the resolution from $\frac{H}{2^i} \times \frac{W}{2^i} \times 6C_i$ to $\frac{H}{2^i} \times \frac{W}{2^i} \times C_i$. A 3×3 convolution followed by Batch Normalization and ReLU activation are added as the last stage to further learn the relevant features.

3 Experiments

The performance of AYANet has been evaluated on 3 standard RS building change detection datasets. Comparison with State-of-the-Art (SOTA) methods is done both quantitatively using standard metrics, and qualitatively by visualizing the change maps. Some of the SOTA methods have been trained and tested in-house to ensure a fair comparison, while for other methods we report the measurements on the standard test split of the datasets published in papers.

3.1 Datasets

LEVIR-CD [4] consists of very high-resolution (VHR) RGB imagery highlighting the change in the development as well as the decline of buildings in Texas, USA. The dataset has 31333 change instances of various types of buildings such as large warehouses, tall apartments, villa residences, and small garages. For the experiment, we cropped 637 pairs of 0.5m resolution images with a size of 1024×1024 pixels to 256×256 patches without overlap. Following the default split of the dataset, the total pairs used for training/validation/test is 7120/1024/2048.

S2Looking [21] has 5000 bitemporal VHR side-viewing satellite imagery obtained at several off-nadir angles. The images are captured from various satellites such as GaoFen, SuperView, and BeiJing-2 with a size of 1024×1024 pixels and spatial resolutions ranging from 0.5m to 0.8m. The S2Looking dataset contains scenes of rural areas from around the world which adds the complexity of features of the dataset. The default split of train/validation/test consists of 3500/500/1000 pairs of images. For the experiment, the images were cropped into 256×256 patches which makes the final split adds up to 56000/8000/16000.

WHU-CD [14] records the building changes in Christchurch, New Zealand between 2012 and 2016. This dataset contains a pair of RGB aerial images with

0.2m spatial resolution. The training split has a size of 21243×15354 pixels and the test split is 11256×15354 pixels. Like the other two datasets, the images were cropped to 256×256, and we randomly split the images to 6096/762/762 for train/validation/test.

3.2 Implementation Details

The implementation of the proposed model was done using PyTorch and we run experiments on two GPUs: NVIDIA Quadro GV100 and NDVIA GeForce RTX4090. Input images were augmented geometrically (random flipping, random cropping) or photometrically (Gaussian blur). Weights of the model were randomly initialized. We trained the model using cross-entropy loss and AdamW optimizer (weight decay 0.01 and beta values $(0.9, 0.999)$). The model started the training with learning rate from 0.0001 linearly decaying to 0. We set the batch size to 8 and stopped the training at 300 epochs.

We utilized Precision, Recall, F1-score, and Intersection over Union (IoU) for the quantitative evaluation of our model.

Table 1. Quantitative results of AYANet and State-of-The-Art models on the LEVIR-CD dataset. The best result is highlighted in bold. Results of all SOTA models are as reported in the original papers.

Model	Precision	Recall	F1-score	IoU
AFCF3D-Net [26]	91.35%	90.17%	90.76%	83.08%
BIT [3]	89.24%	89.37%	89.31%	80.68%
ChangeFormer [2]	92.05%	88.80%	90.40%	82.48%
DMI-Net [10]	92.52%	89.95%	90.71%	82.99%
DUNE-CD [1]	92.27%	88.83%	90.52%	82.68%
FHD [18]	92.61%	89.61%	91.09%	83.63%
GVA-CD [13]	92.63%	87.88%	90.31%	82.51%
MSFCTNet[15]	92.06%	90.00%	91.02%	83.52%
STANet-PAM [4]	83.81%	**91.00%**	87.26%	77.40%
TINYCD [7]	**92.68%**	89.47%	91.05%	83.57%
AYANet	92.60%	90.25%	**91.41%**	**84.17%**

3.3 Comparison with SOTA

We listed the comparison of performances among our proposed model and several SOTA models on the LEVIR-CD dataset in Table 1. We make use of the default train/validation/test split, which has been used by many papers to report their results as well - which allows us a direct comparison with numerous SOTA methods. Furthermore, we only report results published in the original paper of the methods (which are the optimized results of the authors themselves) to guarantee a fair comparison with our method. The SOTA methods

listed in Table 1 represent a broad range of techniques and strategies. AFCF3D-Net [26] treats bitemporal images like a video and uses 3D CNN as its backbone. CNN-based models such as DMI-Net [10], DUNE-CD [1], FHD [18], STANet-PAM [4], and TINYCD [7] incorporate various attention mechanisms including self-, channel-, global-, local-, and cross-attention. GVA-CD [13] focuses on the feature difference method by taking into account the geometric structure of the object. BIT [3], ChangeFormer [2], and MSFCTNet [15] utilizes Transformer either in hybrid style or using it purely without CNN.

It can be observed that AYANet performs better than most of the listed SOTA models and outperforms all in terms of F1-score and IoU. This includes surpassing the models that implement Transformer, for example the CNN-Transformer-hybrid BIT by 2.10% and 3.49%, and the pure Transformer-based ChangeFormer by 1.01% and 1.69%. The proposed model also exceeds the performance of TINYCD by 0.36% and 0.60%. TINYCD also uses EfficientNet as their feature extractor, and a more sophisticated technique to manipulate the features extracted, as opposed to a simpler operation used in our FCM. We intuitively correlate this outcome to the addition of the Gabor Encoder helping extracting relevant features of the buildings such that only a simple feature manipulation is necessary to highlight the change of the buildings. Comparing to the method that explicitly target the pattern of the object on the image *i.e.* GVA-CD which focuses on geometric variation, our proposed model which targets building's textures, has a 1.10% higher F1-score and a 1.66% higher IoU.

Table 2. Quantitative results of AYANet and State-of-The-Art models on the S2Looking dataset. The best result is highlighted in bold. All SOTA models' results are reproduced.

Model	Precision	Recall	F1-score	IoU
BIT [3]	**73.99%**	52.73%	61.58%	44.49%
ChangeFormer [2]	68.04%	57.03%	62.05%	44.98%
STANet-PAM [4]	36.30%	**61.84%**	45.74%	29.65%
AYANet	69.37%	58.70%	**63.59%**	**46.62%**

Table 3. Quantitative results of AYANet and State-of-The-Art models on the WHU-CD dataset. The best result is highlighted in bold. All SOTA models' results are reproduced.

Model	Precision	Recall	F1-score	IoU
BIT [3]	87.65%	90.91%	89.25%	80.59%
ChangeFormer [2]	94.15%	85.52%	89.63%	81.20%
STANet-PAM [4]	70.65%	**93.54%**	80.50%	67.37%
AYANet	**95.56%**	92.89%	**94.21%**	**89.05%**

The evaluation on the S2Looking dataset and the WHU-CD dataset were also done. The S2Looking dataset covers a more challenging task where images are taken from off-nadir angles. Perhaps for this reason, relatively few papers report evaluation results on this difficult dataset and the reported IoU numbers are all below 50%. The WHU-CD dataset does not have a standard train/val/test split such that most of the literature present their results by randomly splitting the set. Unlike on the LEVIR-CD dataset, a fair comparison thus cannot be done based on only the published numbers. Therefore we re-trained and evaluated relevant SOTA methods on the same split of this dataset. We selected three models representing pure CNN (STANet-PAM), hybrid CNN and Transformer (BIT), and pure Transformer models (ChangeFormer). Table 2 and Table 3 show the quantitative results of AYANet and SOTA models on the S2Looking and the WHU-CD datasets respectively. The proposed model shows the highest performance in terms of F1-score and IoU among the evaluated methods on both datasets. There are minimal differences of 1.54% and 1.64% on the S2Looking dataset as well as significant improvement by 4.58% and 7.85% on the WHU-CD dataset, w.r.t. the second-best performer model *i.e.* ChangeFormer.

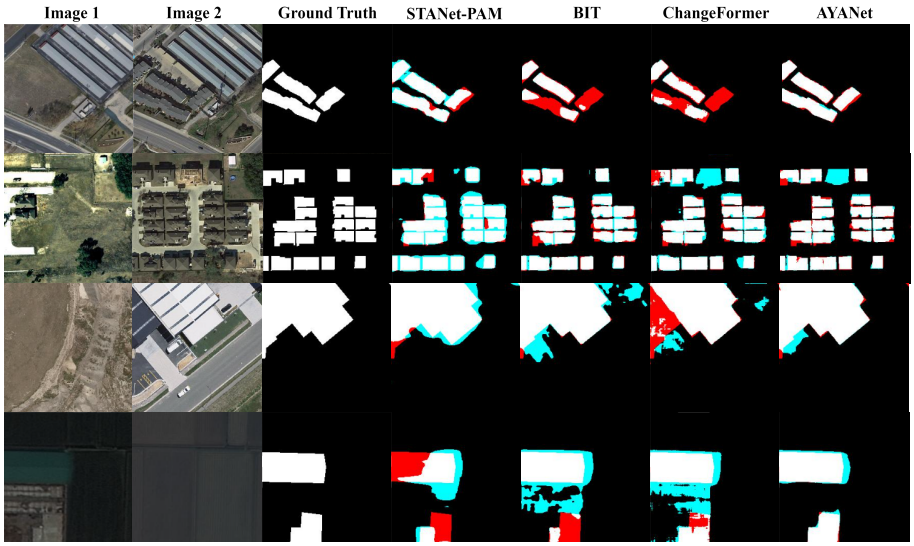

Fig. 5. Qualitative comparison of the change maps predicted by the proposed model and the SOTA models. The first 2 rows are the results on LEVIR-CD, while the third and fourth rows are from the WHU-CD and S2Looking datasets respectively. Color representation: TP (white), FP (light blue), TN (black), FN (red).

The qualitative comparison is shown in Fig. 5 where it can be seen that AYANet's change maps have less false positive (light blue area) and false negative (red area) in several cases, such as detecting changes of big building on the third

row of the figure, recognizing changes in smaller buildings on the first row, as well as change detection in the environment with poor lighting condition shown in the last row of the figure. Moreover, our model produces more precise masks like what we can observed in the second row of the figure where the boundary of the buildings located close to each other appears to be clearer.

We did an additional experiment to test our models that were trained on one particular dataset, with another datasets. The goal of this experiment is to show the proxy of generalization ability of the models. Results in Table 4 show the performance of the models trained on LEVIR-CD and tested on the WHU-CD dataset The proposed model outperforms other models in F1-score and IoU at least by 2.28% and 2.85%. The difference can also be confirmed in Fig. 6 where we can observe that AYANet's change maps have less false positive and false negative prediction. However, note that even our best performing model reaches only 60%, which is obviously much lower than anything trained on the WHU-CD dataset itself. Other SOTA models also reported the same tendency which may be related to the rather large difference in remote sensing imagery making change detection methods generalization challenging, partially because the pre and post-images are already registered so change detection requires only a pixel-wise analysis of changes thus more global changes are not learned well by these models and this is not even their goal to do so.

Table 4. The results of cross-dataset evaluation. All models are trained on the LEVIR-CD dataset and are tested on the WHU-CD dataset.

Model	Precision	Recall	F1-score	IoU
BIT [3]	58.36%	**79.52%**	67.32%	50.74%
ChangeFormer [2]	76.87%	70.10%	73.33%	57.89%
STANet-PAM [4]	28.31%	14.27%	18.98%	10.48%
AYANet	**77.60%**	73.66%	**75.58%**	**60.74%**

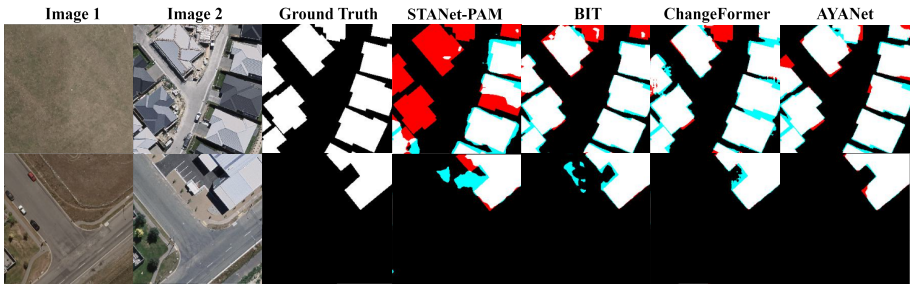

Fig. 6. Qualitative performance of the models on cross-dataset evaluation. Color representation: TP (white), FP (light blue), TN (black), FN (red).

3.4 Ablation Study

An ablation study was conducted to check how the proposed model behaves according to different settings of encoder. Table 5 shows the qualitative results of AYANet with the proposed *double encoder*, and the cases where we only use the Gabor Encoder as well as modified EfficientNet we use as the Encoder, exclusively. We find that using only the Gabor Encoder does not give the model a satisfactory performance as it only reaches 88.92% in F1-score, and 80.05% in IoU compared to AYANet which has 91.41% and 84.17% in the same metrics. However, adding the Gabor Encoder to the deep convolutional feature extractor, EfficientNet does increase the result, especially in IoU which implies a better agreement between the area of prediction and the ground truth. Some examples of the predictions shown in Fig. 7 confirm this IoU improvement. It can be seen that AYANet enhances boundary between buildings compared to the cases when we only use one single encoder.

Table 5. The experiments on the encoder of AYANet on the LEVIR-CD dataset.

Encoder	Precision	Recall	F1-score	IoU
Gabor	90.51%	87.38%	88.92%	80.05%
EfficientNet	92.15%	**90.35%**	91.24%	83.90%
AYANet	**92.60%**	90.25%	**91.41%**	**84.17%**

4 Conclusion

We introduce AYANet, a remote sensing change detection model using double encoder as the features extractor. The design of the double encoder includes CNN-based encoder and the Gabor Encoder which aims to extract the texture features of buildings. Moreover, Feature Conjunction Module is also proposed to process the extracted features from the double encoder in order to characterize the changes. Based on the comparison with SOTA models and the experimental evaluation, the proposed model demonstrates a good performance on 3 different building change detection datasets that have different characteristics. The ablation study confirms that adding the Gabor Encoder to the CNN-based encoder predicts a more accurate boundary between buildings. Future work will focus on a novel learning strategy that accommodates for domain adaptation, and unsupervised or semi-supervised learning approaches to cater to the problem of limited amount of data.

Acknowledgements The authors acknowledge the internship funding support by Inria, France (BMI-NF); the grants TKP2021-NVA-09 and K135728 of the National Research, Development and Innovation Fund, Hungary; and the scholarship by French Government for the first author.

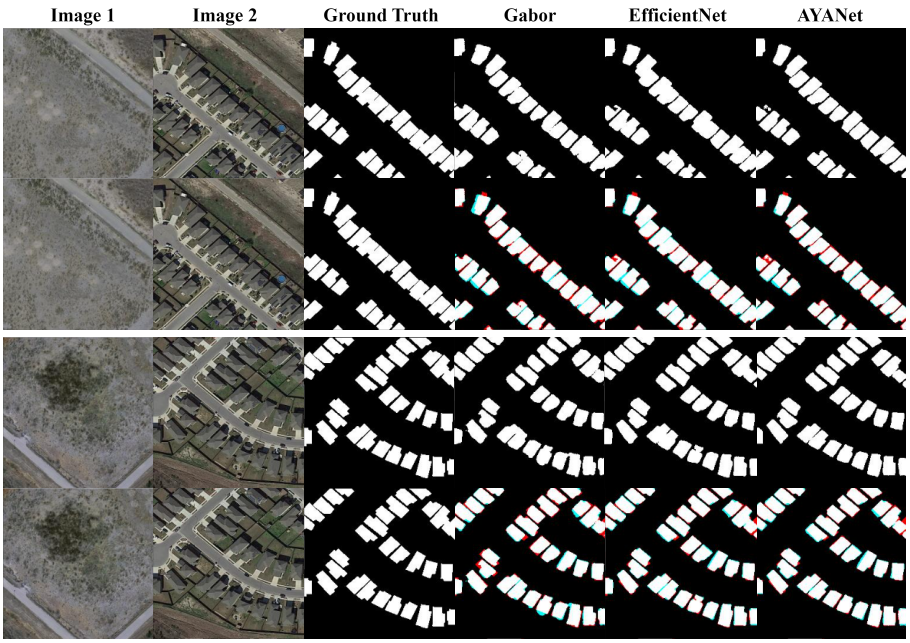

Fig. 7. The visualization of the ablation study on encoder. Visualization is done with and without color representation to make the boundary between buildings more visible. Color representation: TP (white), FP (light blue), TN (black), FN (red).

References

1. Adil, E., Yang, X., Huang, P., Liu, X., Tan, W., Yang, J.: Cascaded U-Net with training wheel attention module for change detection in satellite images. Remote Sensing **14**(24) (2022). https://doi.org/10.3390/rs14246361, https://www.mdpi.com/2072-4292/14/24/6361
2. Bandara, W.G.C., Patel, V.M.: A Transformer-based Siamese network for change detection. In: IEEE International Geoscience and Remote Sensing Symposium IGARSS. pp. 207–210 (2022). https://doi.org/10.1109/IGARSS46834.2022.9883686
3. Chen, H., Qi, Z., Shi, Z.: Remote sensing image change detection with Transformers. IEEE Transactions on Geoscience and Remote Sensing **60**, 1–14 (2022). https://doi.org/10.1109/TGRS.2021.3095166
4. Chen, H., Shi, Z.: A spatial-temporal attention-based method and a new dataset for remote sensing image change detection. Remote Sensing **12**, 1662 (2020). https://doi.org/https://doi.org/10.3390/rs12101662
5. Chen, S., Yang, K., Stiefelhagen, R.: DR-TANet: Dynamic receptive temporal attention network for street scene change detection. In: 2021 IEEE Intelligent Vehicles Symposium (IV). pp. 502–509 (2021). https://doi.org/10.1109/IV48863.2021.9575362

6. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1800–1807 (2016), https://api.semanticscholar.org/CorpusID:2375110
7. Codegoni, A., Lombardi, G., Ferrari, A.: TINYCD: a (not so) deep learning model for change detection. Neural Computing and Applications **35**, 8471–8486 (2023). https://doi.org/10.1007/s00521-022-08122-3, https://doi.org/10.1007/s00521-022-08122-3
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
9. Du, P., Liu, S., Gamba, P., Tan, K., Xia, J.: Fusion of difference images for change detection over urban areas. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **5**(4), 1076–1086 (2012). https://doi.org/10.1109/JSTARS.2012.2200879
10. Feng, Y., Jiang, J., Xu, H., Zheng, J.: Change detection on remote sensing images using dual-branch multilevel intertemporal network. IEEE Transactions on Geoscience and Remote Sensing **61**, 1–15 (2023). https://doi.org/10.1109/TGRS.2023.3241257
11. Gabor, D.: Theory of communication. Journal of Institution of Electrical Engineers **93**(3), 429–457 (1946)
12. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7132–7141 (2018). https://doi.org/10.1109/CVPR.2018.00745
13. Huo, S., Zhou, Y., Zhang, L., Feng, Y., Xiang, W., Kung, S.Y.: Geometric variation adaptive network for remote sensing image change detection. IEEE Transactions on Geoscience and Remote Sensing **62**, 1–14 (2024). https://doi.org/10.1109/TGRS.2024.3363431
14. Ji, S., Wei, S., Lu, M.: Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set. IEEE Transactions on Geoscience and Remote Sensing **57**(1), 574–586 (2019). https://doi.org/10.1109/TGRS.2018.2858817
15. Jiang, M., Chen, Y., Dong, Z., Liu, X., Zhang, X., Zhang, H.: Multiscale fusion CNN-Transformer network for high-resolution remote sensing image change detection. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **17**, 5280–5293 (2024). https://doi.org/10.1109/JSTARS.2024.3361507
16. Liu, C., Wechsler, H.: Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition. IEEE Transactions on Image Processing **11**(4), 467–476 (2002). https://doi.org/10.1109/TIP.2002.999679
17. Luan, S., Chen, C., Zhang, B., Han, J., Liu, J.: Gabor convolutional networks. IEEE Transactions on Image Processing **27**(9), 4357–4366 (2018). https://doi.org/10.1109/TIP.2018.2835143
18. Pei, G., Zhang, L.: Feature hierarchical differentiation for remote sensing image change detection. IEEE Geoscience and Remote Sensing Letters **19**, 1–5 (2022). https://doi.org/10.1109/LGRS.2022.3193502
19. Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., Dosovitskiy, A.: Do vision transformers see like convolutional neural networks? In: Neural Information Processing Systems (2021), https://api.semanticscholar.org/CorpusID:237213700
20. Seydi, S.T., Hasanlou, M., Chanussot, J., Ghamisi, P.: BDD-Net+: A building damage detection framework based on modified Coat-Net. IEEE Journal of Se-

lected Topics in Applied Earth Observations and Remote Sensing **16**, 4232–4247 (2023). https://doi.org/10.1109/JSTARS.2023.3267847
21. Shen, L., Lu, Y., Chen, H., Wei, H., Xie, D., Yue, J., Chen, R., Lv, S., Jiang, B.: S2Looking: A satellite side-looking dataset for building change detection. Remote Sensing **13**(24) (2021). https://doi.org/10.3390/rs13245094, https://www.mdpi.com/2072-4292/13/24/5094
22. Sun, Y., Zhang, X., Huang, J., Wang, H., Xin, Q.: Fine-grained building change detection from very high-spatial-resolution remote sensing images based on deep multitask learning. IEEE Geoscience and Remote Sensing Letters **19**, 1–5 (2022). https://doi.org/10.1109/LGRS.2020.3018858
23. Tan, M., Le, Q.: EfficientNet: Rethinking model scaling for convolutional neural networks. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 6105–6114. PMLR (09–15 Jun 2019), https://proceedings.mlr.press/v97/tan19a.html
24. Wang, G.H., Gao, B.B., Wang, C.: How to reduce change detection to semantic segmentation. Pattern Recognition **138**, 109384 (2023). https://doi.org/https://doi.org/10.1016/j.patcog.2023.109384, https://www.sciencedirect.com/science/article/pii/S0031320323000857
25. Wiskott, L., Fellous, J.M., Krüger, N., von der Malsburg, C.: Face recognition by elastic bunch graph matching. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**, 355–396 (01 1999). https://doi.org/10.1007/3-540-63460-6_150
26. Ye, Y., Wang, M., Zhou, L., Lei, G., Fan, J., Qin, Y.: Adjacent-level feature cross-fusion with 3-D CNN for remote sensing image change detection. IEEE Transactions on Geoscience and Remote Sensing **61**, 1–14 (2023). https://doi.org/10.1109/TGRS.2023.3305499
27. Zhai, X., Kolesnikov, A., Houlsby, N., Beyer, L.: Scaling vision Transformers. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12104–12113 (June 2022)