BiasUNet: Learning Change Detection over Sentinel-2 Image Pairs

Maria Pegia
mpegia@iti.gr
Information Technologies Institute
Centre of Research & Technology - Hellas
Thessaloniki, Greece

Anastasia Moumtzidou
moumtzid@iti.gr
Information Technologies Institute
Centre of Research & Technology - Hellas
Thessaloniki, Greece

Ilias Gialampoukidis
heliasgj@iti.gr
Information Technologies Institute
Centre of Research & Technology - Hellas
Thessaloniki, Greece

Björn Þór Jónsson
bjth@itu.dk
Information Technologies Institute
Centre of Research & Technology - Hellas
Copenhagen, Denmark

Stefanos Vrochidis
stefanos@iti.gr
Information Technologies Institute
Centre of Research & Technology - Hellas
Thessaloniki, Greece

Ioannis Kompatsiaris
ikom@iti.gr
Information Technologies Institute
Centre of Research & Technology - Hellas
Thessaloniki, Greece

ABSTRACT

The availability of satellite images has increased due to the fast development of remote sensing technology. As a result several deep learning change detection methods have been developed to capture spatial changes from multi temporal satellite images that are of great importance in remote sensing, monitoring environmental changes and land use. Recently, a supervised deep learning network called FresUNet has been proposed, which performs a pixel-level change detection from image pairs. In this paper, we extend this method by inserting a Bayesian framework that uses Monte Carlo Dropout, motivated by a recent work in image segmentation. The proposed **Bayesian FresUNet** (BiasUNet) approach is shown to outperform four state-of-the-art deep learning networks on Sentinel-2 ONERA Satellite Change Detection (OSCD) benchmark dataset, both in terms of precision and quality.

CCS CONCEPTS

• **Computing methodologies** → **Supervised learning**; *Neural networks*; • **Applied Computing** → Earth and atmospheric sciences.

KEYWORDS

Change Detection, Deep Learning, Supervised Learning, Sentinel-2

ACM Reference Format:
Maria Pegia, Anastasia Moumtzidou, Ilias Gialampoukidis, Björn Þór Jónsson, Stefanos Vrochidis, and Ioannis Kompatsiaris. 2022. BiasUNet: Learning Change Detection over Sentinel-2 Image Pairs. In *Proceedings of CBMI '22: International Conference on Content-based Multimedia Indexing (CBMI '22).* ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX

CBMI '22, September 14–16, 2022, Graz, Austria

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

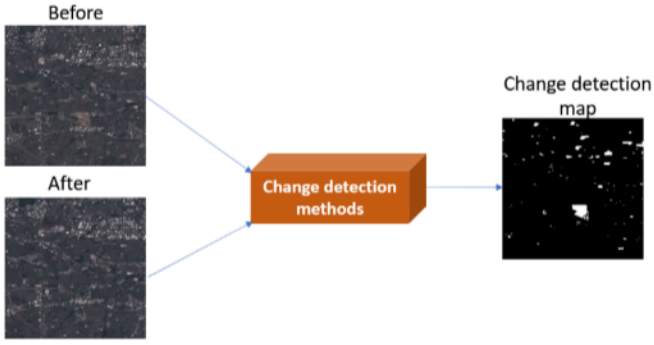

Figure 1: Overview of change detection methods

1 INTRODUCTION

With the progress of remote sensing technology, remote sensing platforms have become capable of collecting a wide range of data. These available data are used as resources for Earth monitoring by detecting changes on the land surface. Change detection is a procedure that detects changes on the same geographical area by observing a set of images captured during different periods. Due to its usage in a plethora of real-world applications, such as fire detection, environmental monitoring, disaster monitoring, urban change analysis and land management, it has attracted the interest of research society and several works have been published.

While the definition of term "change" may vary between applications, change detection can be considered as a well defined classification problem, that aims to assign a binary label per pixel based on a co-registered images pair of a given region taken at different times. A positive label indicates that the area corresponding to that pixel has changed between the acquisitions, while a zero label that there is no difference between the acquisitions. Figure 1 illustrates this binary map, with white for change and black for no-change label.

The main challenges of change detection are (a) the semantic gap between the low-level feature representing and high-level semantics in the images, and (b) the curse of dimensionality, since visual

descriptors usually have hundreds or even thousands of dimensions. Furthermore, shadows or weather changes include noise or atmospheric variations and make change detection a very challenging task. Although a wide range of methods have been developed in remote sensing data analysis over the years, new more efficient and accurate methods are needed.

Change detection methods are categorized into pixel-based change detection [1, 2, 4, 6–11, 15, 15, 17, 17, 18, 22–25] and object-based change detection [19, 21] techniques. The former attempt to identify whether or not a change has occurred at pixel level in the co-registered images pair, whereas the latter attempt to first group pixels that belong to the same object and then use the object's information (e.g., colour, and shape) to determine whether the object has changed during time. Early change detection approaches analyzed pixels directly using manually crafted techniques and descriptors. However, more recent change detection methods use either unsupervised deep learning [1, 4, 6, 8, 9, 11, 18, 22, 23] or supervised deep learning [2, 7, 10, 15, 17, 24, 25] methods to generate change maps, with the latter approach shown to perform better.

Deep learning architectures have achieved state-of-the-art results in almost all classification tasks, nevertheless they still make over-confident decisions. Recent works apply Bayesian deep learning to Convolution Neural Networks (CNN) by adding both a measure of uncertainty and weight regularization to their predictions to surpass this problem.

In this work, we propose a pixel-based change detection approach that involves the use of Bayesian learning for learning the distribution of the weights of the layers of neural network, and the use of a majority voting strategy for producing the predicted change map of a multi temporal image pair. To the best of our knowledge, this is the first deep learning approach that uses a Bayesian framework with Monte Carlo (MC) Dropout for change detection.

The main contributions of this paper are summarized as follows:

- proposal of a novel and more effective approach for change detection.
- employment of a Bayesian network with Monte Carlo Dropout that learns weights of the layers.
- validation of the effectiveness of the proposed approach using a benchmark Sentinel-2 dataset for different band combinations.

The paper is structured as follows. Section 2 presents relevant recent works and Section 3 gives details of the proposed approach. Experimental results are presented in Section 4 and the paper concludes with a brief summary in Section 5.

2 RELATED WORK

In this section, we discuss existing state-of-the-art pixel-based methods unsupervised and supervised methods proposed for change detection.

Unsupervised methods [9] use thresholding criteria (e.g., change vector analysis) for identifying the changed patterns. Bruzzone et al. [8] propose two methods; the first allows the automatic selection of the decision threshold that minimizes the overall change detection error probability, while the second analyzes the difference image by considering the spatial-contextual information included in the neighborhood of each pixel. In [23], the authors propose a method that pre-processes images with a controlled adaptive iterative filtering, compares multi temporal images according to a standard log-ratio operator and generates the change detection map by analysing of the log-ratio image. In [18], a method is presented that combines principal component analysis and k-means clustering to distinguish change from unchanged pixels. The deep neural network learning algorithm in [6] uses auto-encoders for classification changed pixels from unchanged ones. In [11], the authors propose a modular, scalable, and metric free single shot change detection method that exploits a decomposed interconnected graphical model formulation with relaxed similarity constraints. Liu et al [5] propose an unsupervised method that uses iterative training for learning the latent relation between two heterogeneous images. In [4], a change detection method is proposed that is based on keypoints matching, evaluation, and grouping, and doesn't require any image co-registration. Du et al [1] propose deep slow feature analysis model that consists of a two symmetric deep networks for projecting the input data of bi-temporal imagery, uses change vector analysis to find unchanged pixels with high confidence as training samples and calculates the change intensity with chi-square distance and threshold algorithm.

Supervised methods, on the other hand, require having labelled training data. Volpi et al [10] propose a method that combines Support Vector Machines (SVM) and contextual information for supervised change detection. Le Saux et al [2] propose a descriptor Change-Index Histogram of Oriented Gradients that uses SVM. In [7], the authors propose a method based on deep neural networks for image change detection, while in [24, 25] their methods are based on convolutional neural network (CNN) architectures. In [15], the authors propose an Early Fusion (EF) architecture that concatenates the two image patches before passing them through the network for change detection. Daudt et al [15] propose two Siamese encoder-decoder fully convolutional (FCN) architecture (FC-Siam-conc, FC-Siam-diff) for change detection and show that using all available multispectral bands for change detection results in better results. In the work [17], an encoder-decoder FCN architecture (FresUnet) for change detection is proposed that uses the predicted land cover information to help the prediction of changes. A relatively recent supervised Bayesian UNet architecture (BU-Net) has been proposed in [3] for image segmentation. Authors of [13] provide both a semantic segmentation and uncertainty maps.

Recently, some works have been published on change detection using Bayesian framework. Specifically, Gharbi et al [20] propose a hierarchical Bayesian model for change detection that applies Bernoulli-based models to change detection and transforms it to a denoising problem. Moreover, Lin et al in [12] propose a time-series-based normalized S1 intensity image for the flooding detection using Bayesian probability functions to decrease and increase the backscattering intensity. Unlike traditional methods, the weights of a Bayesian network can have a wide range of values and learn more accurately the data distribution. Moreover, from the literature can be observed that Bayesian models are usually preferred in difficult problems [3, 12, 14, 20] because they use probability for uncertainty model representations as well as the uncertainty of the output of the model.

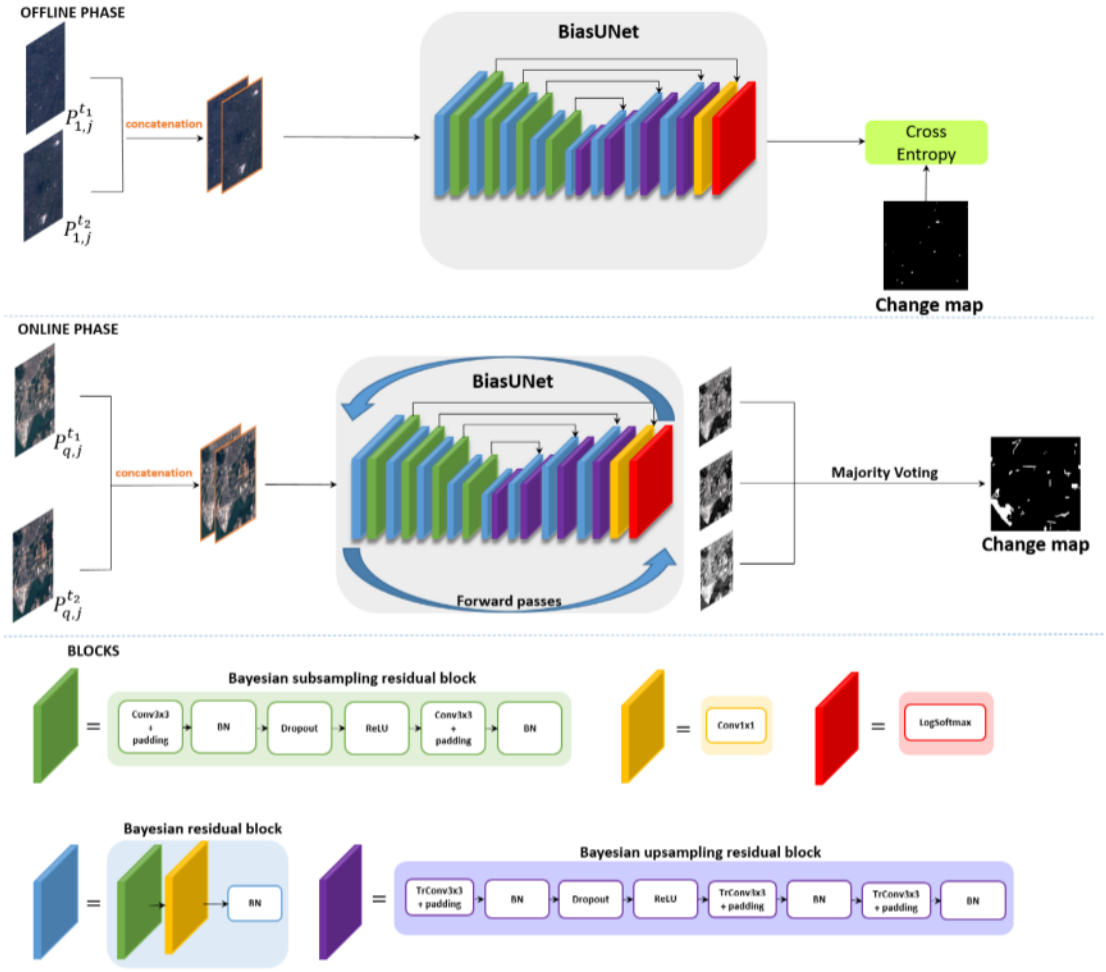

Figure 2: Framework of the proposed method.

Contrary to these approaches, we propose a supervised Bayesian encode-decoder FCN model for change detection. From our knowledge, this work is the first FCN architecture that contains MC Dropout for change detection. Therefore, inspired by the technique described in [3], we proceed with applying MC Dropout and majority voting to learn change maps for change detection problem.

3 METHODOLOGY

The proposed method belongs to FCN architectures that learn to perform change detection solely from change detection datasets without pretraining or transfer learning from other datasets. Similar to the state-of-the-art FCN methods [15, 17], the approach works with patches in order to improve accuracy and speed without affecting significantly the training times. This approach can receive variable number of neurons in its input layer, thus making it more flexible, given the appropriate memory.

Figure 2 shows an overview of the proposed fully convolutional architecture for change detection. The Offline phase corresponds to the training procedure, whereas the online phase corresponds to the testing procedure. The structure of network is the same both for the training and the testing phases.

In the training phase, the method concatenates the two patches before passing them through the network and feeds the output to the encoder-decoder FCN BiasUNet network. After that, the concatenated patch goes through alternating Bayesian residual blocks and Bayesian subsampling residual blocks for four layers in the encoder part. The outcome passes through alternating Bayesian residual blocks and Bayesian upsampling residual blocks for four layers in the decoder part. There are skip connections between Bayesian residual blocks from encoder part to decoder part for keeping spatial details that are present in the earlier layers of the network and producing more accurate class prediction. Next, it uses a convolution layer with kernel size 1 and a log softmax activation function for producing the change map of the multi-temporal patch pair. Finally, it uses cross entropy loss to compute the score between outcome and ground truth patch and this value is fed to the network.

According to the testing phase, the method concatenates the patches and feeds them to the BiasUNet. In contrast to the training phase, each concatenated input goes through the network for a

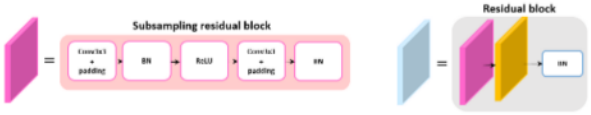

Figure 3: Encoder part architecture of FresUNet.

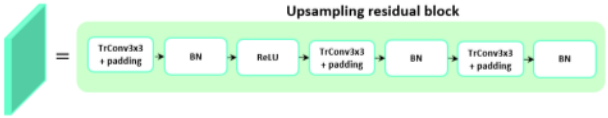

Figure 4: Decoder part architecture of FresUNet.

small number of passes and applies majority voting function to the outcomes to produce the predicted change map of the patch pair.

3.1 Notation

Let $\mathcal{I}$ be the set of image pairs of size $|\mathcal{I}| = N$. Each instance I_i of $\mathcal{I}$ consists of an image pair $(I_i^{t_1}, I_i^{t_2})$ that is captured from the same location at different times t_1, t_2, respectively. Let M be the number of patches that an image will be divided into. For each image $I_i^{t_1}$ we define its j-th patch as $P_{i,j}^{t_1}$, for $j = 1, \cdot, M$. Let O be the training set of size $|O| = n = NM$, with O_k its k-th instance. Each instance O_k corresponds to a patch image pair $(P_{k,j}^{t_1}, P_{k,j}^{t_2})$ from images $I_k^{t_1}$, $I_k^{t_2}$, respectively. We also denote with $Q = (I_q^{t_1}, I_q^{t_2})$ an image pair for testing and let $(P_{q,j}^{t_1}, P_{q,j}^{t_2})$ be the j-th patch pair. We denote with CM_i, CM_i^{gt} the predicted and ground truth change map of image pair I_i with $CM_{i,j}, CM_{i,j}^{gt}$ the j-th pixel of each, respectively. Let f be the number of total forward passes. We define with w, h, c the width, height and number of channels of each image patch. Let x be the value of decoder output $h \times w \times 1$. We denote with z_r the random activation or inactivation coefficients and M_r the weight matrix before dropout layer. Let p_r be the activation probability for r-th layer, T a threshold value of majority voting function, and q an instance of online phase.

3.2 FresUNet

As the proposed method builds on FresUNet, we describe that method first, and then outline the differences in the following subsection. As described before, the FresUNet concatenates the multi-temporal patches both in training and testing phase before feeding them into the network. The concatenation is done on the channel dimension. Therefore, starting with a patch pair $(P_{i,j}^{t_1}, P_{i,j}^{t_2})$ from images $I_i^{t_1}, I_i^{t_2}$, each of dimensions $h \times w \times c$, the input to the network will be of size $h \times w \times 2c$.

Then, the input goes through residual blocks and subsampling residual blocks for the four layers of the encoder part. Each residual block, as well as each subsampling residual block, consists of a combination of convolution layers (Conv3×3), batch normalization (BN) and ReLU functions (Figure 3).

Then, the outcome goes through residual blocks and upsampling residual blocks for the four layers of the decoder part. Each upsampling residual block consist of a combination of transpose convolution layers (TrConv3×3), batch normalization (BN) and ReLU functions (Figure 4).

FresUnet uses log softmax activation function (Eq. (1)) to the output x_i from decoder part to produce the predicted change map CM_i. After that it uses negative log likelihood (Eq. (2)) for updating the weights of the network in the training phase.

$$log\_softmax(x) = log(\frac{e^{x_i}}{\sum_k e^{x_k}}) \tag{1}$$

$$nll\_loss(CM_i, CM_i^{gt}) = \\ -\sum_{j=1}^{c}(CM_{i,j}^{gt} log(CM_{i,j}) + (1 - CM_{i,j}^{gt}) log(1 - CM_{i,j})) \tag{2}$$

3.3 BiasUNet

In the proposed framework, the input is the image patches, which are then concatenated, and use the previous defined blocks, but with some modifications. Inspired by the work of [3] and as already mentioned in the section 2, we insert Bayesian learning layers to encoder and decoder part with MC dropout layers behind each ReLU layer and perform a little number of forward passes to network. The dropout layers are active in both the training and the prediction phases. MC Dropout gives an uncertainty for each weight matrix in Eq. (3):

$$\begin{aligned} z_r &\sim Bernoulli(p_r) \\ W_r &= M_r \times diag(z_r) \end{aligned} \tag{3}$$

During the prediction phase, the network receives as input a patch image pair $(P_{q,j}^{t_1}, P_{q,j}^{t_2})$, performs f forward passes of the input, and gets $S = \{CM_q\}_{k=1}^{k=f}$, a set of $|S| = f$ log softmax outputs. The Bayesian model produces different predictions for the same input data, since the weights are sampled from a distribution. Next, a majority voting function (Eq. (4)) is applied to the predictions to compute the final map.

$$majority\_voting(S) = mean(\{CM_q\}_{k=1}^{k=f} > T) \tag{4}$$

Eventually, a cross entropy loss (Eq. (5)) is used between the predicted results and the ground truth values of each instance.

$$ce\_loss(CM_q, CM_q^{gt}) = \sum_{j=1}^{c}(CM_{q,j}^{gt} log(CM_{q,j})) \tag{5}$$

4 EXPERIMENTS

In this section, we describe the dataset used for evaluation, the performance measures, and the experimental results for different band combinations.

4.1 Dataset

The evaluation of our method and the comparison with existing SOTA methods is done on the widely used benchmark ONERA

Table 1: Evaluation metrics on change detection dataset bands.

No. Bands (c)	Method	Class accuracy [no change, change]	Dice	Kappa	Net-accuracy	Net-loss	Precision	Recall
3	UNet [15]	[**99.151**, 12.923]	0.20088	0.18224	94.74082	0.41417	0.45079	0.12923
	SiamUNet-conc [15]	[98.611, 27.694]	0.36092	0.33730	**94.98349**	0.35196	0.41798	0.27694
	SiamUNet-diff [15]	[98.489, 25.672]	0.33405	0.30936	94.76448	0.39295	**0.47804**	0.25672
	FresUNet [17]	[96.618, 52.578]	0.48837	0.45871	94.36521	0.30837	0.45592	0.52577
	BiasUNet	[94.853, **64.191**]	**0.49439**	**0.46046**	93.28461	**0.27346**	0.40201	**0.64198**
10	UNet [15]	[**99.495**, 23.321]	0.35149	0.33473	**95.59858**	0.38424	**0.71333**	0.23321
	SiamUNet-conc [15]	[98.681, 35.137]	0.44030	0.41807	95.43091	0.34500	0.58951	0.35137
	SiamUNet-diff [15]	[99.158, 24.723]	0.35231	0.33268	95.35049	0.47696	0.61275	0.24723
	FresUNet [17]	[97.110, 55.682]	0.53211	0.50557	94.99129	0.27646	0.50949	0.55682
	BiasUNet	[94.440, **64.211**]	**0.54291**	**0.51858**	93.10010	**0.26756**	0.43818	**0.64211**
13	UNet [15]	[**99.214**, 34.997]	0.46793	0.44793	**95.92925**	0.33623	**0.70587**	0.34997
	SiamUNet-conc [15]	[99.123, 28.351]	0.39255	0.93643	95.51206	0.35407	0.63789	0.28351
	SiamUNet-diff [15]	[99.263, 23.432]	0.34181	0.93403	95.38436	0.46795	0.63153	0.23431
	FresUNet [17]	[94.440, **68.240**]	0.50291	0.46858	93.10010	0.26756	0.23818	**0.68239**
	BiasUNet	[97.379, 64.775]	**0.56541**	**0.94112**	95.39436	**0.25119**	0.55644	0.64775

dataset [16]. ONERA consists of 24 pairs of multispectral images with their ground truth pixel annotation taken from the Sentinel-2 satellites between 2015 and 2018. Following the ONERA creator's guidelines, we split the dataset in train and test set. Thus, we end up with 14 images for training and 10 for testing. Images vary in size and spatial resolution between 10m, 20m and 60m.

4.2 Settings

In this section, we present the settings used in the experiments. Specifically, the model parameter T from Eq. 4 is set to 0.002. We experiment also with the number of the bands used as input, i.e., (i) only the RGB bands ($c = 3$), (ii) all Sentinel-2 bands with resolution <= 20m ($c = 10$) and, (iii) all Sentinel-2 bands ($c = 13$), in order to illustrate the performance of methods according to the number of bands. Furthermore, we perform experiments that check how the different number of epochs affects the network performance. We evaluate the model performance for 30, 50, 100 epochs and present the only when 13 bands are used, because under these setting we observe that even though the FresUNet outperforms all networks, our model presents less fluctuations in change class and better results in no change class in contrast to the other networks. Moreover, we use patch size = 96 and batch size = 16 similar to [15]. The number of forward passes f is set to 3 according to the work in [13], while the Adam optimizer with default parameters is used. According to the experiments of the right graph of Figure 5 the parameter of MC Dropout is set to 0.45.

The proposed BiasUNet method is implemented in Python 3.8.10 with PyTorch, which is powered by a workstation with Intel Xeon Silver 4210 CPU (2.20 GHz, 10 cores, and 125GB RAM) with GeForce RTX2080 Ti TURBO 11GB GDDR6 NVIDIA GeForce and 18.04.5 LTS Ubuntu software. We compare our approach with four state of the art methods UNet[1], SiamUNet-conc[1], SiamUNet-diff[1] and FresUNet[1] .

[1]https://github.com/rcdaudt/fully_convolutional_change_detection

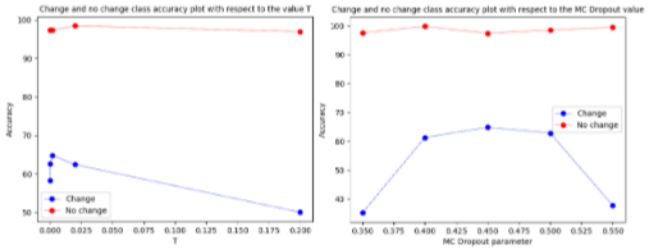

Figure 5: Change and no change class accuracy of BiasUNet according to the values of T and MC Dropout for 30 epochs and 13 bands.

The evaluation metrics used for measuring the performance of each method [15] are the following: class accuracy, dice, kappa, net-accuracy, net-loss, precision and recall. As regards to the time needed for the training and testing of the models, all models need about 2 hours for 30 epochs of training and less than 1 second for testing.

4.3 Results

In this section, we present the evaluation metrics for change and no change accuracy for four state-of-the-art models on ONERA dataset. Additional experiments are included based on number of training epochs, parameter sensitive and some visual results.

Table 1 shows the evaluation metrics for the proposed BiasUNet approach, compared to the four state-of-the-art methods. Kappa and dice measures the consistency and the similarity between predicted and ground-truth classes, respectively. Higher precision shows that the model returns relevant results, while higher recall shows that it returns most of the relevant results (whether or not irrelevant ones

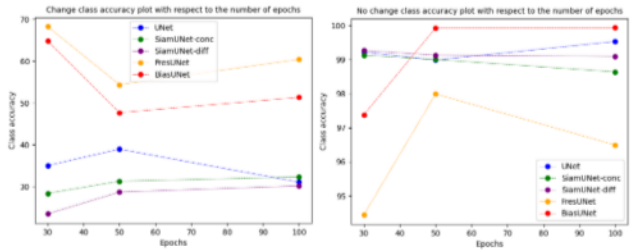

Figure 6: Change and No-change accuracy of all models for 30, 50, 100 epochs.

are also returned). Higher values for net-accuracy and lower values for net-loss show that the model learns to detect better changes.

As Table 1 shows, the BiasUNet outperforms almost all baselines with different band sizes (c = 3, 10, 13 bands) in ONERA benchmark dataset, which demonstrates its effectiveness. We observe that as the number of bands decreases, the performance of the state-of-the-art models mainly decreases. Another conclusion that could be drawn is that the performance of many state-of-the-art methods vary significantly across bands, while the performance of BiasUNet is relatively stable across the bands, reflecting its capability of utilizing smaller bands to better change detection information. Specifically, it captures the second best score of change class accuracy with value 68.24% for 13 bands, whereas for 3 and 13 bands captures the best score approximately 64.19% and 64.21%, respectively.

Figure 6 depicts the change and no change classes accuracy for 13 bands for all the models evaluated in respect to the number of epochs. One can notice that all models detect better the no change class for all epochs. It should be also noted that although FresUNet achieves the best score of change accuracy (~ 68%), it achieves the lowest score of no change accuracy (~ 94%) in 30 epochs. Nevertheless, BiasUNet presents a more stable trade-off between change (~ 65%) and no change (~ 95%) class accuracy in contrast to the other models.

Moreover, experiments of BiasUNet according to the change class and no change class accuracy with threshold parameter T of majority voting function for 30 epochs are given in Figure 5. BiasUNet presents a stable behavior in no change class according to threshold as well as the MC Dropout values. Additionally, it captures a peak T = 0.002 and for MC Dropout 0.45. We notice that the change class has a Bayesian distribution of the values related to MC Dropout values.

Figure 7 offers a qualitative evaluation of the methods compared as it depicts the change maps results for three query image pairs from the ONERA dataset for each of the evaluated methods. The first three columns depict the query at time T_1, T_2 and ground truth (GT) respectively. Each of the following columns of the tableau corresponds to the returned results of one method. Moreover, the different lines correspond to different number of bands used as input. Specifically, the first 3 lines show the outcome when 3 bands are used as input, the next 3 lines the results when 10 bands are used and, finally the last 3 lines the results in case of 13 bands. As far as the color labels, white color corresponds to true positive, black to true negative, green to false positive and magenta to false

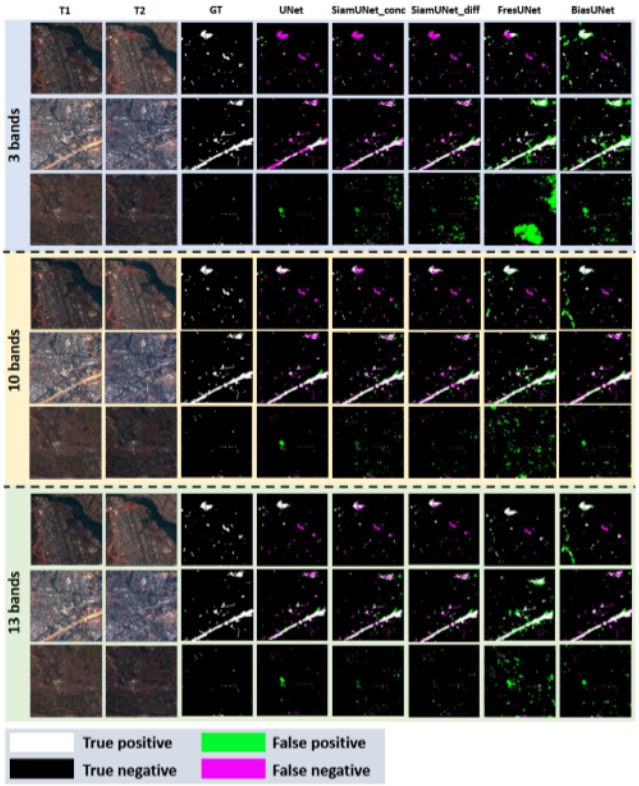

Figure 7: Change detection performance of the proposed BiasUNet and compared networks on the ONERA benchmark dataset with different number of bands.

negative. As the figure shows, the BiasUNet returns more stable and less noisy results for the given multi-temporal image pairs in contrast to the state-of-the-art networks.

5 CONCLUSION

In this work, we have modified the FresUNet network by using Bayesian learning. The advantages of Bayesian learning is that it can adapt data from their statistical properties and include regularization parameters in the estimation process. The experiments performed show that the proposed BiasUNet network outperforms almost four state of the art networks with more stable trade-off between change/no change class accuracy over ONERA dataset. In the future, we plan to investigate the evaluation of the areas where the network predicted the wrong class change with high confidence.

ACKNOWLEDGEMENTS

This work was supported by the EU's Horizon 2020 research and innovation programme under grant agreement H2020-101004152 CALLISTO.

REFERENCES

[1] Du B., Ru L., and Zhang L. 2019. Unsupervised Deep Slow Feature Analysis for Change Detection in Multi-Temporal Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing* 57, 12 (Sept. 2019), 9976 – 9992. https://doi.org/10.1109/TGRS.2019.2930682

[2] Le Saux B. and Randrianarivo H. 2013. Urban change detection in SAR images by interactive learning. *IEEE International Geoscience and Remote Sensing Symposium* (Feb. 2013), 3990–3993. https://doi.org/10.1109/IGARSS.2013.6723707

[3] Dechesne C., Lassalle P., and Lefèvre S. 2021. Bayesian U-Net: Estimating Uncertainty in Semantic Segmentation of Earth Observation Images. *Remote Sensing* 13, 19 (Oct. 2021). https://doi.org/10.3390/rs131938363

[4] Liu G., Gousseau Y., and Tupin F. 2019. A contrario comparison of local descriptors for change detection in very high spatial resolution satellite images of urban areas. *IEEE Transactions on Geoscience and Remote Sensing* 57, 6 (jun 2019), 39045–3918. https://doi.org/10.1109/TGRS.2018.2888985

[5] Liu J., Gong M., Qin K., and Zhang P. 2015. A deep convolutional coupling network for change detection based on heterogeneous optical and radar images. *IEEE Transactions on Neural Networks and Learning Systems* 29, 3 (jun 2015), 545–559. https://doi.org/10.1109/TNNLS.2016.2636227

[6] Zhao J., Gong M., Liu J., and Jiao L. 2014. Deep learning to classify difference image for image change detection. *IEEE International Joint Conference on Neural Networks (IJCNN)* (Sept. 2014), 411–417. https://doi.org/10.1109/IJCNN.2014.6889510

[7] Zhao J., Gong M., Liu J., and Jiao L. 2014. Deep learning to classify difference image for image change detection. *IEEE International Joint Conference on Neural Networks* (July 2014), 411–417. https://doi.org/10.1109/IJCNN.2014.6889510

[8] Bruzzone L. and Prieto D.F. 2000. Automatic analysis of the difference image for unsupervised change detection. *IEEE Geoscience and Remote Sensing Letters* 38, 3 (May 2000), 1171–1182. https://doi.org/10.1109/36.843009

[9] Hussain M., Chen D., Cheng A., Wei H., and Stanley D. 2013. Change detection from remotely sensed images: From pixel-based to object-based approaches. *Photogramm. Remote Sensing* 80 (June 2013), 91–106. https://doi.org/10.1016/j.isprsjprs.2013.03.006

[10] Volpi M., Tuia D., Bovolo F., Kanevski M., and Bruzzone L. 2013. Supervised change detection in VHR images using contextual information and support vector machines. *International Journal of Applied Earth Observation and Geoinformation* 20 (Feb. 2013), 77–85. https://doi.org/10.1016/j.jag.2011.10.013

[11] Vakalopoulou M., Karantzalos K., Komodakis N., and Paragios N. 2015. Simultaneous registration and change detection in multitemporal, very high resolution remote sensing data. *IEEE Conference on Computer Vision and Pattern Recognition Workshops* (jun 2015), 61–69. https://doi.org/10.1109/CVPRW.2015.7301384

[12] Lin Y. N., Yun S.-H., Bhardwaj A., and Hill E. M. 2019. Urban Flood Detection with Sentinel-1 Multi-Temporal Synthetic Aperture Radar (SAR) Observations in a Bayesian Framework: A Case Study for Hurricane Matthew. *Remote Sensing* 11, 15 (July 2019). https://doi.org/10.3390/rs11151778

[13] Pourdarbani R., Sabzi S., Hernández M., García-Mateos G., Kalantari D., and Molina-Martínez J. M. 2019. Comparison of Different Classifiers and the Majority Voting Rule for the Detection of Plum Fruits in Garden Conditions. *Remote Sensing* 11, 21 (Sept. 2019). https://doi.org/10.3390/rs11212546

[14] J. N. K. Rao. 2011. Impact of Frequentist and Bayesian Methods on Survey Sampling Practice: A Selective Appraisal. *Statist. Sci.* 26, 2 (May 2011), 240–256. https://doi.org/10.1214/10-STS346

[15] Daudt R.C., Le Saux B., and Boulch A. 2018. Fully convolutional siamese networks for change detection. *IEEE International Conference on Image Processing* (Oct. 2018), 4063–4067. https://doi.org/10.1109/ICIP.2018.8451652

[16] Daudt R.C., Le Saux B., Boulch A., and Gousseau Y. 2018. Urban change detection for multispectral earth observation using convolutional neural networks. *IEEE International Geoscience and Remote Sensing Symposium* (July 2018), 2115–2118. https://doi.org/10.1109/IGARSS.2018.8518015

[17] Daudt R.C., Le Saux B., Boulch A., and Gousseau Y. 2019. Multitask learning for large-scale semantic change detection. *Computer Vision and Image Understanding* 187 (Oct. 2019). https://doi.org/10.1016/j.cviu.2019.07.003

[18] Celik T. 2009. Unsupervised change detection in satellite images using principal component analysis and k-means clustering. *IEEE Geoscience and Remote Sensing Letters* 6, 4 (Oct. 2009), 772–776. https://doi.org/10.1109/LGRS.2009.2025059

[19] Liu T., Yang L., and Lunga D. 2021. Change detection using deep learning approach with object-based image analysis. *Remote Sensing of Environment* 256 (April 2021). https://doi.org/10.1016/j.rse.2021.112308

[20] Gharbi W., Chaari L., and Benazza-Benyahia A. 2020. Unsupervised Bayesian change detection for remotely sensed images. *Signal, Image and Video Processing* 15, 2 (April 2020), 205–213. https://doi.org/10.1007/s11760-020-01738-9

[21] Wang X., Liu S., Du P., Liang H., Xia J., and Li Y. 2018. Object-Based Change Detection in Urban Areas from High Spatial Resolution Images Based on Multiple Features and Ensemble Learning. *Remote Sensing* 10, 2 (Dec. 2018). https://doi.org/10.3390/rs10020276

[22] Zheng X., Chen X., Lu X., and Sun B. 2021. Unsupervised Change Detection by Cross-Resolution Difference Learning. *IEEE Transactions on Geoscience and Remote Sensing* 60 (May 2021). https://doi.org/10.1109/TGRS.2021.3079907

[23] Bazi Y., Bruzzone L., and Melgani F. 2005. An unsupervised approach based on the generalized Gaussian model to automatic change detection in multitemporal SAR images. *IEEE Geoscience and Remote Sensing Letters* 43, 4 (Oct. 2005), 874–887. https://doi.org/10.1109/TGRS.2004.842441

[24] Chen Y., Ouyang X., and Agam G. 2018. Mfcnet: End-to-end approach for change detection in images. *IEEE International Conference on Image Processing* (Oct. 2018), 4008–4012. https://doi.org/10.1109/ICIP.2018.8451392

[25] Zhan Y., Fu K., Yan M., Sun X., Wang H., and Qiu X. 2017. Change detection based on deep siamese convolutional network for optical aerial images. *IEEE Geoscience and Remote Sensing Letters* 14, 10 (July 2017), 1845–1849. https://doi.org/10.1109/LGRS.2017.2738149