A Hybrid Metaheuristic and Deep Learning Approach for Change Detection in Remote Sensing Data

Yacine Slimani
Department of Computer Science
Laboratory of Intelligent Systems
University of Ferhat Abbas Setif 1
Setif, Algeria
slimaniy09@univ-setif.dz

Rachid Hedjam
Department of Computer Science
Sultan Qaboos University
Muscat, Oman
rachid.hedjam@squ.edu.om

Received: 6 August 2022 | Revised: 17 August 2022 | Accepted: 20 August 2022

Abstract-This study aimed to adapt Convolutional Neural Networks (CNN) to solve the problem of change detection using remote sensing imagery. Specifically, the goal was to investigate the impact of each CNN layer to detect changes between two satellite images acquired on two different dates. As low-level CNN layers detect fine details (small changes) and higher-level layers detect coarse details (large changes), the idea was to assign a weight to each layer and use a genetic algorithm based on a training dataset to generalize the detection process on the test dataset. The results showed the effectiveness of the proposed approach based on two real-life datasets.

Keywords-change detection; remote sensing; deep learning; convolutional neural networks; genetic algorithms

I. INTRODUCTION

The aim of binary Change Detection (CD) in remote sensing is to compare two images acquired at two different dates to detect meaningful differences [1]. Usually, two CD approaches are used: supervised or unsupervised. Supervised CD requires temporal reference data for the training phase [2, 3], while unsupervised CD is based on a direct comparison of input images without using labeled data [4-6]. In general, unsupervised binary CD techniques consist of two steps: i) compute the difference between the features of the two input images to generate a difference image or change index, and ii) generate the binary change map by segmenting (thresholding) the difference image computed in the first step into change and no-change regions. However, traditional CD methods that use handcrafted features are not effective in complex situations, because the designed features cannot accurately capture high- and medium-level image representations [7]. Recently, deep learning has emerged and has become a state-of-the-art approach for CD. Deep learning is very effective in extracting representative features from low, middle, and higher image representation levels. The advantages of deep learning are that, it learns discriminant features and computes them automatically without relying on the involvement of an expert.

Several deep-learning methods for CD have been proposed. In [8], a method was proposed to compute the difference image using a backpropagation algorithm and a deep belief network. A deep belief network learns low and high-level features around a pixel neighborhood and the backpropagation algorithm builds the difference image using training samples. Finally, a simple segmentation algorithm was used to compute the binary change map. A method was proposed in [9] that combined deep features, saliency detection, and Convolutional Neural Networks (CNNs) to compute the change. A patch-based Siamese Neural Network was presented in [10], where external images, whose textures resembled the changing area, were used to generate genuine and imposter pair samples for the training process. A method that combined CNN features to create a single higher feature vector was proposed in [7], using the pixel-wise Euclidean distance between the extracted feature vectors after having been transformed into matrices to compute the change map. A review of the deep learning-based CD methods can be found in [11]

This study extends [12] and is related to [7]. The difference between this study and [7] lies in the combination process of the CNN layers. In [7], all CNN layers were combined into a single feature vector, but this study proposes the assignment of weights to the CNN layers before combining them. In [12], the weights were binary (i.e. 1 for considering a layer and 0 for not) and assigned manually. This study used a Genetic Algorithm (GA) to automatically learn the weights based on training data. The GA aims to find the best weights that lead to the best match between the CD reference (ground truth) and the change map detected by the proposed algorithm. Therefore, the weight vector can be seen as a mask, where the goal is to demonstrate that the layers can be assigned different weights before being combined to detect different regions that represent the changes between the two input images. The assumption was that, to detect large changes, high-level layers are assigned higher weights, i.e. considered more.

II. THE PROPOSED APPROACH

Usually, the spatial changes in remote sensing images are specific patterns with special features in terms of color, shape, and texture. Therefore, their CNN characteristics are different from those of the same location in the image before the change. In other terms, an unchanged spatial area between the two

Corresponding author: Yacine Slimani

inputs should have almost similar CNN features, whereas the changed areas have different. Thus, it seems reasonable to compute the features of the two input images using the same CNN structures and then compute their difference to generate the difference image, where the brighter pixels represent the changed areas due to the larger difference values. To detect the changed areas, the difference image can be segmented by a thresholding method into "changed" and "unchanged" classes. In a CNN, the lower layers capture low-lever image features such as edges, color, and gradient orientation, while the mid-high level layers capture coarse patterns of the images that can represent whole objects in the images [13]. Since the changes are almost a random natural process, they may affect areas with different sizes from fine to coarse. Therefore, detecting changes between the two input images based on the difference between their last CNN feature is not effective. To overcome this limitation, a new change detection method was proposed, which is an improvement over [7]. The procedure obeys the following: i) extract low, mid and higher features from each image using a pre-trained CNN (e.g. VGG19 [14]), ii) resize the layers to the same size and combine them into a single feature vector for each image, iii) reshape the two feature vectors into square matrices, and finally iv) compute the Euclidean distance between the two matrices to generate the difference image, which will be segmented into two classes, "changed" and "unchanged".

In [12], a binary weight was assigned manually to each layer to include it or not in the combination process of the layers. In other terms, the binary vector of weights played a mask role that allowed or prevented some layers from the combination process. If the weight of a given layer was equal to one (1), it meant that this layer was included in the combination process, while if it was equal to zero (0) it was not. Assigning weights to layers is application dependent. In practice, it is best to consider high-level layers when detecting large changes, while low-level layers should be taken into account when detecting minor changes.

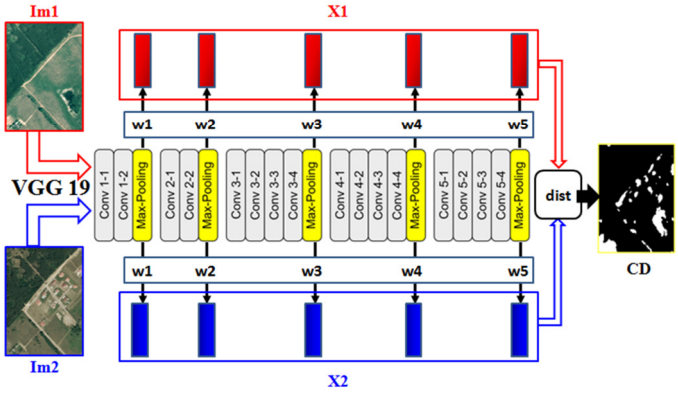


Fig. 1. Convolutional feature-based change detection with VGG19.

In the training phase, N image tuples were used $\{(Im1; Im2; Tr)\}_i$, i=1…N, where $Im1$ and $Im2$ are the images before and after a change and Tr is the corresponding ground truth. Each image is divided into M patches of size d×d. In other terms, there is a set of S=N×M patches before the change, i.e. $\{P_{j1}, j=1...S\}$, a set of patches after the change, i.e. $\{P_{j2}, j=1...S\}$, and the same number of ground truth patches. Each patch is fed to a VGG19 CNN to extract 5 feature maps from 5 different layers. This study used the 3rd, 6th, 10th, 14th, and 18th layers of the VGG19. From each input patch, 5 layers (feature maps) were extracted and resized to the same size.

Formally, let $[X_{j11}, X_{j12}, X_{j13}, X_{j14}, X_{j15}]$ be the list of the feature maps extracted from P_{j1}, and $[X_{j21}, X_{j22}, X_{j23}, X_{j24}, X_{j25}]$ be the list of the feature maps extracted from P_{j2}. Thus, the corresponding weighted feature maps are:

$$F_{j1} = [w_1 \times X_{j11}, w_2 \times X_{j12}, w_3 \times X_{j13}, w_4 \times X_{j14}, w_5 \times X_{j15}]$$

$$F_{j2} = [w_1 \times X_{j21}, w_2 \times X_{j22}, w_3 \times X_{j23}, w_4 \times X_{j24}, w_5 \times X_{j25}]$$

where W=[w_1, w_2 …, w_5] are the continuous weights learned by a GA. The difference image for the j^{th} patch-pair is then computed as follows (see Figure 2):

$$DI_j = \text{dist}(F_{j1}, F_{j2}) = \sqrt{\sum_{k=1}^{5}\left(w_k \times X_{j1k} - w_k \times X_{j2k}\right)^2} \quad (1)$$

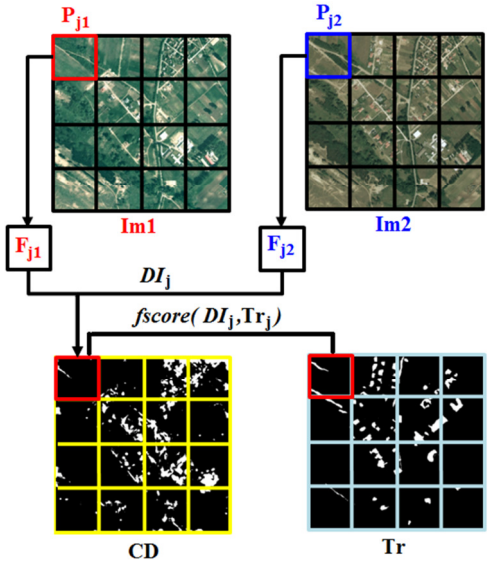


Fig. 2. CNN-based features vectors process.

The goal is to learn the weight vector W using a GA that maximizes the fitness (f-score) between the detected change and the corresponding ground truth for all the patches used in the training phase [15]. Once the optimal weights are learned, they will be used in the test phase. The overall GA for learning the weights, shown in Figure 3, was:

- Step 1 (Initialize population): The first step of GA is to randomly create and initialize the chromosomes of the initial population. Nb_Ind vectors $W = \{W^{(i)} | i = 1..Nb\_Ind\}$ are generated. Each $W^{(i)}$ is a chromosome with five (5) real-valued genes $[w_k^{(i)} | k = 1...5]$.

- Step 2 (Evaluation): The fitness function measures the fitness of the change map between each pair of the patch and the corresponding ground truth patches as follows:

$$Fitness(W^{(i)}) = \frac{\sum_{j=1}^{S} fscore(DI_j, Tr_j)}{S} \quad (2)$$

```
Algorithm 1: GA-based Learning Process.
  Data: BaseImages,NB Ind, NB Gen, ProbSelect,
        ProbCross, ProbMutation
  Result: W
  /*CurrentGeneration = InitGeneration(NB Ind)*/;
  for i=1..Nb Ind do
  |   for Gene [j=1..5] do
  |   |   Wj = random(0, 1);
  |   end
  |   CurrentGeneration[i]=W;
  end
  /*Evaluation of population*/;
  for i=1..Nb Ind do
  |   Fitness[i] = CNN(CurrentGeneration[i]);
  end
  OldGeneration = Sort(CurrentGeneration);
  foreach Generation [i=1..NB Gen] do
  |   /*Selection(PropSelect)*/;
  |   for i=1..Nb Ind*ProbSelect do
  |   |   CurrentGeneration.Append(OldGeneration[i])
  |   end
  |   /*Crossover(ProbCross)*/;
  |   for i=1..Nb Ind*ProCross do
  |   |   WFather = Select(OldGeneration);
  |   |   WMother = Select(OldGeneration);
  |   |   for Gene [j=1..5] do
  |   |   |   Wj = Choice(WFather[j],WMother[j])
  |   |   end
  |   |   CurrentGeneration.Append(W)
  |   end
  |   /*Duel(ProbDuel)*/;
  |   for i=1..Nb Ind*ProCross do
  |   |   Ind1 = Select(OldGeneration);
  |   |   Ind2 = Select(OldGeneration);
  |   |   if Fitness[Ind1] > Fitness[Ind2] then
  |   |   |   W = OldGeneration[Ind1];
  |   |   else
  |   |   |   W = OldGeneration[Ind2];
  |   |   end
  |   |   CurrentGeneration.Append(W)
  |   end
  |   /*Mutation(ProbMutation)*/;
  |   for i=1..Nb Ind*ProbMutation do
  |   |   Ind Mut = randint(0, Nb Ind);
  |   |   W = CurrentGeneration[Ind Mut];
  |   |   Gene Mut = randint(0, 5);
  |   |   WGene Mut = random(0, 1);
  |   end
  |   /*Evaluation of population*/;
  |   for i=1..Nb Ind do
  |   |   Fitness[i] = CNN(CurrentGeneration[i]);
  |   end
  |   OldGeneration = Sort(CurrentGeneration)
  end
  return best(W)
```

Fig. 3. GA-based learning process algorithm.

- Step 3 (Selection): The elitism selection method selects the best $W^{(i)}$ chromosomes from the previous population to integrate them into the next population. According to the best fitness function values $Fitness(W^{(i)})$, a portion of (*ProbSelect%*) from the precedent population was selected to breed a new generation.
- Step 4 (Crossover): The crossover method creates a portion of (*ProbCross%*) from the precedent population. A one-point crossover method was used.
- Step 5 (Mutation): The goal of this function was to introduce diversity into the population. A portion of (*ProbMut%*) was chosen and a random value was assigned to one randomly chosen gene.
- Redo steps 2, 3, 4, and 5 until stability (no change in the fitness) (see Algorithm 1). Finally, the best *W* is chosen to be used in the test phase.

III. EXPERIMENTATION AND EVALUATION

A. Dataset Description

Two datasets were used to evaluate the proposed change detection framework, namely the SZTAKI AirChange Benchmark set [16] and the Onera Satellite Change Detection dataset [17]. The SZTAKI AirChange Benchmark set contains 13 aerial image pairs of 952×640 pixels with a resolution of 1.5m/pixel, and binary change masks (a ground truth defined by experts). The Onera Satellite Change Detection dataset consists of 24 pairs of multispectral images taken using the Sentinel-2 satellites between 2015 and 2018. The locations were picked from all over the world, Brazil, the United States, Europe, the Middle East, and Asia. For each location, registered pairs of 13-band multispectral satellite images are required. The images vary in spatial resolution between 10m, 20m, and 60m. The pixel-level change ground truth is provided for the image pairs. The annotated changes focus on urban changes, such as new buildings or new roads.

B. GA Parameter Setting

In the training phase, a pre-trained VGG19 was used to extract the feature maps. The GA requires several parameters to search for the optimal layer weights:

- Number of generations *Nb_Gen*=200
- Number of Individuals *Nb_Ind*=100
- Probability of selection *ProbSelect*=40%
- Probability of crossover *ProbCross*=40%
- Probability of mutation *ProbMutation*=20%

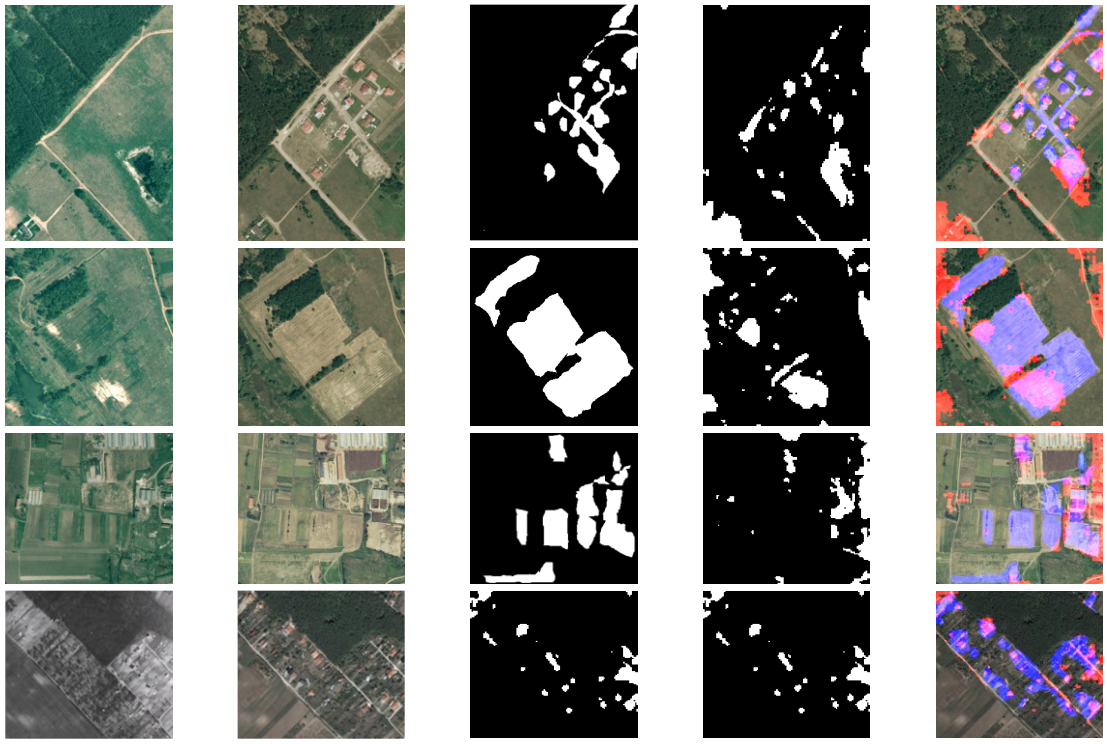
C. Results, Evaluation, and Comparison to Other Methods

The proposed method was compared with two classes of existing change detection methods, traditional and deep learning based. The traditional methods were the Iteratively Reweighted Multivariate Alteration Detection Method (IMAD) for change detection [18], Slow Feature Analysis (SFA) algorithm for change detection [19], Principal Component Analysis and k-means clustering (PCA-Kmeans) [20], and Change Vector Analysis (CVA) [21]. The deep learning-based methods were two simple CNN-based without layer weighting: the VGG19 [7], and the ResNet50 [22].

Table I shows the change detection results using different methods in terms of f-score, recall, precision, and accuracy, based on the SZTAKI dataset. Based on f-score and accuracy, the proposed method gave the best results with the test images Szada3, Tiszadob2, and Archive (accuracy was 0.90, 0.85, and 0.85 respectively). In the case of the Szada4 image, the proposed method gave better results than the original VGG19 algorithm (accuracy was 0.70 versus 0.69) but had lower accuracy than IMAD, which can be justified by the poor quality of the ground truth.

TABLE I. CHANGE DETECTION RESULTS OF THE SZTAKI AIRCHANGE DATASET

Images	Methods	f-score	Recall	Precision	Accuracy	White pixels detected
Szada3	**IMAD**	0.45	0.63	0.35	0.86	4261
	ISFA	0.35	0.38	0.33	0.89	2553
	PCA-Kmeans	0.37	0.50	0.29	0.84	3368
	CVA	0.16	0.60	0.09	0.42	4062
	VGG 19	0.31	0.32	0.30	0.87	2166
	ResNet50	0.41	**0.66**	0.29	0.82	4462
	Proposed	**0.45**	0.45	**0.44**	**0.90**	3078
Szada4	**IMAD**	0.57	0.50	**0.67**	**0.78**	14438
	ISFA	0.42	0.39	0.45	0.67	11350
	PCA-Kmeans	**0.59**	0.57	0.61	0.76	16548
	CVA	0.38	**0.57**	0.29	0.45	16418
	VGG 19	0.27	0.20	0.45	0.69	5646
	ResNet50	0.50	0.45	0.57	0.73	12884
	Proposed	0.30	0.24	0.48	0.70	6865
Tiszadob2	**IMAD**	0.39	0.48	0.32	0.70	7608
	ISFA	0.36	0.47	0.29	0.66	7555
	PCA-Kmeans	**0.48**	0.63	0.39	0.74	10011
	CVA	0.32	**0.67**	0.21	0.45	10734
	VGG 19	0.23	0.17	0.36	0.78	2664
	ResNet50	0.36	0.36	0.37	0.75	5769
	Proposed	0.44	0.35	**0.60**	**0.85**	4945
Archive	**IMAD**	**0.40**	0.45	0.36	0.76	6392
	ISFA	0.31	0.22	0.56	0.83	3118
	PCA-Kmeans	0.40	0.46	0.35	0.76	6544
	CVA	0.31	**0.71**	0.20	0.45	10092
	VGG 19	0.31	0.20	0.65	0.84	2863
	ResNet50	0.45	0.48	0.42	0.80	6776
	Proposed	0.32	0.21	**0.70**	**0.85**	2958

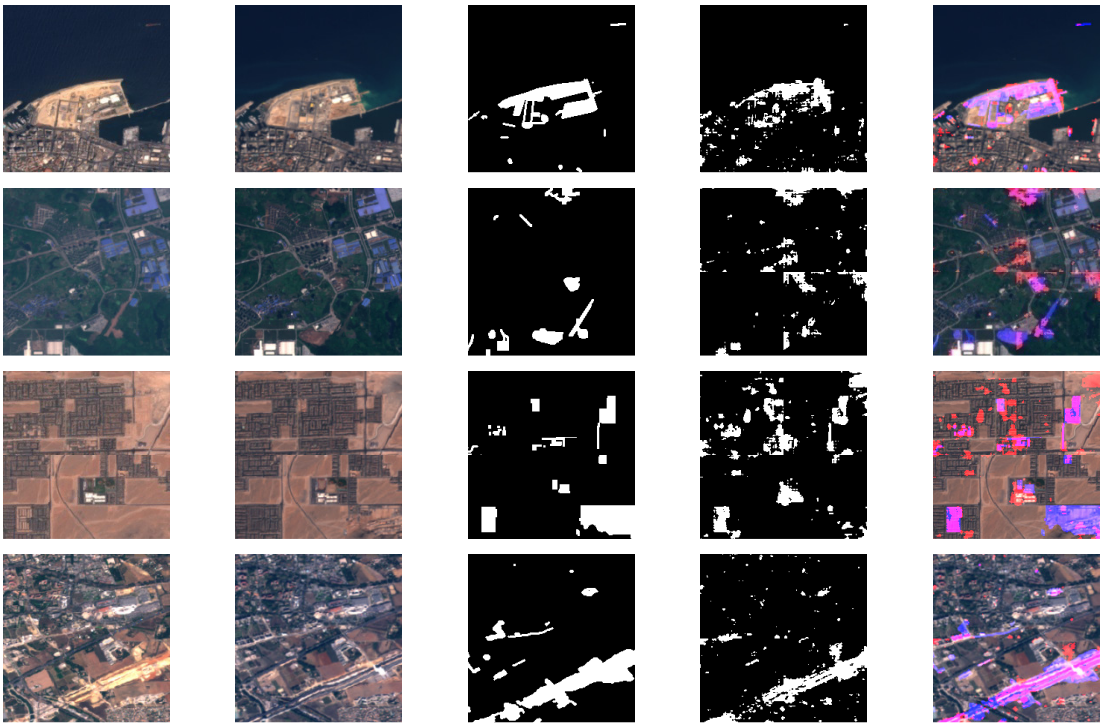


(a) (b) (c) (d) (e)

Fig. 4. Subjective results. From left to right: (a) image (1) before change, (b) image (2) after change, (c) ground-truth, (d) CD map, (e) overlay of CD on image (2). Images from SZTAKI, from top to bottom, Szada3, Szada4, Tiszadob2, Archive.

TABLE II. CHANGE DETECTION RESULTS OF THE ONERA SATELLITE DATASET

Images	Methods	f-score	Recall	Precision	Accuracy	White pixels detected
Beirut	**IMAD**	0.46	0.58	0.39	0.90	2262
	ISFA	0.42	**0.94**	0.27	0.80	3682
	PCA-Kmeans	0.54	0.71	0.43	0.90	2804
	CVA	0.17	0.82	0.09	0.35	3231
	VGG 19	0.26	0.17	0.53	0.92	670
	ResNet50	0.43	0.55	0.36	0.89	2165
	Proposed	**0.55**	0.57	**0.67**	**0.94**	2252
Chongqing	**IMAD**	**0.56**	0.75	**0.45**	0.94	1897
	ISFA	0.52	**0.77**	0.39	0.93	1943
	PCA-Kmeans	0.42	0.53	0.35	0.93	1340
	CVA	0.07	0.50	0.04	0.36	12682
	VGG 19	0.22	0.19	0.27	0.93	492
	ResNet50	0.36	0.45	0.31	0.92	1125
	Proposed	0.41	0.43	0.44	**0.94**	1095
Las Vegas	**IMAD**	0.60	0.61	0.59	**0.92**	3183
	ISFA	0.33	0.55	0.24	0.77	2858
	PCA-Kmeans	0.54	0.49	**0.60**	0.91	2558
	CVA	0.17	**0.65**	0.10	0.37	3346
	VGG 19	0.37	0.42	0.32	0.85	2196
	ResNet50	**0.64**	0.94	0.48	0.89	4853
	Proposed	0.40	0.46	0.49	0.90	2362
Montpellier	**IMAD**	0.67	0.65	0.69	**0.92**	4141
	ISFA	0.62	0.54	**0.72**	0.92	3451
	PCA-Kmeans	**0.69**	**0.76**	0.62	0.91	4825
	CVA	0.26	0.75	0.16	0.47	4725
	VGG 19	0.51	0.40	0.71	0.90	2554
	ResNet50	0.63	0.63	0.62	0.91	4014
	Proposed	0.44	0.35	0.60	0.85	4945



(a) (b) (c) (d) (e)

Fig. 5. Subjective results. From left to right: (a) image (1) before change, (b) image (2) after change, (c) ground-truth, (d) CD map, € overlay of CD on image (2). Images from Onera. From top to bottom, Beirut, Chongqing, Las Vegas, Montpellier.

Table II shows the change detection results of the different methods in terms of f-score, recall, precision, sensitivity, and accuracy, based on the Onera Satellite dataset. The proposed method was again better than VGG19 in any case, and it was the best for Beirut and Chongqing test images, but weaker than IMAD for Las Vegas and Montpellier.

Figures 4 and 5 show the results of change detection on the test image from the SZTAKI and the Onera datasets respectively. The last column shows the overlay of the change detection map on the image after the change, where the changes are highlighted in red and the ground truth is highlighted in cyan blue. It can be noted that most of the changes that occurred were detected. Unfortunately, the method detected unwanted changes, which means that it triggered more false alarms. Finally, the results obtained show that the proposed method gave better results than [7] and [12], unveiling the problem with the weight of the layer. Moreover, the GA learning phase made it possible to detect fine and coarse details, by finding the best weight vector.

IV. CONCLUSION

This paper presented an artificial intelligence-based change detection approach for remote sensing. The challenge of finding the best weighted CNN layers to the change detection problem was solved using a genetic algorithm. The proposed approach combines a CNN (pre-trained VGG19) and genetic algorithm to build a near-optimal weight vector. This vector was combined with the feature maps to compute their difference and produce the change map. The purpose of this study was to investigate the adaptation of CNNs to detect changes in remote-sensing images. Two datasets were used to show that it is recommended to specify the CNN layers to be used to detect different changes. More precisely, detecting small changes requires the use of the first CNN layers, whereas large changes require the use of the last layers. Therefore, a natural way to solve this problem is to use a meta-heuristic optimization method to find the best weights for the layers to be combined to generate the relevant feature maps. Future work could investigate other CNN architectures and metaheuristic search methods and extend the work on a larger number of datasets.

REFERENCES

[1] P. Coppin, E. Lambin, I. Jonckheere, and B. Muys, "Digital change detection methods in natural ecosystem monitoring: a review," in *Analysis of Multi-Temporal Remote Sensing Images*, vol. Volume 2, Singapore: World Scientific, 2002, pp. 3–36.

[2] B. Demir, F. Bovolo, and L. Bruzzone, "Classification of Time Series of Multispectral Images With Limited Training Data," *IEEE Transactions on Image Processing*, vol. 22, no. 8, pp. 3219–3233, Dec. 2013, https://doi.org/10.1109/TIP.2013.2259838.

[3] J. Collomb, P. Balland, P. Francescato, Y. Gardet, D. Leh, and P. Saffré, "Thermomechanical Optimization and Comparison of a Low Thermal Inertia Mold with Rectangular Heating Channels and a Conventional Mold," *Advances in Materials Science and Engineering*, vol. 2019, May 2019, Art. no. e3261972, https://doi.org/10.1155/2019/3261972.

[4] T. L. Dammalage and N. T. Jayasinghe, "Land-Use Change and Its Impact on Urban Flooding: A Case Study on Colombo District Flood on May 2016," *Engineering, Technology & Applied Science Research*, vol. 9, no. 2, pp. 3887–3891, Apr. 2019, https://doi.org/10.48084/etasr.2578.

[5] S. Liu, L. Bruzzone, F. Bovolo, and P. Du, "Unsupervised Multitemporal Spectral Unmixing for Detecting Multiple Changes in Hyperspectral Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 5, pp. 2733–2748, Feb. 2016, https://doi.org/10.1109/TGRS.2015.2505183.

[6] X. Zheng, X. Chen, X. Lu, and B. Sun, "Unsupervised Change Detection by Cross-Resolution Difference Learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2022, https://doi.org/10.1109/TGRS.2021.3079907.

[7] A. M. E. Amin, Q. Liu, and Y. Wang, "Convolutional neural network features based change detection in satellite images," in *First International Workshop on Pattern Recognition*, Jul. 2016, vol. 10011, pp. 181–186, https://doi.org/10.1117/12.2243798.

[8] G. Cao, B. Wang, H.-C. Xavier, D. Yang, and J. Southworth, "A new difference image creation method based on deep neural networks for change detection in remote-sensing images," *International Journal of Remote Sensing*, vol. 38, no. 23, pp. 7161–7175, Dec. 2017, https://doi.org/10.1080/01431161.2017.1371861.

[9] D. Peng and H. Guan, "Unsupervised change detection method based on saliency analysis and convolutional neural network," *Journal of Applied Remote Sensing*, vol. 13, no. 2, May 2019, Art. no. 024512, https://doi.org/10.1117/1.JRS.13.024512.

[10] R. Hedjam, A. Abdesselam, and F. Melgani, "Change Detection from Unlabeled Remote Sensing Images Using SIAMESE ANN," in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, Yokohama, Japan, Jul. 2019, pp. 1530–1533, https://doi.org/10.1109/IGARSS.2019.8898672.

[11] L. Khelifi and M. Mignotte, "Deep Learning for Change Detection in Remote Sensing Images: Comprehensive Review and Meta-Analysis," *IEEE Access*, vol. 8, pp. 126385–126400, 2020, https://doi.org/10.1109/ACCESS.2020.3008036.

[12] Y. Slimani and R. Hedjam, "Which Cnn Layer For Which Change? A Cnn Adaptation Approach For Change Detection In Remote Sensing Data," in *2020 Mediterranean and Middle-East Geoscience and Remote Sensing Symposium (M2GARSS)*, Tunis, Tunisia, Mar. 2020, pp. 5–8, https://doi.org/10.1109/M2GARSS47143.2020.9105168.

[13] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *2011 International Conference on Computer Vision*, Aug. 2011, pp. 2018–2025, https://doi.org/10.1109/ICCV.2011.6126474.

[14] S. Sahel, M. Alsahafi, M. Alghamdi, and T. Alsubait, "Logo Detection Using Deep Learning with Pretrained CNN Models," *Engineering, Technology & Applied Science Research*, vol. 11, no. 1, pp. 6724–6729, Feb. 2021, https://doi.org/10.48084/etasr.3919.

[15] B. K. Alsaidi, B. J. Al-Khafaji, and S. A. A. Wahab, "Content Based Image Clustering Technique Using Statistical Features and Genetic Algorithm," *Engineering, Technology & Applied Science Research*, vol. 9, no. 2, pp. 3892–3895, Apr. 2019, https://doi.org/10.48084/etasr.2497.

[16] C. Benedek and T. Sziranyi, "Change Detection in Optical Aerial Images by a Multilayer Conditional Mixed Markov Model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 10, pp. 3416–3430, Jul. 2009, https://doi.org/10.1109/TGRS.2009.2022633.

[17] B. L. Saux, R. C. Daudt, A. Boulch, and Y. Gousseau, "OSCD - Onera Satellite Change Detection." IEEE, Oct. 09, 2019, Accessed: Aug. 21, 2022. [Online]. Available: https://ieee-dataport.org/open-access/oscd-onera-satellite-change-detection.

[18] A. A. Nielsen, "The Regularized Iteratively Reweighted MAD Method for Change Detection in Multi- and Hyperspectral Data," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 463–478, Oct. 2007, https://doi.org/10.1109/TIP.2006.888195.

[19] C. Wu, B. Du, and L. Zhang, "Slow Feature Analysis for Change Detection in Multispectral Imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 5, pp. 2858–2874, Feb. 2014, https://doi.org/10.1109/TGRS.2013.2266673.

[20] T. Celik, "Unsupervised Change Detection in Satellite Images Using Principal Component Analysis and k-Means Clustering," *IEEE Geoscience and Remote Sensing Letters*, vol. 6, no. 4, pp. 772–776, Jul. 2009, https://doi.org/10.1109/LGRS.2009.2025059.

[21] W. A. Malila, "Change Vector Analysis: An Approach for Detecting Forest Changes with Landsat," in Symposium on Machine Processing of Remotely Sensed Data and Soid Information Systems and Remote Sensing and Soil Survey Proceedings, Jun. 1980, pp. 326–335.

[22] K. L. de Jong and A. Sergeevna Bosman, "Unsupervised Change Detection in Satellite Images Using Convolutional Neural Networks," in *2019 International Joint Conference on Neural Networks (IJCNN)*, Budapest, Hungary, Jul. 2019, pp. 1–8, https://doi.org/10.1109/IJCNN.2019.8851762.