A CONVOLUTIONAL NEURAL NETWORK FOR FLOOD MAPPING USING SENTINEL-1 AND SRTM DEM DATA: CASE STUDY IN POLDOKHTAR-IRAN

B. Hosseiny [1,] *, N.Ghasemian [1], J.Amini [1]

[1] School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran, Iran - (ben.hosseiny, n.ghasemian, jamini) @ut.ac.ir

Commission III, WG III/7

KEY WORDS: Change detection, Deep learning, Classification, Satellite images, Synthetic aperture radar

ABSTRACT:

Flood contributes a key role in devastating natural and man-made areas. Floods usually are occurred when there is a considerable number of clouds in the sky making optic data useless. Synthetic aperture radar (SAR) images can be a valuable data source in earth observation tasks. The most important characteristic of the radar image is its ability to penetrate the cloud and dust. Therefore, monitoring earth in cloudy or rainy weather can be available by this kind of dataset. In the last few years by improving machine learning methods and development of convolutional neural networks in remote sensing applications we are facing with extremely high improvement in classification tasks. In this paper, we use dual-polarized VV and VH backscatter values of Sentinel-1 and Shuttle Radar Topography Mission (SRTM) digital elevation model (DEM) dataset in a proposed convolutional neural network to generate a land cover map of a flooded area before and after happening. Obtained classification results vary between 93.3% to 98.5% for different training sizes. By comparing the generated classified maps, flooded areas of each class can be extracted.

1. INTRODUCTION

Flood contributes a key role in devastating natural and man-made areas. In March 2019, a destructive inundation occurred in Iran which put down the demise of hundreds of people and extinction of properties. Flood mapping using remote sensing techniques has the merit of helping authorities to have a thorough overview of working out the amount of damage, and it alleviates the emergency procedures (Benoudjit and Guida, 2019; Domeneghetti et al., 2019).

Floods usually are occurred when there is a considerable number of clouds in the sky making optic data useless. So, Flood mapping using SAR images can be a necessity. One way to map flood is by using Change detection approaches when two-time data of a region are available (Zhao et al., 2019). One of conventional change detection approaches is post-classification comparison technique where we compare the classification results of the data at two different times. SAR image classification can be a challenging task due to its complicated backscattering mechanism and the presence of speckle noise. Conventional classification algorithms like SVM and Neural Network cannot handle the disturbances and distortions existence in the SAR dataset and we need an integrated classification system capable of doing feature extraction and partitioning simultaneously. Convolutional Neural Networks (CNN) present an automatic feature extractor connected to a fully connected layer that partition the feature space after removing the uncertainties inherent in the input data. CNN is popular in remote sensing image classification because it alleviates the problem of big data analysis that has been always an issue in satellite image processing.

Change detection techniques can be categorized to supervised and unsupervised (Zhao et al., 2019). Unsupervised techniques create a difference or log-ratio map and need further analysis to assign a threshold to change and no change areas. Supervised methods are more dependent on the availability of pure training samples. Recently, Deep Neural Network has acquired a great interest in change detection related tasks as they can be applied in both a supervised or unsupervised manner. Gong et al., 2015 applied stacked Restricted Boltzmann Machines (RBM) for change detection of Synthetic Aperture Radar (SAR) images using Fuzzy C-Means (FCM) clustering as the pre-training stage. Also, in another research, the sparse autoencoder (SAE) and Convolutional Neural Networks (CNN) were integrated to present a ternary change detection method (Gong et al., 2017). Ternary here refers to classifying pixels to the positive change, negative change, and unchanged. Gao et al., (2016) utilized a kind of deep neural network, PCANet, for change detection of radar images. Firstly, FCM clustering and Gabor wavelets were applied to find pixels with a high probability of being changed and unchanged. Second, remaining pixels further processed using PCA filters to extract relevant features and an SVM classifier to partition the unlabeled pixels. Liu et al., (2016) designed a CNN called Symmetric Coupled Convolutional Network (SCCN), for change detection of heterogeneous optic and SAR images. After representation of two input images in suitable feature space using the mentioned network, the difference image was extracted, and change detection was done based on a Euclidean distance measure. Li et al. (2019) used PCANet to extract the prominent training samples for change detection of SAR images. In another research, RBM and tensor-based information were applied for change detection of hyperspectral images (Huang et al., 2019). Gao et al., (2019) made advantage of a kind of CNN, Convolutional-Wavelet, to monitor sea ice change. Mou et al., (2019) designed a recurrent CNN for extracting spatial, spectral, and temporal features of bi-temporal images simultaneously. While few works have focused on addressing flood mapping as a change detection problem with the aid of deep neural networks, we can refer to some recent works in this area. Nogueira et al., (2018) made advantage of the architectural diversity of dilated ConvNets and deconvolutional ConvNets to feed the resulted feature maps into an SVM classifier for mapping vulnerable areas to flood. Y. Li et al., 2019 proposed a temporal-ensembling CNN and a powerful self-learning framework to map flooded area using intensity and interferometric features of Terra-SAR-X data. Also, stacked modules of CNN and one Recurrent Neural Network (RNN) were utilized to segment flooded areas (Rahnemoonfar et al., 2018). Kang et al., (2018) embedded eight convolutional, two deconvolutional, and fusing layers in their proposed CNN. They integrated deep (global) and shallow (local) features via fusion layers to map flooded areas using GaoFen-3 SAR images. In another work, areas destructed by the tsunami were detected using an autoencoder by imposing the idea that for desired changes the reconstruction loss of the deep network is relatively high compared to trivial changes (Sublime and Kalinicheva, 2019). All of the studies mentioned above have one limitation in common, and it is a lack of training data. As far as supervised change detection using deep learning is concerned, training data plays an important role. Because of the lack of training data related to flood pixels, it is more straightforward to build the classification model based on unchanged pixels and after that apply this trained model to classify both images at two different times sequentially or in parallel. So, the most contribution of our work is using only one classification model for both images at two different times. This kind of model can reduce the inconsistency between the classification results always being claimed to be a dilemma in post-classification change detection.

In this paper, we investigate the use of a 2D CNN in flood mapping of Pol-e-Dokhtar city, Lorestan, Iran using two –time Sentinel-1 dataset, taken before and after the flood, one at 3rd of March of 2019 and another at 2rd of April of the same year. Also, we used SRTM DEM data to increase the number of our input features. The Sentinel-1 dataset is freely available from the Copernicus Open Access Hub and SRTM DEM data are also available from USGS site at no charge.

The next sections of this paper are organized as follows. In Sections 2 and 3, the theoretical background of the convolutional neural networks and our proposed architecture will be presented, respectively. Section 4 belongs to experimental results based on the proposed network. In this section, we also compare the CNN classification result with a 1D CNN and a multi-layer perceptron. Finally, our conclusions will be presented in section 5.

2. THEORETICAL BACKGROUND

2.1 Convolutional neural networks (CNN)

Convolutional neural networks (CNN) can be considered as one kind of the neural networks applied on grid-like topology data such as 2D satellite images. The name “convolutional” comes from their architecture using one kind of linear mathematical operation called convolution. These kinds of neural networks use convolution rather than matrix multiplication used in ordinary neural networks.

Convolutional neural networks can handle distortions and disturbances using three concepts including, sparse connectivity, parameter sharing, and equivariant representations.

2.1.1 Pre-Processing: Before importing the input data to the network, it is essential to make the data zero-centered and normalized. Zero centering is accomplished by subtracting the mean of the whole data (both train and test) from each data point and normalization is achieved by dividing the zero-centered data to the variance of each dimension. We can show the zero centering and normalization equations as follows (Khan et al., 2018):

$$x' = x - \hat{x} \qquad \hat{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{1}$$

$$x'' = \frac{x'}{\sqrt{\frac{\sum_{i=1}^{N}(x_i-\hat{x})^2}{N-1}}} \tag{2}$$

In equation (1), x' is the zero-centered input feature vector, $\hat{x}$ is the mean of the feature vector x. x'' is the normalized feature vector in equation (2). The input to the CNN network is x''.

2.1.2 Convolutional layer: Generally, convolution can be defined as an operation of two functions with real-valued arguments. In equation 3, the asterisk indicates the convolution operation.

$$s(t) = (x * W)(t) = \int x(a)W(t-a)da \tag{3}$$

The first argument, the function x, is referred to as the input and the second, W, is called the kernel in a convolutional neural network.

2.1.3 Non-linearity: The output of a convolutional or fully connected layer is fed into a non-linear or piece-wise linear function. This allows the network to learn non-linear mappings. If we eliminate this non-linearity, only modelling linear functions will be possible. Non-linearity also controls the degree of response of the neuron to a particular input. Non-linearity must be differentiable according to the backpropagation learning rule. The nonlinearity is used both after a convolutional layer and in a fully connected layer. In this work, we applied the Rectified Linear Unit (ReLU) activation function as the non-linearity of our proposed network.

2.1.4 Pooling layer: Pooling operation is applied to the output of the non-linearity layer. It represents a statistical summary of the data and removes the distortions and disturbances in the primary feature map and also lessens the computational burden of the data. For example, max-pooling chooses the maximum unit in a rectangular neighborhood. Some other pooling operations include average pooling, L2 norm in a rectangular neighborhood or weighted average in a rectangular neighborhood with weights based on distance. Pooling layers ensure us that the learned function is invariant to the small changes in the input data and thus improve the generalization ability of the network. Figure 1 demonstrates an example of the max pooling operation on the input units. Pooling kernel size and stride are considered as 3 and 2 respectively. For the last unit, a pooling size of 1 was applied in order to consider the information of all the units.

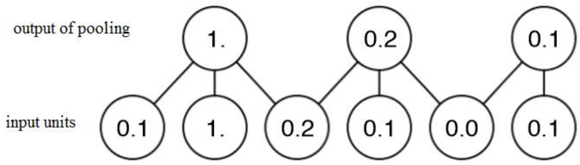

Figure 1: max pooling operation on the input units (Goodfellow et al, 2016).

2.1.5 Fully connected layer: Fully connected layers are usually placed at the end of a CNN. These layers are the same as the weight layers of a Multi-Layer Perceptron and can be considered as a convolutional layer with a filter size of 1×1. The input data are multiplied by a weight matrix and are added to a bias vector and after passing through the activation function f the output vector is obtained. In Equation (4), y is the output the network which can be a classification label and f is the activation function. The W matrix and b vector are the trainable parameters of the fully connected layer and are referred to as the weight and bias vector.

$$y = f(W^T x + b) \tag{4}$$

3. PROPOSED METHOD

Figure 2 shows the steps of the proposed method. In this paper, post-classification change detection method is implemented to extract flooded areas. The mentioned method contains three main steps: 1) Data pre-processing and preparation, 2) Classification, 3) Change detection. In the first step, radiometric and geometric calibrations are implemented on the acquired satellite images to obtain sigma naught values of each pixel in the azimuth and ground range directions. Each calibrated data is stacked with the SRTM DEM and then, in order to feature scaling all the feature layers are normalized. In the second step, for each dataset, the classification model is trained and the classified map is generated, separately. In the last step, the change map can be extracted by comparing the classified maps.

Our proposed classifier network in this work contains two convolutional layers connected to a multi-layer perceptron with two fully-connected hidden layers as a classifier. Each convolutional layer contains two-dimensional 3 by 3 filters that convolve with input patches. In order to increase nonlinearity of the model, rectified linear unit (ReLU) function is used as activation function in every layer of the network. Drop-out and batch-normalization layers are considered in order to increase the network's stability and generalization ability. Layers are ordered as:

- Input patch with the size of 5×5 equivalent to $50^m \times 50^m$ on the ground
- First convolutional layer:
  50 filters with 3×3 kernel size,
  ReLU activation function,
  10% dropouts and batch normalization
- Second convolutional layer:
  50 filters with 3×3 kernel size,
  ReLU activation function,
  10% dropouts and batch normalization
- Flattening convolutional layers output in a one-dimensional vector
- First fully connected layer:
  100 dense neurons
  ReLU activation function
  50% dropouts and batch normalization
- Second fully connected layer:
  100 dense neurons
  ReLU activation function
  50% dropouts and batch normalization
- Output layer:
  Neurons equal to the number of labels
  Softmax activation function

4. RESULTS AND DISCUSSIONS

In this section, we examine our proposed network on a case study from a flooded area in Poldokhtar, Iran. Figure 3 demonstrates the google earth view, digital elevation model, and VV polarized decibel image of Sentinel-1 data before and after occurring flood of our study area. This area contains the urban area, agricultural lands, bare soils, vegetated area and river, and wetlands. Therefore, our selected labels are Water, Vegetation, Bare soils, and Urban area. Our available data are dual-polarized ground range detected (GRD) Sentinel-1 data (VV-VH) and SRTM

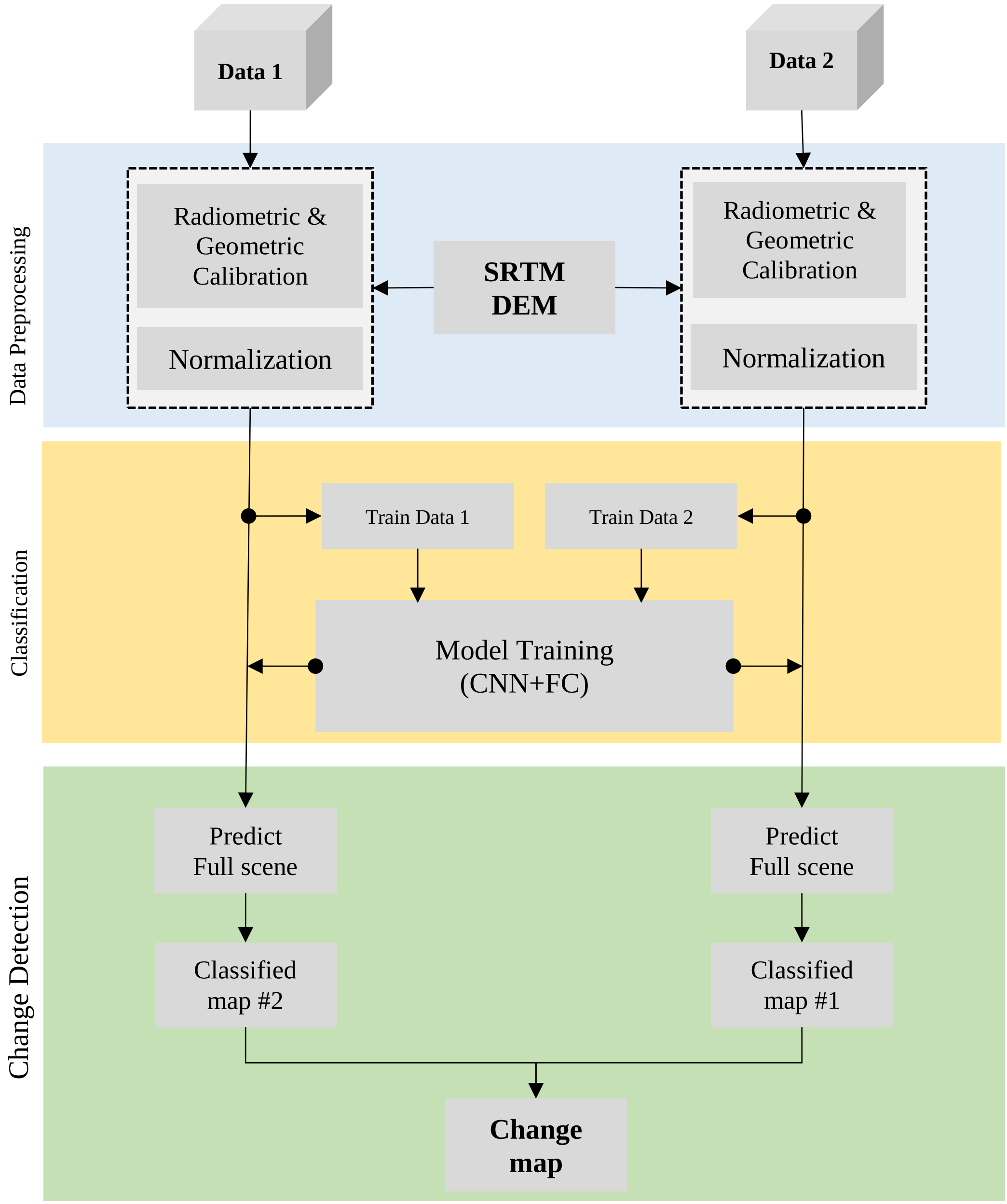

Figure 2. Flowchart of the proposed method

digital elevation model (DEM). Table 1 summarizes the main specifications of the used data. We use these three layers of the data (VV, VH, and DEM) as input feature cube to the proposed network. The acquired results from our CNN-based network are compared with two other simpler networks: the first network is a convolutional network with a similar architecture to our proposed network but with one-dimensional filters, and the second is a simple multi-layer perceptron network with the completely similar architecture of our proposed network's fully-connected section.

Satellite Imagery	Acquisition date	Frequency Band (GHz)	Polarization	Resolution (Range×Az.) (m)
Sentinel-1	2019/03/03	C (5.4)	VV/VH	20×22
Sentinel-1	2019/04/02	C (5.4)	VV/VH	20×22

DEM	Ground Resolution (m)	Vertical Precision (m)
SRTM	30×30	16

Table 1. Used data specifications

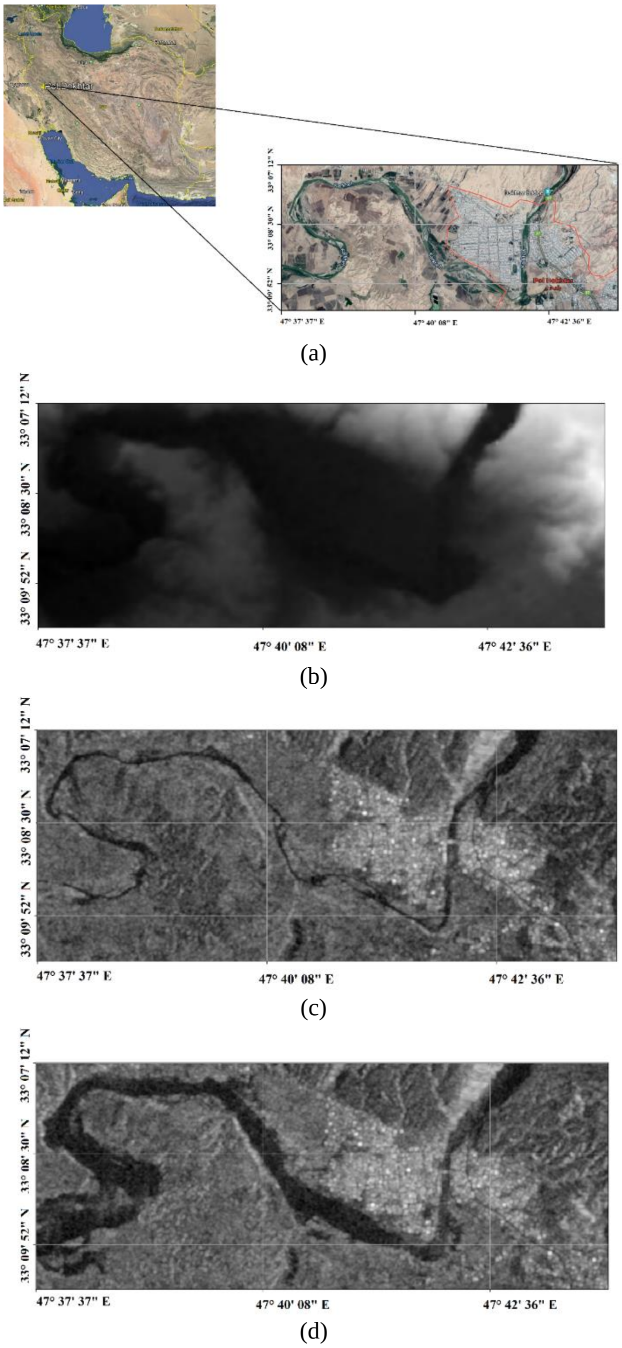

Figure 3. Study area in Poldokhtar, Iran (a) Google Earth view (b) SRTM DEM (c) Sentinel-1-VV, before occurring flood (d) Sentinel-1-VV, after occurring flood

We examined the performance of our proposed network with different training sizes. Obtained overall accuracy (OA) of the test data with different training sizes for three different artificial neural network architectures are gathered in table 2. The second column of this table belongs to our proposed network and third and fourth columns belong to results obtained from 1-D CNN and MLP. All three networks were trained with the learning rate of 0.00003 in 400 epochs. By changing the training size, test overall accuracy of the 1D and 2D CNN-based classifiers vary between 91.03% ~ 93.57% and 94.90% ~ 98.5%, respectively. However, test overall accuracies obtained from the simple MLP network varies between 69.4% ~ 83.07%. By comparing these results, we can notice that convolutional layers improve the classification results more than 10% in every case. In addition to that, it is noticeable that CNN-based models are robust to training data size.

Train Samples	Test OA (%)		
	2D CNN+MLP	1D CNN+MLP	MLP
100	94.90	91.03	69.4
200	97.23	92.31	76.88
300	98.5	93.57	83.07

Table 2. Obtained classification results for different training samples

Tables 3-5 indicate obtained classification results for every class, based on simple MLP, 1-D CNN-FC, and 2-D CNN-FC classifiers, respectively. As table 3 shows, classification results of the simple MLP contains poor results in some classes such as vegetation. Thanks to deep and nonlinear feature extraction of convolutional layers, acquired results of the CNN-based classifiers (tables 4, 5) are superior to the simple MLP model. Also, 2-D CNN-FC classifier due to deep spatial feature extraction in each input layer has obtained the best classification result compared to other mentioned methods.

Train Samples	100	200	300
	Test OA (%)		
Water	88.4	88.2	87.9
Vegetation	19.6	56.3	65.6
Bare soil	89.3	84.4	90.6
Urban	82.3	86.9	87.4

Table 3. Obtained classification results of every class for different training sizes for the simple MLP

Train Samples	100	200	300
	Test OA (%)		
Water	99.1	99.1	99.3
Vegetation	87.4	92.4	96.2
Bare soil	89.6	91.5	91.7
Urban	92.5	92.3	93.1

Table 4. Obtained classification results of every class for different training sizes for the proposed 1-D convolutional network

Train Samples	100	200	300
	Test OA (%)		
Water	99.1	99.1	99.7
Vegetation	93.7	98.1	99.6
Bare soil	96.1	96.9	98.9
Urban	94.2	96.8	97.5

Table 5. Obtained classification results of every class for different training sizes for the proposed 2-D convolutional network

After evaluating the proposed network's performance, All the ground truth data from the first and second dataset are fed to the classifier network to train each model separately. Table 6 shows the classification results of the second dataset based on the proposed 2-D CNN-FC method. Figure 4 (a) shows the ground truth map of the study area, which is collected with the help of

Google earth's high-resolution visible imagery. Figure 4 (b, c) show the generated land cover map of the study area before and after occurring the flood, respectively.

Train Samples	100	200	300
	Test OA (%)		
Water	98.2	98.2	99.3
Vegetation	92.4	96.3	98.7
Bare soil	95	98.3	97
Urban	88.3	93.6	97.5
OA	93.3	96.29	97.62

Table 6. Classification results of the second dataset for the proposed 2-D convolutional network

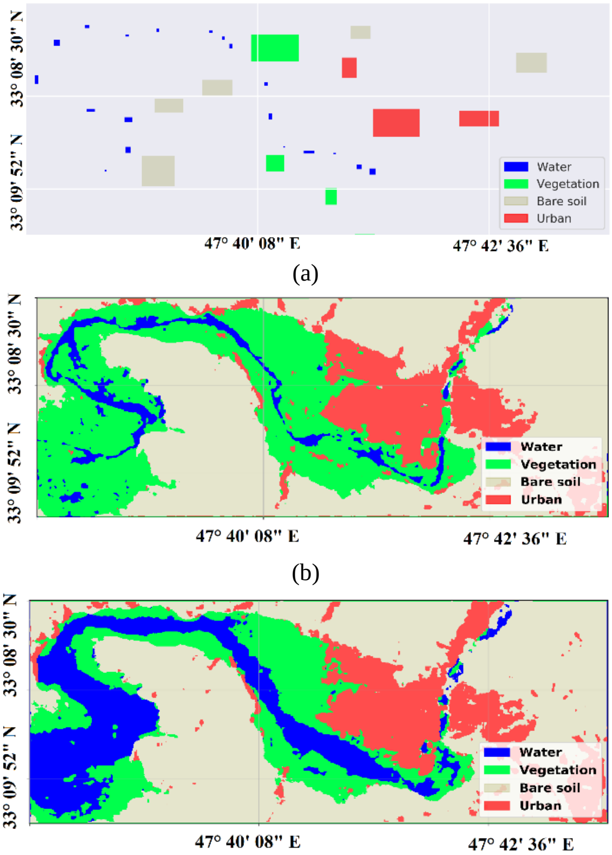

Figure 4. (a) Ground truth map and (b) Classified map of the study area before the flood (c) Classified map of the study area after flood

By comparing the two classified maps we can estimate the water inundated areas. Figure 5 shows water inundated areas after happening flood in our study area, where white areas indicate the water-covered area after occurring the flood. In order to reduce the influence of the classification error, water inundated map is used as a mask and only the flooded area is studied. The final change map is shown in figure 6. Also, Table 7 indicates areas of changed classes to water class individually in squared meters, by considering that every pixel-size is $10^m \times 10^m$. Based on this table, about 2591100 (m^2) of vegetated areas are covered by water. Also, 19500 (m^2) of bare lands and 29400 (m^2) of urban areas are covered by water after occurring flood.

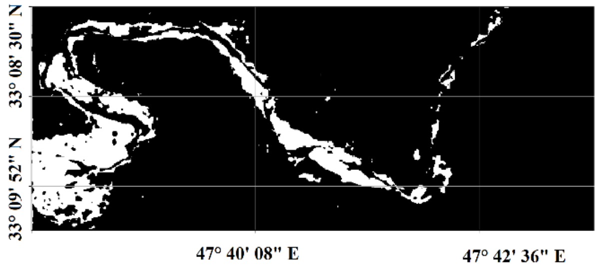

Figure 5. Water inundated map

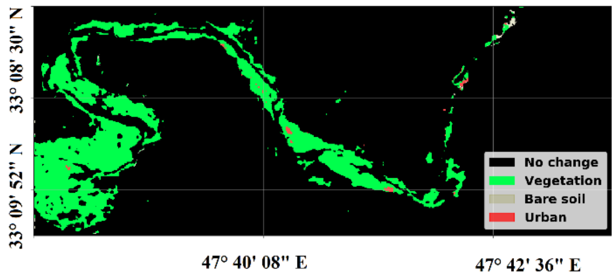

Figure 6. Change map

Class labels	Vegetation	Bare soil	Urban
Changed area (m^2)	2591100	19500	29400

Table 7. Water covered areas after occurring flood divided by every class

5. CONCLUSION

This paper presented a classification method based on two-dimensional convolutional neural networks in order to classify and extract flooded areas from Sentinel-1 satellite radar imagery and SRTM digital elevation model. The proposed network was applied for land cover mapping of a study area, before and after occurring flood. The results of the proposed network compared with results of one dimensional CNN and a simple MLP network indicate more than 10% of improvement in classification accuracy. After classifying before and after flood images, water inundated areas were extracted and divided by each class.

REFERENCE

Benoudjit, A. and Guida, R., 2019. A Novel Fully Automated Mapping of the Flood Extent on SAR Images Using a Supervised Classifier. *Remote Sensing*, *11*(7), p.779.

Domeneghetti, A., Schumann, G.J.P. and Tarpanelli, A., 2019. Preface: Remote Sensing for Flood Mapping and Monitoring of Flood Dynamics.

Gao, F., Dong, J., Li, B. and Xu, Q., 2016. Automatic change detection in synthetic aperture radar images based on PCANet. *IEEE Geoscience and Remote Sensing Letters*, *13*(12), pp.1792-1796.

Gong, M., Yang, H. and Zhang, P., 2017. Feature learning and change feature classification based on deep learning for ternary change detection in SAR images. *ISPRS Journal of Photogrammetry and Remote Sensing*, *129*, pp.212-225.

Goodfellow, I., Bengio, Y. and Courville, A., 2016. *Deep learning*. MIT press.

Huang, F., Yu, Y. and Feng, T., 2019. Hyperspectral remote sensing image change detection based on tensor and deep learning. *Journal of Visual Communication and Image Representation*, *58*, pp.233-244.

Kang, W., Xiang, Y., Wang, F., Wan, L. and You, H., 2018. Flood Detection in Gaofen-3 SAR Images via Fully Convolutional Networks. *Sensors*, *18*(9), p.2915.

Khan, S., Rahmani, H., Shah, S.A.A. and Bennamoun, M., 2018. A guide to convolutional neural networks for computer vision. *Synthesis Lectures on Computer Vision*, *8*(1), pp.1-207.

Liu, J., Gong, M., Qin, K. and Zhang, P., 2016. 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), pp.545-559.

Li, M., Li, M., Zhang, P., Wu, Y., Song, W. and An, L., 2019. SAR Image Change Detection Using PCANet Guided by Saliency Detection. *IEEE Geoscience and Remote Sensing Letters*, *16*(3), pp.402-406.

Li, Y., Martinis, S. and Wieland, M., 2019. Urban flood mapping with an active self-learning convolutional neural network based on TerraSAR-X intensity and interferometric coherence. *ISPRS Journal of Photogrammetry and Remote Sensing*, *152*, pp.178-191.

Nogueira, K., Fadel, S.G., Dourado, Í.C., Werneck, R.D.O., Muñoz, J.A., Penatti, O.A., Calumby, R.T., Li, L.T., dos Santos, J.A. and Torres, R.D.S., 2018. Exploiting ConvNet diversity for flooding identification. *IEEE Geoscience and Remote Sensing Letters*, *15*(9), pp.1446-1450.

Rahnemoonfar, M., Murphy, R., Miquel, M.V., Dobbs, D. and Adams, A., 2018, July. Flooded area detection from uav images based on densely connected recurrent neural networks. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium* (pp. 1788-1791). IEEE.

Zhao, M., Ling, Q. and Li, F., 2019. An Iterative Feedback-Based Change Detection Algorithm for Flood Mapping in SAR Images. *IEEE Geoscience and Remote Sensing Letters*, *16*(2), pp.231-235.