FLOODED AREA DETECTION BY GABOR FILTERING BASED ON CONVOLUTIONAL NEURAL NETWORK

Andrada Livia CIRNEANU[1], Dan POPESCU[2]

In this paper, we analyze different areas affected by floods using images taken with a UAV (Unmanned Aerial Vehicle). The proposed methodology considers the development of a neural network-based algorithm defined with a custom convolutional layer filtered by a bank of Gabor filters and implemented with a backpropagation approach for the training phase. The neural network is trained to recognize flood patches from UAV images affected by flood. Thus, two classes are defined: flood and non_flood. For each patch selected for the training phase, a feature vector is built using the co-occurrence matrix information generated from the output of the convolutional layer and pixels value information extracted from HSV representation. For experimental results, the neural network was trained over a series of patches from both classes. After achieving the ability to distinguish patches from each class, it was tested on over 50 UAV images offering good results regarding the flood detection accuracy.

Keywords: texture, co-occurrence matrix, convolution, neural network, image processing, unmanned aerial vehicle

1. Introduction

Over the last years, the requirement for extracting/ generating various information about a region/territory has increased, as a basis for the emergency and recovery planning or the humanitarian strategies and procedures mandatory for the management of natural disasters. As a result, the detection of areas affected by natural disasters became an important challenge for every image processing application in this field. The solutions provided by the researchers should answer to some specific criteria like real time response or high accuracy detection of these areas.

The technology has also advanced far enough to provide researchers diverse methods to promptly obtain specific information about the investigated area. One of these is represented by the UAV (unmanned aerial vehicle) technology. By producing spatial analyzes, this technology provides an interpretation and understanding of the real-time situation over the areas affected by natural disasters and it accelerates the decisions making process.

[1] PhD Eng., Automatic Control and Industrial Informatics Department, University POLITEHNICA of Bucharest, Romania, e-mail: andrada.cirneanu@gmail.com

[2] Prof., Automatic Control and Industrial Informatics Department, University POLITEHNICA of Bucharest, Romania, e-mail: dan_popescu_2002@yahoo.com

There are many recent solutions proposed for the detection of specific regions from areas affected by different types of natural disaster. For example, in [1] it is presented an automated image detection solution that demonstrated high accuracy and reliability in identifying areas affected by flood from multiple test locations. The geomorphological instrument of the flood (GFA) enables rapid and efficient mapping of floods by the aid of a linear binary classification based on the recent geomorphic flood index (GFI).

The authors from [2] propose a method that relies on the merger of UAV photogrammetry, digital elevation model (DEM) and the orthoimage (an ortho-rectified image). Initially, the test features in the orthoimage are calculated. Then DEM is used to obtain a flood depth pattern and a field resistance index. Finally, all information is merged with the spelling image to form a multi-lane dataset that is used for object-based segmentation and supervised classification of river formations. The experimental results obtained have demonstrated that fusion of 2D and 3D data with derived qualitative layers has significantly improved the detection process of river flow.

To analyze the content of an image, an image descriptor is required to capture the particularities of each image by computing certain features. Many image classification techniques implemented until the present time are based on Gabor filtration because it offers the maximum resolution in both space and frequency domains with minimal uncertainty. Thus, in [3], the authors propose a novel spectral-spatial classification approach based on Gabor filter response using the first three principal components of the hyperspectral image. A set of features is composed from Gabor features and spectral features, which are supplied to a deep network.

Another significant utility of the Gabor filter manifests when the scale and the direction of a texture are changed [4]. To extract the texture, the frequency components are transformed into time domain and the curvature factor is introduced based on the traditional Gabor model. The experimental results prove that this solution is able to extract the polarized image texture.

For real time processing applications, a method periodic/quasi-periodic de-noising in order to remove unwanted periodic patterns from Gabor-transform corrupted images is considered in [5]. The algorithm saves the low-frequency region, increases the contrast of any corrupted image spectrum using Gabor transform and applies an adaptive exponential thresholding procedure.

The images obtained from remote systems are also the object of various Gabor filter implementations. For this purpose, in [6] it is presented a novel descriptor based on magnitude/phase combination using the multi-oriented phase congruency and magnitude information of log-Gabor filters. The method was evaluated with three different datasets and the experimental results confirmed that the algorithm superior to the current state-of-the-art local feature descriptors.

On the other hand, the development of new versions for Gabor filter is still a research interest. Thus, the authors in [7] introduce a novel approach for computing a 2D complex Gabor filter bank. The runtime of the Gabor filter bank is reduced by reusing intermediate results obtained for a specific orientation. The results proved to have a faster run time in comparison with the state-of-the-art methods.

As an alternative, the solutions proposed in [8] and [9] are based on artificial intelligence. The solution [8] consists in the integration of densely connected CNN (convolutional neural network) and RNN (recurrent neural network), which allows the segmentation of significant semantic object boundaries. The proposed network is applied to aerial images of several flooded areas in Houston, TX. On the other hand, in case of managing emergencies crisis generated by floods, a deep learning approach using the SegNet network architecture is considered in [9] for more accurate information retrieval. The experiment results proved to offer a high performance in the detection of areas of interest in case of image obtained with a UAV along a riverbank area.

By processing two sets of data generated from the RBG color space, the authors in [10] manage to map and monitor floods in a particular area by developing an application that analyzes and compares spectral responses of urban types, shadows and thick water of floods.

Neural networks have proven their utility in image analysis and gained more popularity in this domain due to several advantages like the efficiency in recognition and classification of patterns or a lower computational cost and a lower number of operations obtained through the reduction of the number of parameters. However, the biggest challenge is the processing of very large-size images (for example, the algorithm presented in this paper uses images of dimension 6000×4000 pixels). Therefore, the methodology proposed in this paper for the evaluation of areas affected by floods considers the integration of a classic backpropagation neural network with a custom convolutional layer based on a bank of Gabor filters. The features used for both training and testing phases of the neural network are the mean contrast and mean dissimilarity co-occurrence matrix properties computed based on the output of the proposed convolutional layer and the mean intensity values of H and V component of HSV color model.

2. Methodology

The solution proposed in this paper for flooded areas detection is a neural network implemented with backpropagation for the training phase. The neural network input is supplied by a patch-processing algorithm that is based on a custom convolutional layer in conjunction with a bank of Gabor filters. The approach from this paper is sustained by the simplicity in data classification

offered by neural networks with a single hidden layer (based on activation functions and weights generated in the training phase) and the flexibility of CNN that can extract relevant features at a low computational cost.

A CNN is a special category of neural networks with applicability ranging from object classification and detection to face recognition or semantic segmentation tasks. The CNN architecture is designed to fit any number of image representations and any number or dimensions of the convolutional filters. From a mathematical point of view, the output O of a CNN for an input x can be described as in (1) [11]:

$$O(x) = f_n(f_{n-1}(f_{n-2}(\dots f_2(f_1(x, w_1); w_2) \dots; w_{n-2}); w_{n-1}); w_n) \quad (1)$$

where each f_i, $i=1,\dots,n$ represent a computational block known as layer. A layer f receives inputs of type x(certain features extracted from images specific to the classification requirement) and depending on the logic implemented inside each layer, various transformations are applied over the input x resulting in a new entry for the next layer. Each defined layer can have a different set of parameters (features) and a different set of weights w can be computed and adjusted during the training phase for each layer.

Although there is no technique for the best architecture or connection between layers, all defined parameters for each layer influence directly the performance of the network and eventually, its accuracy [12]. Hence, the proposed algorithm consists in two phases: a training phase in which patches from both defined classes (flood/non_flood) are processed and a testing phase in which UAV images are processed patch by patch for flooded areas detection. Both phases are built on top of the patch-processing algorithm presented in Fig.1 that has the role of generating features for the neural network training and testing phases.

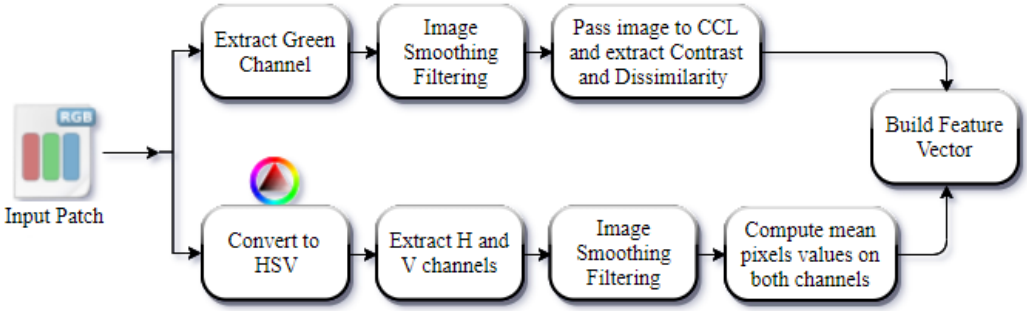

Fig.1. General architecture of patch-processing algorithm

The proposed patch-processing algorithm has two parts:

- Processing of the patch Green channel from RGB color model using a smoothing filtration and passing the result to the custom convolutional layer(CCL) after which the co-occurrence matrix properties are extracted.
- Processing of the patch H and V channels from HSV color model using the same smoothing filtration and extracting features related to the mean intensity values.

With the purpose of removing high frequency content, a smoothing filtration is implemented by convolving the patch with a low-pass normalized kernel K_S, which simply computes the average of all pixels, positioned inside the kernel area and replaces the central element with the obtained value. The kernel used in this paper is presented in (2) having a 9×9 dimension. In the end, the output filtered patch has the same dimensions as the input patch [13], [14].

$$K_S = \frac{1}{9 \cdot 9}\begin{bmatrix} 1 \cdots 1 \\ \cdots \\ 1 \cdots 1 \end{bmatrix} \tag{2}$$

Following this, the low-pass normalized kernel is applied over the patch Green channel version and the output is passed to the CCL. The operations flow during patch transformations within the CCL is presented in Fig. 2 where five elements can be observed:

- Smoothing filtration output that represents the input element for the CCL.
- Convolution operation output using a bank of seven Gabor filters.
- ReLU activation function output over each feature map returned by the convolutional operation.
- Max Pooling operation output generated over each map returned by the ReLU activation function.
- Co-occurrence matrix features (contrast and dissimilarity) generated using a mean operation over all maps returned by the Max Pooling operation.

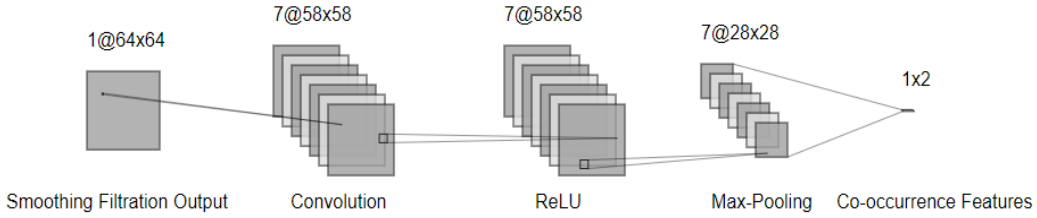

Fig. 2. Operations flow inside the custom convolutional layer

The convolution operation proposed in this paper implies the use of a bank of seven Gabor filters with different orientations taking into consideration that a Gabor filter detects a pattern on a specific orientation and the process of acquirement UAV images is definitely influenced by weather conditions. A two-

dimensional Gabor filter is represented by a Gaussian function modulated by a sine wave oriented at a certain angle and it is defined as in (3) where σ_x and $\sigma_y$represent the standard deviations of the Gaussian function on the x and y directions, ω is the frequency of the sinusoidal wave and θ indicates the filter orientation [15].

$$G(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \cdot \exp\left(-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2}\right) \cdot \exp[j \cdot \omega^2(x+y)] \tag{3}$$

In the proposed methodology, the bank of Gabor filters F_{Gs} is defined as a collection of seven Gabor filters with dimensions 7×7 and orientations θ_k, $k=1,2,\dots 7$ in the range$\{0°, 30°, 60°, \dots, 180°\}$.The standard deviation σ_x and σ_y were set to 1 and frequency ω to 0.1 for all filters [16].

$$F_{Gs} = \{G_1(\theta_1), G_2(\theta_2), \dots, G_7(\theta_7)\} \tag{4}$$

The formula for computing the convolution operation between pixels positioned in the line I and column j in an image I and a filter H with dimension $w \times w$ is presented in (5), where k represents the first and last lines and columns from the analyzed image that are ignored during this operation. The value of k is equal to the largest integer that does not exceed the w dimension of the filter H divided by 2. Given the Gabor filters dimension (7×7), the value of k is set to 3 in the proposed algorithm. Consequently, since the input patch has a fixed width and height equal to 64, no stride nor padding were used during the convolution and seven Gabor filters were defined, seven features maps of dimensions 58×58 are generated by the convolution operation.

$$Conv(i,j) = \sum_{u=1}^{w}\sum_{v=1}^{w}(H[u,v] \cdot I[i+u-k, j+v-k]) \tag{5}$$

Afterward, the ReLU activation function is applied over each feature map generated by the convolution operation and its role is to ensure the nonlinear behavior of the algorithm. The ReLU activation function is defined in (6) where x represents an element from one of the seven feature maps. Thus, the ReLU activation function output is represented by another set of seven feature maps with the same dimensions [17].

$$f_{ReLU}(x) = \max(0, x) \tag{6}$$

Further, the pooling operation is applied over the seven feature maps generated by ReLU. Within this operation, a size reduction of the feature maps is achieved, which improves the performance of the algorithm. Two parameters are important for this operation: the pooling mask size P_{Ms} and the stride P_{St} on which the pooling operation output dimension depends. Thus, in the proposed methodology, both parameters values were set to 2 meaning that the feature maps

dimensions generated by the pooling operation P_{Md} is equal to 28×28 by following the formula in (7):

$$P_{Md}=\frac{F_{Ms}-P_{Ms}}{P_{St}} \tag{7}$$

where F_{Ms} is the input feature map size. A simple representation of the pooling operation over a small matrix is presented in Fig. 3.

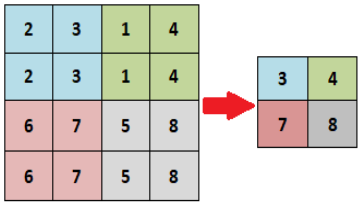

Fig. 3. Pooling operation exemplification

Finally, using the seven feature maps resulted from pooling operation, the algorithm calculates the co-occurrence matrix P for each feature map and same features are extracted. Experimentally, it has been proven that only contrast and dissimilarity properties gave the best results in differentiating the flood class from the non_flood class. Having seven feature maps generated by the Pooling operation, the contrast $\bar{C}$ and dissimilarity $\bar{D}$ are computed as in (8) and (9) where $P(l,i,j)$ represent the element positioned in the line i and the column j in the co-occurrence matrix computed for the l feature map generated by the Pooling operation [18].

$$\bar{C}=\frac{\sum_{l=1}^{7}\sum_{i=1}^{256}\sum_{j=1}^{256}P(l,i,j)\cdot(i-j)^2}{7} \tag{8}$$

$$\bar{D}=\frac{\sum_{l=1}^{7}\sum_{i=1}^{256}\sum_{j=1}^{256}P(l,i,j)\cdot|i-j|}{7} \tag{9}$$

On the other hand, the second part of the proposed patch-processing algorithm uses the HSV representation of the patch. H and V channels are extracted and filtered with the same low-pass normalized kernel from (1) and the mean intensity values $\bar{H}$ and $\bar{V}$ are computed over the filtered version of H and V channels.

Having all proposed features established, a feature vector f_V is defined as a one-column vector with 4 elements (10): the mean contrast $\bar{C}$ and the mean dissimilarity $\bar{D}$ values of the co-occurrence matrix computed based on the output of the proposed convolutional layer and the mean intensity values $\bar{H}$, $\bar{V}$ computed based on the proposed patch processing on HSV representation.

$$f_V=[\bar{C};\bar{D};\bar{H};\bar{V}] \tag{10}$$

The proposed methodology uses a neural network with backpropagation to detect flooded areas from UAV images.

The general architecture of the proposed neural network is presented in Fig. 4, where the following components are highlighted:

- The input parameter represented by the f_V feature vector generated by the proposed patch-processing algorithm.
- The set of weights W_k ,$k=1,...,4$ corresponding to each element of f_V.
- The dot product between f_V and W_k.
- The class determination process using above dot product and the sigmoid function.
- The weights adjustments process (highlighted in blue) which it is used only in the training phase of the proposed neural network and it consists in computing the error between the desired output and the predicted output and the sigmoid function gradient of the predicted output.

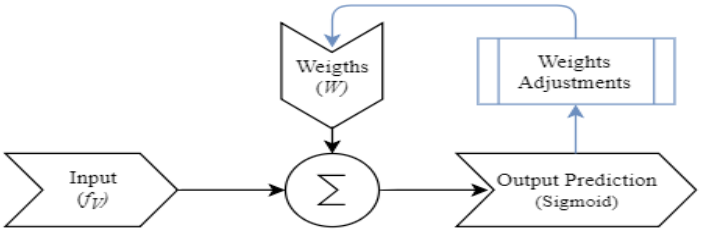

Fig. 4. Neural Network Architecture

During the training phase, to reach a high accuracy for class prediction over the training set, the neural network is trained for a specific number of epochs with the purpose of applying small adjustments to the weights after each epoch. Since too many epochs can lead to over-training, the number of epochs is variable and it depends on a training tolerance of 10^{-5}.

The final step of the proposed methodology is the testing phase in which UAV images are analyzed patch by patch. During this phase, f_V feature vector is computed for each extracted patch and based on set of weights generated during the training phase, the class of the patch NN_{Class}(11) is established based on the neural network output NN_O which varies from *0* to *1*.

$$NN_{Class}(Patch) = \begin{cases} flood, NN_O > 0.5 \\ non\_flood, NN_O \leq 0.5 \end{cases} \quad (11)$$

3. Experimental results

The flooded areas detection algorithm proposed in this paper was developed using Python software and a series of additional image processing libraries: OpenCV used for computer vision and machine learning functions,

NumPy for numerical operations and Scikit-image for various image processing functions. The proposed algorithm was trained and tested on over 50 images with dimension of 6000×4000 pixels.

The images used in this research in both training and testing phases were acquired with a UAV called MUROS which is a surveillance system. This system has a flexible robotic architecture and can acquire big data which can be processed using the proposed developed algorithm with the aim of detecting the areas of interest. As it was described in the methodology section, a set of 200 patches manually extracted from several UAV images were used in the training phase to build the set of weights W of the neural network based on the features vector f_V generated by the proposed patch-processing algorithm. The weights W are required in the testing phase in which the class of a patch extracted from an UAV image is established.

For the training phase, a number of 100 patches were used for the flood class and another 100 for the non_flood class. Fig. 4 and Fig. 5 present a selection of representative patches for each class.

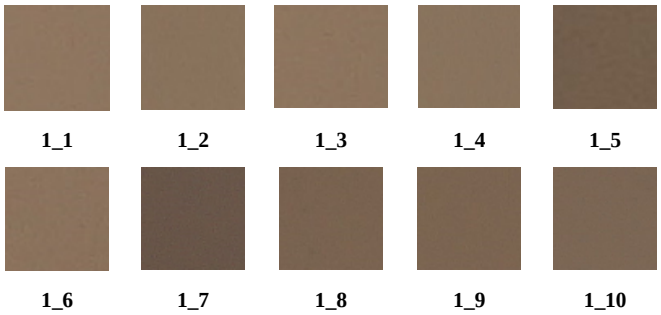

Fig. 4. Representative "flood" patches for training phase

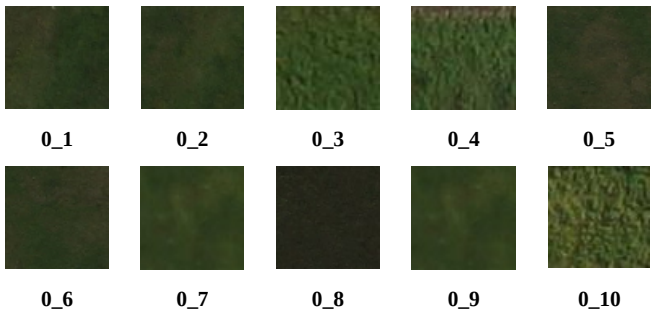

Fig. 5. Representative "non_flood" patches for training phase

In order to present how the features vector f_V was computed for each patch from both classes and since the number of patches used in the training phase is big, in table I the closed interval of values for each feature value ($\overline{\boldsymbol{C}}, \overline{\boldsymbol{D}}, \overline{\boldsymbol{H}}, \overline{\boldsymbol{V}}$) can be found.

Table I

Training data computed for selected patches

f_v	*Min*	*Max*
Flood Class		
$\bar{C}$	0.092280	0.556122
$\bar{D}$	0.094582	0.509316
$\bar{H}$	0.045683	0.058874
$\bar{V}$	0.390339	0.563551
Non_Flood Class		
$\bar{C}$	0.874223	22.875110
$\bar{D}$	0.701197	3.681233
$\bar{H}$	0.087545	0.188275
$\bar{V}$	0.135159	0.313873

At this moment, the training phase of the neural network can be initiated, and it consists in a weights computation process starting with an initial random generation and followed by an adjustment at every iteration until the optimum is reached, meaning the error minimization process between the desired output and the actual output. At the end of the training phase, a set of weights is generated, its values being specific to the training patches (Fig. 5) used in this paper. Over 50 images were tested with the proposed neural network for flooded areas detection. A selection of tested UAV images is presented in Fig. 6 where the detected flooded areas are highlighted with green.

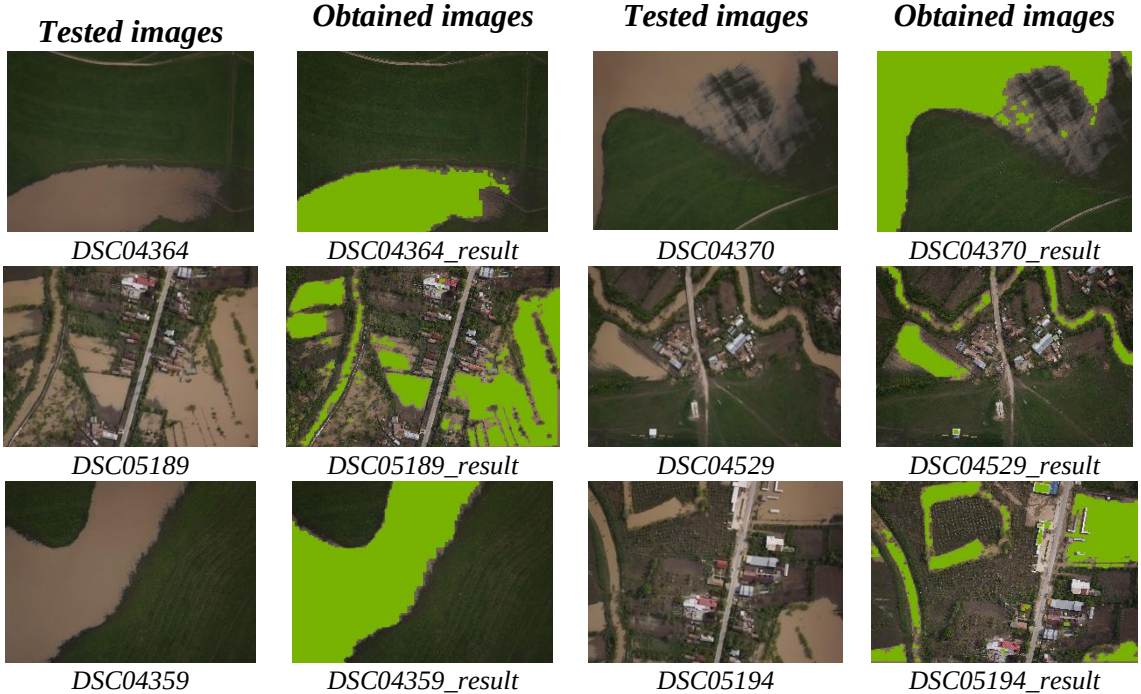

Fig. 6. Selection of tested UAV images

The proposed methodology was applied on complex images that contained different areas (floods, vegetation, houses, roads or land) and proved to offer a very good performance. The correct classification of patches is very important in managing areas affected by natural disasters where proper identification should not fail because people's lives can be endangered. A very encouraging result is that in all tested images the neural network proposed has correctly identified the affected areas obtaining a high rate.

For the presented neural network solution for flood detection, the accuracy is the parameter which provides the performance of the developed methodology. In consequence, the accuracy obtained in this paper has the value of 98%. By comparison with another solution developed for flood detection from UAV images [19], the solution proposed in this paper proved to offer a better level of accuracy. Even though the method from [19] does not use the same data set, the approach based on the integration between CNN and RNN networks is basically similar with the methodology proposed in this paper. Additionally, for a fair comparison, the size of the training data and the computational time for detecting the flood regions from UAV images were not included.

4. Conclusions

In this study, an efficient method for flooded areas detection from UAV images is proposed in which the key element is a backpropagation neural network defined with a custom convolutional layer based on a bank of Gabor filters. The novelty of this method is the fact that the process of acquirement UAV images is definitely influenced by weather conditions and the proposed CNN solution is able to detect and differentiate the floods regions from other regions (vegetation, houses, roads or land) in any orientation change of the textures present in the analyzed images. Despite the fact that the tested images were complex, the experimental results on over 50 UAV images show good results regarding the flood detection accuracy.

R E F E R E N C E S

[1]. *C. Samela, R. Albano, A. Sole, S. Manfreda,* "A GIS tool for cost-effective delineation of flood-prone areas", Computers, Environment and Urban Systems Volume 70, July 2018, Pages 43-52.

[2]. *J. Langhammer, T. Vacková,* "Detection and Mapping of the Geomorphic Effects of Flooding Using UAV Photogrammetry", T. Pure Appl. Geophysics, 2018, 175: 3223. https://doi.org/10.1007/s00024-018-1874-1.

[3]. *X. Kang, C. Li, S. Li, H. Lin,* "Classification of Hyperspectral Images by Gabor Filtering Based Deep Network", IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Volume:11, Issue: 4, April 2018.

[4]. *B. Yuana, B. Xiaa, D. Zhangb,* "Polarization Image Texture Feature Extraction Algorithm Based on CS-LBP Operator", 8th International Congress of Information and Communication Technology (ICICT-2018) Procedia Computer Science 131 (2018) 295–301.

[5]. *D. Chakraborty, M. K. Tarafder, A. Banerjee, S. R. Bhadra Chaudhuri*, "Gabor-based spectral domain automated notch-reject filter for quasi-periodic noise reduction from digital images", Multimed Tools Appl. (2018), https://doi.org/10.1007/s11042-018-6194-z.

[6]. *Z. Fu, Q. Qin, B. Luo, H. Sun, C. Wu*, "HOMPC: A Local Feature Descriptor Based on the Combination of Magnitude and Phase Congruency Information for Multi-Sensor Remote Sensing Images", Remote Sens. 2018, 10(8), 1234, https://doi.org/10.3390/rs10081234.

[7]. *J. Kim, S. Um, D. Min*, "Fast 2D Complex Gabor Filter With Kernel Decomposition", IEEE Transactions on Image Processing, Volume: 27, Issue: 4, April 2018.

[8]. *M. Rahnemoonfar, R. Murphy, M. Vicens, D. Dobbs, A. Adams*, "Flooded Area Detection from UAV Images Based on Densely Connected Recurrent Neural Networks", IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium.

[9]. *W. Boonpook, Y. Tan, Y. Ye, P. Torteeka, K. Torsri, S. Dong*, "A Deep Learning Approach on Building Detection from Unmanned Aerial Vehicle-Based Images in Riverbank Monitoring", Sensors 2018, 18, 3921.

[10]. *Y. Zhang, F. Canisius, B. Guindon, C. Zhen, B. Feng, J. Wang*, "Effectiveness of RGB imagery from diverse sources for real-time urban flood water mapping", Proceedings Volume 10793, Remote Sensing Technologies and Applications in Urban Environments III; 107930E (2018) https://doi.org/10.1117/12.2323495 Event: SPIE Remote Sensing, 2018, Berlin, Germany.

[11]. *A. Liberda, A. Lilja, B. Langborn, J. Lindstrom*, "Image Segmentation and convolutional neural networks as tools for indoor scene understanding", Mai 2016.

[12]. *I. Goodfellow, Y. Bengio, A. Courville*, Deep Learning, MIT Press, 2016.

[13]. *Umbaugh, Scott E.*, Computer Vision and Image Processing, Prentice Hall, NJ, 1998, ISBN 0-13-264599-8.

[14]. *R.C. Gonzales, R.E.Woods*, Digital Image Processing (2-nd Edition), Prentice Hall, 2002.

[15]. *J. Li, T. Wang, M. Gao, A. Zhu, G. Shan, H. Snoussi*, "Two Stream Neural Networks with Traditional CNN and Gabor CNN for Object Classification", 2018 37th Chinese Control Conference (CCC), DOI: 10.23919/ChiCC.2018.8483992.

[16]. *J.-K. Kamarainen, V. Kyrki, H. Kalviainen*, "Invariance properties of Gabor filter-based features-overview and applications", IEEE Transactions on Image Processing (Volume: 15, Issue: 5, May 2006), DOI: 10.1109/TIP.2005.864174.

[17]. *W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, Z. Wang*, "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1874-1883.

[18]. *V.K Ayyadevara*, "Convolutional Neural Network", Pro Machine Learning Algorithms Apress(2018), Berkeley, CA.

[19]. *M. Rahnemoonfar, R. Murphy, M.V. Miquel, D. Dobbs, A. Adams*, "Flooded Area Detection from UAV Images Based on Densely Connected Recurrent Neural Networks", IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, DOI: 10.1109/IGARSS.2018.8517946.