AUTOMATIC FLOOD DETECTION USING SENTINEL-1 IMAGES IN GOOGLE EARTH ENGINE

Le Hung Trinh[1,*], Dinh Sinh Mai[1], Nhu Hung Nguyen[1], Van Phu Le[1]

[1]*Le Quy Don Technical University*

Abstract

The paper presents a method to detect flooded areas based on the classification of superpixels from Sentinel-1 satellite image data. The automatic thresholding method based on the Simple Non-Iterative Clustering and the OTSU algorithm extracts flooded areas from Sentinel-1 images on the Google Earth Engine platform. Experimental results in Quang Binh province in October 2016 show the efficiency of the proposed method (Simple Non-Iterative Clustering algorithm combined with the OTSU algorithm). The proposed method accuracy is 91% higher than the OTSU algorithm, with about 86% in classifying flooded areas in SAR images. The results obtained in the study area can provide timely information about the flooded area, contributing to effective response and reduction of damage caused by floods. The study also shows the power of the Google Earth Engine platform. Google Earth Engine has supported rapid data analysis and processing thanks to rich data sources, multi-time, diverse data, and various algorithms.

Keywords: *Remote sensing; flood area detection; SAR image; Sentinel-1; Google Earth Engine.*

1. Introduction

Due to climate change, natural disasters in Vietnam have become more severe and tend to increase in both frequency and intensity. Due to the characteristics of climate and topography, floods are considered one of the most powerful natural disasters in Vietnam, causing significant damage to people and property. During floods, weather conditions affect optical satellite images, leading to substantial monitoring and disaster response [1]. These limitations can be overcome when using microwave satellite images (SAR-Synthetic Aperture Radar images) because the radar data is less affected by weather conditions and season time phases. Floods are among the most damaging natural disasters globally, especially in coastal countries with dense river systems like Vietnam. In recent years, floods have been complicated, especially in the central and southern regions of Vietnam, causing significant damage to people and properties [2]. Data classification techniques have been applied in many practical problems [3]. Detecting and classifying flooded areas from remote sensing data is a functional problem, providing objective and timely input for flood response and management

[*] Email: trinhlehung@lqdtu.edu.vn
https://doi.org/10.56651/lqdtu.jst.v4.n02.341.sce

models, supporting public search and rescue operations, helping to reduce damage caused by floods [4].

Because the weather during floods is often unfavorable, optical remote sensing image data is ineffective in classifying and detecting flooded areas. These limitations can be overcome when using SAR images because weather conditions affect radar signals less. Several algorithms for detecting flooded areas on SAR data have been proposed as both supervised and unsupervised classification methods [5], the method using thresholding [6, 7], image analysis based on subjects [8], and combined methods [9]. Among these approaches, thresholding using clustering algorithms is the most commonly applied one to analyze SAR images to separate water and no-water areas. The system is based on the backscatter difference of the low water object in the image, with surrounding objects having a higher backscatter value [10].

Currently, Google Earth Engine (GEE) is a free platform that enables cloud-based geospatial data collection and analysis. Therefore, Sentinel-1 satellite images were used on the GEE platform to identify flooded areas [1, 11]. In addition, Sentinel-1 images have been subjected to image preprocessing based on geometric errors and noise during image acquisition [12]. First of all, the exact orbit of the satellite was calculated continuously before generating the products on the European Space Agency website [13]. Besides, thermal noise removal is performed by calculating based on an available noise table with Sentinel-1 images [14]. In addition, terrain correction was performed using SRTM to simulate SAR images. Besides, the image's pixel values are converted into backscatter coefficients [15]. There are many clustering algorithms, among which Otsu algorithm is quite commonly used because of its low computational complexity and ease of implementation. In Vietnam, Otsu’s algorithm is applied in many fields, such as supporting disease diagnosis [16], remote sensing image processing [17, 18].

Therefore, the study uses Otsu’s algorithm combined with the Simple Non-Iterative Clustering algorithm on the Sentinel-1 image dataset to map flood-affected areas in Quang Binh during the October 2016 storm on the GEE platform.

2. Materials and Methodology

2.1. Materials

Quang Binh is a province located in central Vietnam where the weather is severe due to the influence of southwesterly winds. This wind carries a lot of moisture, so it often causes rain. Sea storms and northeast monsoons often cause heavy rains in Quang Binh and the central provinces. The study area is located at 17° 2' 50.25" - 17° 48' 44.86" North latitude, 106°11'26.65" - 107° 2' 2.17" East longitude.

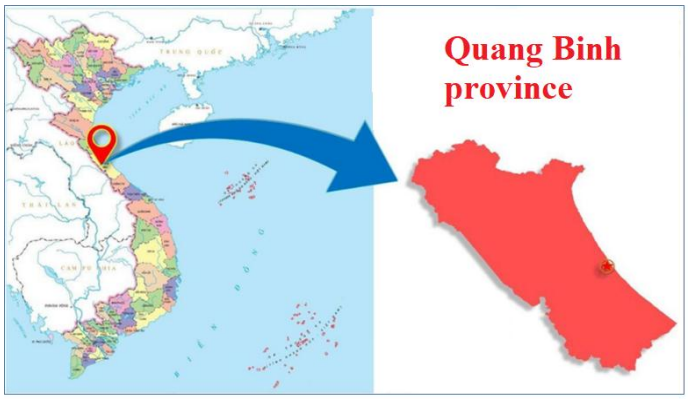
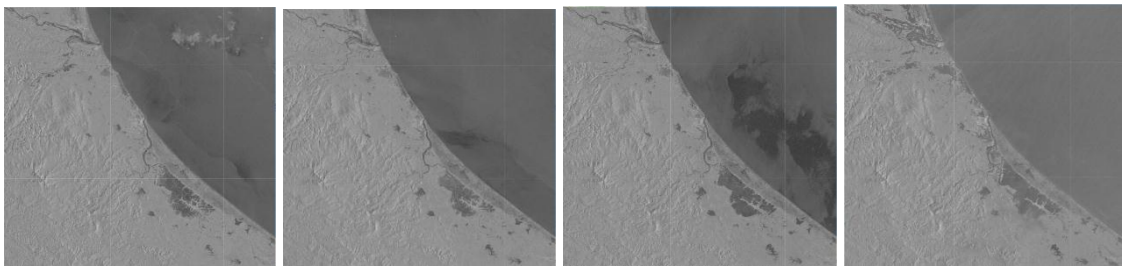


Figure 1. Research area on Vietnam map - Quang Binh province.

During the October 2016 storm, Quang Binh province suffered heavy damage caused by big storms. There were dozens of people killed and injured due to floods, tens of thousands of houses were flooded, roofs were blown off, thousands of hectares of vegetables and crops were damaged, and thousands of livestock and poultry were killed and washed away. Other categories were also severely affected. Some images caused by floods are shown in Figure 2. Figure 2a described the Quang Binh area when there was no flood. Figures 2b, 2c, and 2d described the Quang Binh area during the flood. Research using GRD product images of Sentinel-1 satellite image data. Level-1 Ground Range Detected (GRD) products focus on diverse, detected, and projected SAR data to the ground using a geometric model of the Earth [19].

Table 1. Specifications of Sentinel-1 satellite data and data used in the study

Data	**Time**	**Specifications**			
		Observation mode	**Band**	**Polarization**	**Spatial resolution**
Sentinel-1 GRD	13-09-2016	Wide interferometer mode (IW)	Band C (5.405 GHz)	VV	10m x 10m
	07-10-2016				
	19-10-2016				
	31-10-2016				



(a) *(b)* *(c)* *(d)*

Figure 2. Sentinel-1 GRD image of Quang Binh province used in the study: (a) 13-09-2016, (b) 07-10-2016, (c) 19-10-2016, (d) 31-10-2016 [19].

Figure 3. Some pictures of Quang Binh province during the flood 10/2016.

The ellipsoid projection of the GRD products is corrected using the topographic elevation of the product. The resulting product has an approximately square resolution, and the ground coordinates using the Earth ellipsoid model WGS84 are the inclined coordinates projected onto the ellipsoid projection of the Earth. The remote sensing data used in the study were collected in September and October 2016, which is the time when heavy rain occurs in Quang Binh province. The remote sensing data used in this study is shown in Figure 3.

2.2. Methodology

Image thresholding is one of the techniques in digital image processing. One of the thresholding algorithms is the OTSU algorithm. OTSU is the name of a Japanese researcher who came up with the idea for automatically calculating thresholds based on the pixel value of the input image to replace using a fixed threshold. OTSU's thresholding method is based on linear discriminant criteria, assuming that the image consists of foreground and background. The OTSU sets a threshold to minimize the overlap of layers [11]. The histogram is a graph that represents the frequency with which gray levels of an image appear. The OTSU algorithm calculates the parameters based on the gray histogram of the image.

First, the algorithm uses a gray histogram to represent the frequency of occurrence of gray levels. Each gray level i represents some pixel values, calculated by the formula:

$$p_i = \sum_{i=0}^{L-1} \frac{n_i}{P} \tag{1}$$

where n_i is the frequency of occurrence of the gray level of i value (the pixel number value), P is the total number of pixel values in the image, L is the number of intervals in which pixel values are divided.

Next, choose a threshold $T_k = k$ ($0 < k < L–1$) to divide the input image into two classes: Layer C_1 (the set of pixels with values $\leq k$). The ratio between the number of pixels of class C_1 to the total number of pixels is denoted by, similarly to class C_2 characterized by, the value of $P_1(k)$ and $P_2(k)$ is calculated according to formula (2) and (3), respectively:

$$P_1(k) = \sum_{i=0}^{k} p_i \tag{2}$$

$$P_2(k) = \sum_{i=k+1}^{L-1} p_i = 1 - P_1(k) \tag{3}$$

Then, calculate the mean m_1 of class C_1 according to formula (4):

$$m_1(k) = \frac{1}{P_1(k)} \sum_{i=0}^{k} ip_i \tag{4}$$

Similarly, calculate m_2 according to formula (5):

$$m_2(k) = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} ip_i \tag{5}$$

Algorithmically, OTSU will find the threshold k^* at which the value at which the difference between the two classes reaches the maximum value, denoted, is calculated by the formula (6):

$$\sigma_B^2(k^*) = \underset{0 \leq k \leq L-1}{Max}\, \sigma_B^2(k) \tag{6}$$

in which $\sigma_B(k)$ is the two-class variance C_1 and C_2. The value of $\sigma_B(k)$ is calculated by the formula (7):

$$\sigma_B^2(k) = \frac{[\mathrm{m}_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]} \tag{7}$$

where m_G is the average value of the image, calculated by the formula (8):

$$m_G = \sum_{i=0}^{L-1} ip_i = P_1 m_1 + P_2 m_2 \tag{8}$$

If there are many equal maximum values, we will choose the k with the most significant value, for example k^*, then we perform the threshold binary transformation according to formula (9):

$$g(x,y) = \begin{Bmatrix} 1, f(x,y) \leq k^* \\ 0, f(x,y) > k^* \end{Bmatrix} \tag{9}$$

The pseudocode of the OTSU algorithm is presented as follows:

Algorithm 1. OTSU threshold algorithm

Input: Image I and gray histogram of the input image. The number of gray histogram divisions L.
Output: Optimal thresholding value *t*.

1: Initialization: $m[:2] = 0$
2: **For** $k \in [0, 1, ...L\text{-}1]$ **do**:
3: Compute $P_1(k)$ and $P_2(k)$ by equations (*2*) and (*3*)
4: Compute $m_1(k)$ and $m_2(k)$ by equations (*4*) and (*5*)
5: **End**
6: Initialization $t = L/2$
7: Compute $\sigma_B^2(t)$ by equation (*7*) and m_G by equation (*8*)
8: **For** $k \in [0, 1, ...L\text{-}1]$ **do**:
9: Compute $\sigma_B^2(k)$ by equation (*7*)
10: **If** $\sigma_B^2(k) > \sigma_B^2(t)$ **then**:
11: Set $\sigma_B^2(t) = \sigma_B^2(k)$ and $t = k$
12: **End**
13: Return t

The Simple Non-Iterative Clustering algorithm (SNIC) [3] is built on the GEE platform. Theoretically, SNIC is a pixel clustering algorithm built on the Simple Linear Iterative Clustering algorithm (SLIC) [4]. SLIC performs *k*-means clustering on images with selected cluster centers and normalized color and spatial distance of pixels with cluster centers. Like SLIC, SNIC also initiates cluster centers which are established pixels on the image plane. The relationship of any pixel to a cluster center is measured by the distance in multidimensional space, including its color and spatial coordinates. This distance is calculated based on the normalized color and spatial distance. With spatial position *x* and color *c*, the distance of any pixel *j* to the center of the k^{th} superpixel cluster is made by the formula:

$$d_{j,k} = \sqrt{\frac{\left|x_j - x_k\right|_2^2}{s} + \frac{\left|c_j - c_k\right|_2^2}{m}} \tag{10}$$

where *s* and *m* are the normalization coefficients for spatial distance and color, respectively. Each of the expected total *K* superpixels will contain *N/K* pixels for an image with *N* pixels. The value of *s* in formula (10) is given by $\sqrt{N/K}$, and *m* value is provided by user. A higher value corresponds to the creation of smaller superpixels and vice versa.

In each *k*-means iteration, SLIC recomputes the cluster center by calculating the means of all the pixels closest to it in terms of *d* calculated by the formula (10). Thus, these pixels have the same label as the cluster center. Therefore, SLIC needs to request the number of iterations of the *k*-means to update the cluster centers. Thus, unlike SLIC, which requires multiple *k*-means iterations to update cluster centers, SNIC updates cluster centers in a single iteration. The cluster centers are updated directly after changing the pixels in the cluster. The pixels are, in turn, labeled with the original number of cluster centers. The algorithm terminates when all pixels are labeled.

In this study, the Sentinel-1 image dataset is processed on the GEE platform, a cloud platform that supports geospatial data processing to analyze the Earth's surface. In addition, GEE also provides a code editor based on Javascript language, where the code is developed with the functions of image preprocessing, image analysis, and mapping of flooded areas.

The research method is described in detail in Figure 4. Each processing step is detailed as follows:

Step 1: Image preprocessing

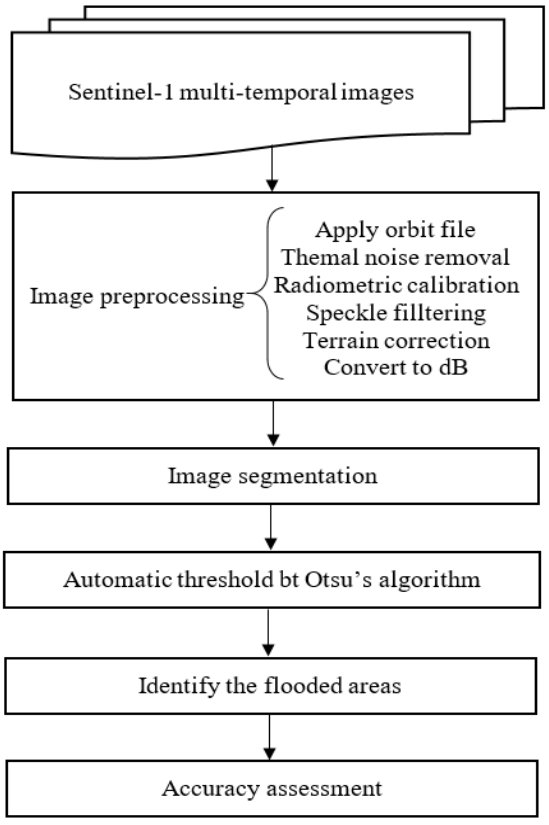


Figure 4. Research model.

This section uses multi-temporal Sentinel-1 image data to map flooded areas. Sentinel-1 GRD images are collected according to image type, image capture time, image observation mode. After that, the image preprocessing steps were performed. This includes a series of geometric correction and transformation steps related to the images used in the study. They are (1) Apply orbital file, (2) Remove thermal noise, (3) Radiometry calibration, (4) Noise filter, (5) Terrain correction, (6) Convert to dB.

Sentinel-1 images were preprocessed in the Google Earth Engine cloud platform. Finally, the new values for each pixel in the new position of the image are converted to the backscatter

coefficient. This factor is calculated in dB according to formula (11):

$$\sigma_{dB} = 10 \log abs(DN) \tag{11}$$

whereby,

$$DN = \frac{received\ energy\ by\ the\ sensor}{energy\ reflected\ in\ an\ isotropic\ way}$$

The backscattered coefficient can be a positive number if there is a focusing of backscattered energy towards the radar; it can be a negative number if there is a focusing of backscattered energy away from the radar (e.g., smooth surface).

Step 2: Fragment the image

The small speckle-noise effect of the original SAR image can cause errors. Before image fragmentation, the SAR image dataset is applied to a Gaussian filter. In addition, the image fragmentation process can misinterpret shaded areas related to the terrain, and mountain slopes can be mistaken as small bodies of water. At the same time, the water area created by the flood can be fragmented due to interference. First, the SAR images need to be smoothed by applying a Gaussian filter to reduce this effect. SAR image data set is defragmented, all pixels of the image will be grouped into clusters of superpixels. In which, the centers of superpixels are called cluster centers. The number of cluster centers K in algorithm (2) input is determined through the seeds parameter. The size of each piece is determined based on the value of the seeds. The value of tiny seeds corresponds to the number of large amounts and the long computation time. At the same time, the water areas will be split into many parts. However, if the value of seeds is significant, this will cause the size of the fragments to be substantial; the flooded areas will be able to interfere with surrounding objects. In the study, the value of seeds was 80, determined experimentally for the study area. This value corresponds to the initial cluster centers being 80 pixels apart. During SNIC image fragmentation, the algorithm calculates and gives the average value per cluster. The image fragmentation will consist of clusters that tend to be unequal. This is consistent with the natural characteristics of the study area observed during the flood.

Step 3: Automatic thresholding by OTSU algorithm

On SAR images, water bodies are usually on a flat, smoother surface than other rough ground objects. Besides, the electromagnetic wave signal emitted by the SAR is less reflected on its surface. Therefore, the gray values of the water pixels in the SAR image are lower, showing black or dark gray in the image. Due to their simplicity and

ease of use, thresholding methods are widely used in water extraction from radar images. The value of the classification threshold affects the accuracy of water extraction in SAR images.

The thresholding method based on the OTSU algorithm is used to determine the threshold in the study. OTSU is a widely used algorithm to determine the classification threshold value automatically. The advantage of the OTSU algorithm is that it is not affected by the contrast and brightness of the image. Research using the OTSU algorithm to look for the appropriate threshold *t* to separate water and no water areas in SAR images. The main principle is to use the idea of the OTSU algorithm, divide the image into two parts according to the gray level so that the difference between the parts is maximum. The process of calculating the optimal threshold *t* of the OTSU algorithm in the study is as follows:

$$M = P_{nw}.M_{nw} + P_w.M_w \tag{12}$$

$$\sigma^2 = P_{nw}.(M_{nw} - M) + P_w.(M_w - M)^2 \tag{13}$$

$$P_{nw} + P_w = 1 \tag{14}$$

$$t = \underset{0 \le t \le L-1}{ArgMax}\left\{\sigma^2\right\} \tag{15}$$

wherein formulas (13) and (15) are the variance between water and non-water classes. M_w and M_{nw} presented in formulas (12) and (13) are the mean pixel values of water and non-water objects, respectively. P_w and P_{nw} presented in equations (13) and (14) are values of the likelihood that a single pixel is water or not. The *M* value in formula (12) represents the average gray value of the entire image.

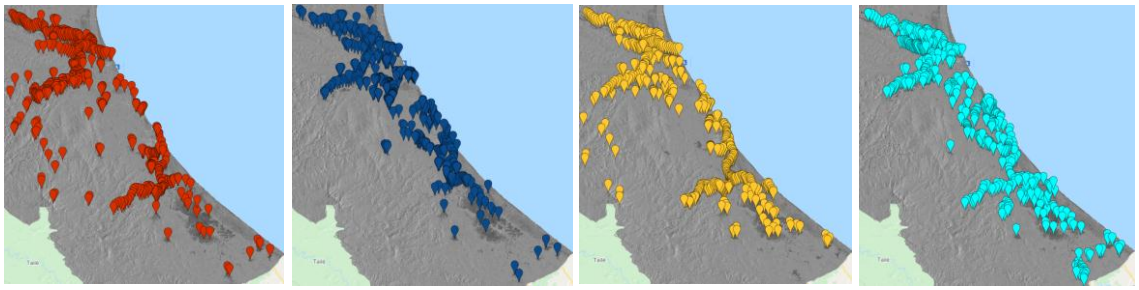
Step 4: Flood mapping

Based on the optimal threshold value *t* in formula (15), the SAR image will be divided into two parts based on the value of the superpixels. Determining the extent of inundation requires the acquisition of pre-and post-flood images. Therefore, after extracting the water by the thresholding method based on the OTSU algorithm, overlapping the water objects before and after flooding is necessary to form the flooded area.

Step 5: Evaluate the accuracy

In the study, the authors extracted random water and non-water points on the original image to evaluate the accuracy of the OTSU algorithm in the extraction of

water. Samples are randomly selected on the image. Each image sample represents a pixel on the original image. At the same time, the sample data set is also used to compare the accuracy between pixel-based image analysis and superpixel-based image analysis. For the most parts, water areas are based on discrete pixels. This is the main difference from superpixel-based extracted waters. Therefore, the samples taken affect the accuracy of the classification results. In particular, the mountainous areas often have noise, causing interference during the sampling process. The number of samples used in the images on September 13, 2016, October 7, 2016, October 19, 2016, and October 31, 2016, is 456, 453, 472, 481 samples, respectively.

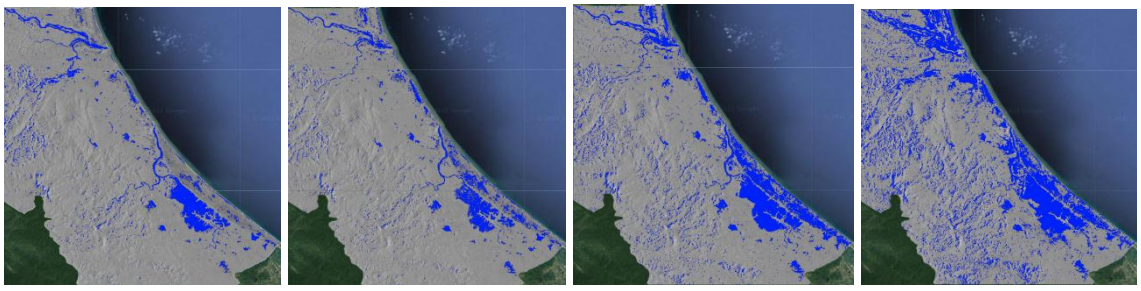


(a) *(b)* *(c)* *(d)*

Figure 5. Extract data truth to evaluate the accuracy of the OTSU algorithm (a) 13-09-2016, (b) 07-10-2016, (c) 19-10-2016, (d) 31-10-2016.

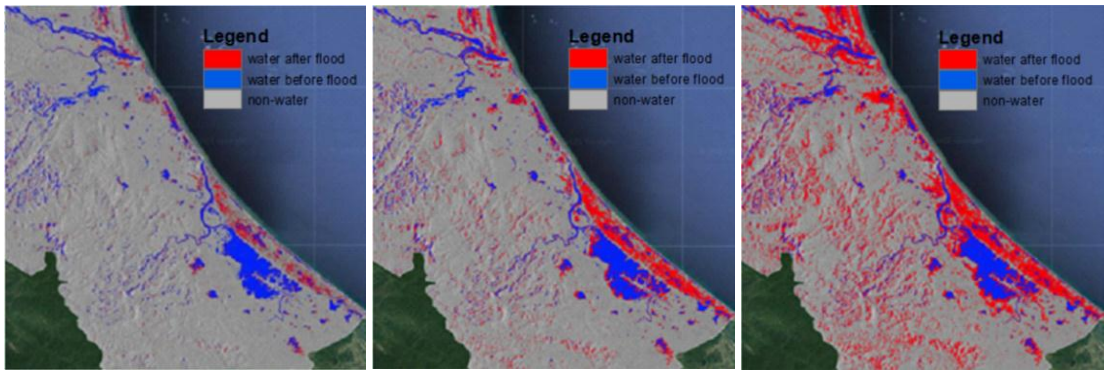
3. Result and discussion

Figure 6 is the result of surface water classification using the OTSU algorithm four times: when there was no flood (a) and three times when the flood occurred (b, c, d). Figure 7 shows the change in the flooded area, shown in red in the image. Similarly, Figure 8 and Figure 9 are the classification results by the improved OTSU algorithm and the change in surface water at three times when floods have occurred compared to when no floods have occurred.



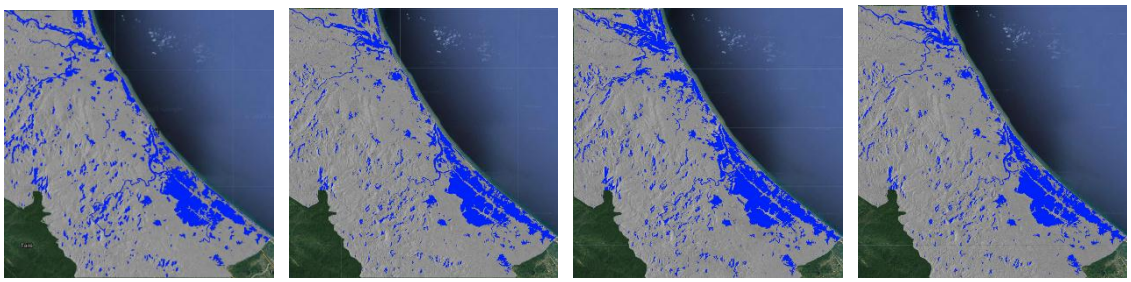
(a) *(b)* *(c)* *(d)*

Figure 6. Surface water extraction image using pixel-based OTSU algorithm: (a) 13-09-2016, (b) 07-10-2016, (c) 19-10-2016, (d) 31-10-2016.



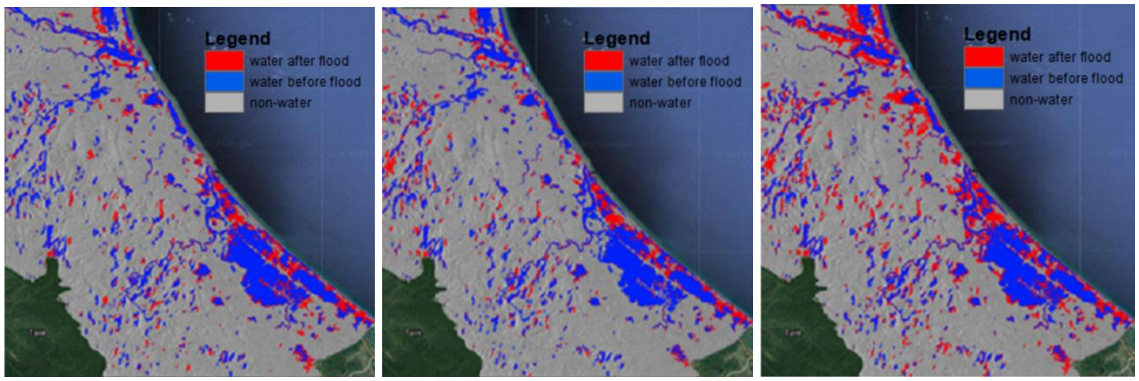
(a) *(b)* *(c)*

Figure 7. Change of flooded area over time using pixel-based OTSU algorithm. The red area represents the flooded area between the shooting times: (a) September 13, 2016 to October 7, 2016, (b) September 13, 2016 to October 19, 2016, (c) September 13, 2016 to October 31, 2016.



(a) *(b)* *(c)* *(d)*

Figure 8. Surface water extraction image using OTSU algorithm based on super pixels: (a) 13-09-2016, (b) 07-10-2016, (c) 19-10-2016, (d) 31-10-2016.



(a) *(b)* *(c)*

Figure 9. Change of flooded area over time using OTSU algorithm based on superpixels. The red area represents the flooded area between the shooting times: (a) September 13, 2016 to October 7, 2016, (b) September 13, 2016 to October 19, 2016, (c) September 13, 2016 to October 31, 2016.

Table 2. Accuracy of classification methods

Image data	Superpixels			Classification based on pixels		
	True	False	Accuracy	True	False	Accuracy
13-09-2016	417	39	91.48%	420	36	**92.71%**
07-10-2016	416	37	**91.81%**	401	52	88.52%
19-10-2016	437	35	**92.58%**	416	56	88.14%
31-10-2016	449	32	**93.15%**	418	63	86.90%

Table 3. Area of the flooded area by the time of image acquisition

Image data	13-09-2016	07-10-2016	19-10-2016	31-10-2016
Total surface water area	240.53 km^2	495.21 km^2	539.48 km^2	557.59 km^2
Flooded area		254.68 km^2	298.95 km^2	317.06 km^2
Percentage of surface water	5.77%	11.88%	12.94%	13.37%

The use of a superpixel-based algorithm combined with the OTSU algorithm to perform thresholding of surface water and non-water layers for images of Quang Binh province in October 2016 has shown better results than the pixel-based classifier. The accuracy of water classification is over 90% in both methods (Table 2). However, when the flooding process occurred, the superpixel-based classification algorithm performed better than the pixel-based classification algorithm. At the same time, as the surface water area on the SAR image becomes larger and larger, the difference in accuracy between the two methods becomes more and more apparent. This result is based on the alignment of pixels close to each other in creating superpixels. With the pixel-based algorithm, most wrong pixels fall on the edge of the surface water areas.

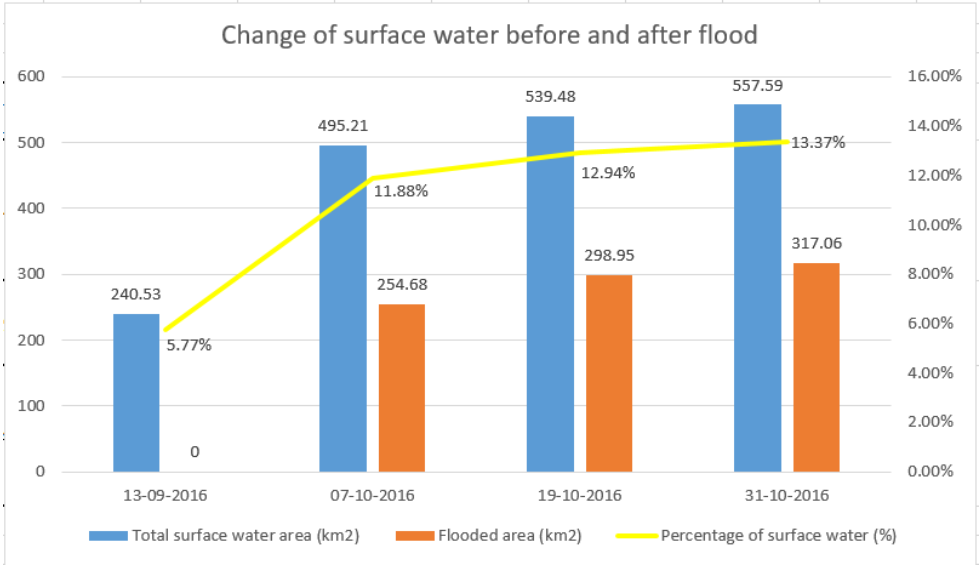


Figure 10. The graph shows the change of surface water before and after the flood.

Moreover, in Table 3, the surface water area has changed markedly before and after the flood. Before the flood, the surface water area of the study area was 240 km^2. However, right at the beginning of the flood, the surface water area more than doubled before the flood, reaching 495.21 km^2, equivalent to 11.88% of the study area. In the later period of flooding, the flooded area increased slightly, reaching 557.59 km^2 as of October 31, 2016. It shows that the area of water formed by the flood is larger than the flood's initial surface water area when there is no flood. Most of the flooded areas are concentrated at the edge, adjacent to the sea. This is a place with flat terrain, low elevation. Therefore, when a flood occurs, this is the place where most of the surface water is formed during the flood. At the same time, in the western part of the study area, this is a forest area with mostly tall trees in the pre-flood period. Therefore, when a flood occurs, most of the flooded area is formed in low-lying water holes located at the foot of the hill. The flooded area in this area is much less than in the delta area. In addition, some surface water is formed in rising water in rivers and lakes, causing flooding in the surrounding regions.

4. Conclusion

This paper presents a method to create inundation maps from SAR image data using the improved OTSU automatic thresholding algorithm. This paper uses the SNIC clustering technique to create superpixels before classification by OTSU. Thereby helping the OTSU algorithm to classify directly on these superpixels. This will help the OTSU algorithm run faster and still ensure accuracy. The experimental results show that the OTSU method combined with the SNIC algorithm gives 91% higher accuracy than the method using only the OTSU algorithm with only 86% or more accuracy. This research result can temporarily classify flood areas from SAR image data to support search and rescue and disaster reduction. The study also shows that GEE is a free platform that enables cloud-based geospatial data analysis, assisting in creating various types of real-time maps.

References

[1] Amitrano D., Martino G., Iodice A., Riccio D., Ruello G., "Unsupervised rapid flood mapping using Sentinel-1 GRD SAR images", *IEEE Transactions on Geoscience and Remote Sensing*, 56(6), 3290-3299, 2018, DOI: 10.1109/TGRS.2018.2797536.

[2] Chapman B., McDonald K., Shimada M., Rosenqvist A., Schroeder R., Hess L., "Mapping regional inundation with spaceborne L-band SAR", *Remote Sensing*, 7, 5440-5470, 2015.

[3] Achanta R., Shaji A., Smith K., Lucchi A., Fua P., Susstrunk S., "SLIC superpixels compared to state-of-the-art superpixel methods", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11), 2274-2282, 2012.

[4] Achanta R., Susstrunk S., "Superpixels and polygons using simple non-iterative clustering". *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 4651-4660, 2017, 10.1109/CVPR.2017.520.

[5] Trinh Le Hung, Andrade E., Pham Tuan Anh, "Application of remote sensing to extract flood areas using ENVISAT ASAR data", *Journal of Sciences, Orel State Agrarian University*, 1(52), 36-42, 2015.

[6] Cao H., Zhang H., Wang C. Zhang B., "Operational flood detection using Sentinel 1 SAR data over large areas", *Water*, 11(4), 2019.

[7] Li J., Wang J., Ye H., "Rapid flood mapping based on remote sensing cloud computing and Sentinel-1", *Journal of Physics: Conference Series*, 1952, 022051, 2021.

[8] Simon R., Tormos T., Danis P., "Geographic object based image analysis using very high spatial and temporal resolution radar and optical imagery in tracking water level fluctuations in a freshwater reservoir", *South-Eastern European Journal of Earth Observation and Geomatics,* 3, 287, 2014.

[9] Gasparovic M., Klobucar D., "Mapping floods in lowland forest using Sentinel-1 and Sentinel-2 data and an object-based approach", *Forests*, 12, 553, 2021.

[10] Qiu J., Cao B., Park E., Yang X., Zhang W., Tarolli P., "Flood monitoring in rural areas of the Pearl river basin (China) using Sentinel-1 SAR", *Remote Sensing,* 13, 1384, 2021.

[11] Otsu N., "A threshold selection methodfrom gray-level histograms", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 9, 62-66, 1979.

[12] Moothedan A., Dhote P., Thakur P., Garg V., Aggarwal S., Mohapatra M., "Automatic flood mapping using Sentinel-1 Grd Sar images and Google Earth Engine: A case study of Darbhanga, Bihar", *The Proceedings of National Seminar on Recent Advances in Geospatial Technologies & Applications*, IIRS Dehradun, India, March 02, 2020.

[13] Uddin K., Matin M., Meyer F., "Operational flood mapping using multi-temporal Sentinel-1 SAR images: A case study from Bangladesh", *Remote Sensing,* 11(13), 1581, 2019.

[14] Elfadaly A., Abate N., Masini, N., Lasaponara R., "SAR Sentinel 1 imaging and detection of palaeolandscape features in the Mediterranean area", *Remote Sensing,* 12(16), 2611, 2020.

[15] Park J., Korosov A., Babiker M., Sandven S., Won, J., "Efficient thermal noise removal for Sentinel-1 TOPCAR cross-polarization channel", *IEEE Transactions on Geoscience and Remote Sensing,* 56(3), 1555-1565, 2017.

[16] Richards J. A., "Remote sensing with imaging radar", *Springer,* 380p, 2009, ISBN: 9783642020193.

[17] Nguyen Le Mai Duyen, Truong Minh Thuan, "A combination of threshold and Graphcut method in medical image analysis assists diagnosis", *DTU Journal of Science and Technology,* 01(32), 88-99, 2019.

[18] Tran Thanh Tung, Mai Duy Khanh, "Nghiên cứu quy luật diễn biến đổi cát ven bờ khu vực cửa Tiên Châu bằng ảnh vệ tinh Landsat", *Khoa học Kỹ thuật Thủy lợi và Môi trường*, Số 71 (12/2020), 2020.

[19] https://developers.google.com/earthengine/datasets/catalog/COPERNICUS_S1_GRD update on September 2016.

[20] https://sentinels.copernicus.eu/web/sentinel/user-guides/sentinel-1-sar/overview update on September 2016.

TỰ ĐỘNG PHÁT HIỆN VÙNG NGẬP LỤT TỪ ẢNH SENTINEL-1 SỬ DỤNG NỀN TẢNG GOOGLE EARTH ENGINE

Trịnh Lê Hùng, Mai Đình Sinh, Nguyễn Như Hùng, Lê Văn Phú

Tóm tắt: *Bài báo trình bày giải pháp phát hiện các khu vực ngập lụt dựa trên việc phân loại các siêu điểm ảnh từ dữ liệu ảnh vệ tinh Sentinel-1. Phương pháp tạo ngưỡng tự động dựa trên thuật toán phân cụm không lặp lại đơn giản (SNIC) và OTSU được sử dụng để trích xuất vùng ngập nước từ hình ảnh Sentinel-1 trên nền tảng Google Earth Engine (GEE). Kết quả thực nghiệm tại tỉnh Quảng Bình vào tháng 10 năm 2016 cho thấy hiệu quả của phương pháp đề xuất (thuật toán SNIC kết hợp với thuật toán OTSU) cho độ chính xác cao hơn 91% so với phương pháp chỉ sử dụng thuật toán OTSU với khoảng 86% trong việc phân loại vùng ngập từ ảnh SAR. Kết quả nhận được trong nghiên cứu có thể cung cấp thông tin kịp thời về khu vực ngập lụt, góp phần ứng phó hiệu quả và giảm thiểu thiệt hại do lũ lụt gây ra. Nghiên cứu cũng cho thấy tiềm năng của nền tảng GEE. GEE đã hỗ trợ việc phân tích và xử lý dữ liệu nhanh nhờ nguồn dữ liệu phong phú, đa thời gian, đa dạng dữ liệu và đa dạng thuật toán.*

Từ khóa: Viễn thám; phát hiện vùng ngập lụt; ảnh ra đa; Sentinel-1; Google Earth Engine.

Received: 05/11/2021; *Revised:* 24/11/2021; *Accepted for publication:* 28/12/2021

❑