Article

Three-Dimensional Convolutional Neural Network on Multi-Temporal Synthetic Aperture Radar Images for Urban Flood Potential Mapping in Jakarta

Indra Riyanto ¹ , Mia Rizkinia ¹, Rahmat Arief ² and Dodi Sudiana ^{1,*}

¹ Department of Electrical Engineering, Faculty of Engineering, Universitas Indonesia, Depok 16424, Indonesia; indra.riyanto81@ui.ac.id (I.R.); mia@ui.ac.id (M.R.)

² Aeronautics and Space Research Organization, National Research and Innovation Agency, Jakarta 13710, Indonesia; rahmat.arief@lapan.go.id

* Correspondence: dodi.sudiana@ui.ac.id

Abstract: Flooding in urban areas is counted as a significant disaster that must be correctly mitigated due to the huge amount of affected people, material losses, hampered economic activity, and flood-related diseases. One of the technologies available for disaster mitigation and prevention is satellites providing image data on previously flooded areas. In most cases, floods occur in conjunction with heavy rain. Thus, from a satellite's optical sensor, the flood area is mostly covered with clouds which indicates ineffective observation. One solution to this problem is to use Synthetic Aperture Radar (SAR) sensors by observing backscatter differences before and after flood events. This research proposes mapping the flood-prone areas using machine learning to classify the areas using the 3D CNN method. The method was applied on a combination of co-polarized and cross-polarized SAR multi-temporal image datasets covering Jakarta City and the coastal area of Bekasi Regency. Testing with multiple combinations of training/testing data proportion split and a different number of epochs gave the optimum performance at an 80/20 split with 150 epochs achieving an overall accuracy of 0.71 after training in 283 min.

Keywords: urban flood; Sentinel-1a; Synthetic Aperture Radar (SAR); 3D Convolutional Neural Network; multi-temporal data

Citation: Riyanto, I.; Rizkinia, M.; Arief, R.; Sudiana, D. Three-Dimensional Convolutional Neural Network on Multi-Temporal Synthetic Aperture Radar Images for Urban Flood Potential Mapping in Jakarta. *Appl. Sci.* **2022**, *12*, 1679. <https://doi.org/10.3390/app12031679>

Academic Editors: Anselme Muzirafuti and Dimitrios S. Paraforos

Received: 20 December 2021

Accepted: 31 January 2022

Published: 6 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Flooding is one of the most detrimental disasters, especially in cities such as Jakarta, because it affects a large number of residents in ways such as material losses resulting from damaged properties due to flood inundation and diseases caused by degraded sanitation in the flooded area. A major flood in Jakarta results in 8.7 trillion IDR or 625 million USD of losses and recovery efforts [1]. At present, most of the flood mapping in Indonesia has not fully utilized satellite spatial data because it still relies on data reported by the local government in the form of numerical data [2]. The visualization of the flood map is based on tabulated data in the area map that does not represent the actual conditions, resulting in a discrepancy between the reported flood area and the actual area. This difference will affect the handling of floods, such as calculating the impact of damage, the number of residents affected by the flood, and the inefficient distribution of aid. Problems that arise due to limited spatial information regarding floods can be solved by using multi-sensor remote sensing satellite data. Many technologies have been developed to predict, prevent and mitigate flood disasters more accurately, including remote sensing technology using images obtained from airborne and spaceborne platforms [3–5]. The earlier and most common form of remote sensing is optical photography, with overhead images providing information on the affected area.

When a disaster occurs, urban floods usually coincide with rain, so when observed using optical sensors on remote sensing satellites, the area is covered with clouds. With this occlusion, satellite optical sensors are not effectively used for flood observation. One solution to observing floods in cloud-covered areas is to use Synthetic Aperture Radar (SAR) sensors such as Sentinel-1, ALOS PALSAR, TerraSAR-X, and other radar sensors. The image produced by SAR is a monochromatic image containing reflectance information from the observation area by observing the difference in the backscatter before and after the incident to identify the flooded area [6].

Wide-scale Earth monitoring satellites begin with the Landsat program (Land Satellite) to monitor the Earth's surface. At present, many optical satellite systems operate at high resolution. The most widely used images are from QuickBird, SPOT-5, and Worldview Series. Despite having the ability to detect very sharp objects, optical satellite systems have the potential to be unable to detect objects on the Earth's surface due to cloud cover. Until recently, SAR data processing was mostly used for rural areas rather than in urban areas. For urban areas, SAR data has its problems, namely speckle noise, because, in urban areas, many buildings cause radar waves to experience much scattering, and the reading of the reflected waves is disturbed by multipath interference [7,8]. The double-bounce characteristic of radar signals caused by buildings is a challenge related to its contribution to SAR image speckle noise. However, it can be used to detect the presence of buildings and distinguish them from other surfaces such as soil, vegetation, and water [9–11]. The solution to detect the presence of floods through SAR images is multi-temporal filtering, which is filtering based on changes in the backscattering characteristics of SAR images taken at different times. The method used to identify and distinguish the occurrence of flood from SAR data is generally divided into two groups, namely polarization and interference. The polarization method detects the presence of water, based on changes in backscatter polarization caused by specular reflections from the water surface [10,12–14], while the interference method detects water based on changes of coherence due to changes in the spatial distribution pattern of the object that produces the backscatter [10,15–17].

Several studies related to floods in settlements used SAR and Light Detection and Ranging (LiDAR) data for small towns located on one side of a river and areas with homogeneous slopes [18–20]. Another study used the Decision Tree Classifier (DTC) to differentiate surface types based on light reflection [21,22], a condition that is difficult to meet because flooding occurs with cloudy skies. A seasonal disaster mapping system based on field observations is practical for generating initial data [23,24] but impractical for periodic events. Research on radar satellites for disaster management to date still has elevation resolution in the range of 2 m [6,7,25,26]. For Jakarta, especially in the city center to the northern coast, which often experiences floods, this is not appropriate because the flood depth is less than 2 m [27].

To map floods, previous methods were initially dominated by thresholding [6,11,13,14,28], Probability Density Function (PDF) [9,10], and more recently, Logistic Regression [29] and the Storm Water Management Model (SWMM) [30]. More specifically, in remote sensing image segmentation applications, some researchers use the Normalized Difference Vegetation Index (NDVI) to classify vegetation, classify water levels with the Normalized Difference Water Index (NDWI), and the classification of floods using Normalized Difference Flood Index (NDFI) [31,32]. Other researchers detected changes in land surface with radar images using the principle of interferometry to find coherence between images [15,16,33]. Research in the last three years has led to the use of Machine Learning (ML) in order to segment and classify increasingly complex datasets, such as the Adaptive Neuro-fuzzy Inference System (ANFIS) [34], Support Vector Machine (SVM) [35,36], Convolutional Neural Network (CNN) [36–38], and more recently with various Swarm Intelligence (SI) variants such as Particle Swarm Optimization (PSO) [39,40]. Sameen and Pradhan used a Residual Neural Network to detect the potential for landslides, where this method is intended to detect changes in soil texture as the initiation of landslides [38]. The CNN model with this residual block is used to process LiDAR data. The neurons on CNN are activated using the Rectified

Linear Unit (ReLU). This study proposes three-dimensional CNN in mapping flood areas to filter and weigh neurons and map the potential flood areas in urban areas with better accuracy and fewer number of images.

Yu Li et al. proposed an Active Self Learning method on CNN to detect floods in urban areas from the SAR image ensemble [37]. The dataset used is four TerraSAR-X images of HH polarization with the composition of one pre-event image, one co-event image, and two post-event images. Linyi Li et al. proposed a high-resolution urban flood mapping method (Super-Resolution Mapping of Urban Flood, SMUF) with the fusion of the Support Vector Machine and General Regression Neural Network (FSVM-GRNN) [35]. Because the urban flood area in the observation area is not very dense, the accuracy of this FSVM-GRNN is 80.2%.

Shen et al. proposed a machine learning process to make corrections to the mapping of flood inundation areas in near-real-time (NRT) using SAR, where the observation area is an open area without many obstacles on the surface [41]. At the time of segmentation, there are difficulties in classifying areas that are flooded with areas that have surface reflection properties similar to those of the water surface. ML is performed to correct speckle noise and another scattering, which can interfere with data reading and classification. The filtering method is used in most SAR image processing but its effect is to reduce the effective resolution and change the signal statistics and cannot completely remove noise. To overcome this, Shen et al. used the Logistic Binary Classifier (LBC) in a correction step to practice detecting the presence of water in the pixels contained in the water bodies and the surrounding buffer areas.

The objective of this work is to investigate the mapping of flood potential in Jakarta and nearby coastal areas using three-dimensional CNN on co-polarized (VV) and cross-polarized (VH) Sentinel-1a SAR images. A three-dimensional classification combines the two-dimensional image and one-dimension multi-temporal processes into a single convolution. The images are then pre-processed into grayscale images to be converted into a vector data format. The 2 January 2020 images were also sampled as flood and non-flood target sub-images, along with the corresponding locations from other images, to form the multi-temporal value changes of the flooded locations along with the consistency of the non-flooded locations. The CNN training is performed with training/test percentage values of 70/30; 80/20; and 90/10 with varying epochs between 100 and 160 iterations to obtain the best combination with the highest accuracy and the shortest processing time.

2. Materials and Methods

2.1. Location and Data

A radar image is generated from the reflection of active microwaves emitted from the radar vehicle (airplane or satellite). The transmitter in a radar system is an antenna array consisting of a waveguide and emitting a very narrow beam of microwave waves. The radar sensor moves along a trajectory, and the area highlighted by the radar moves (known as the footprint) along the surface being swept to form an image. A digital radar image consists of many pixel dots representing the backscatter or backscatter of a point on the surface. Figure 1 shows an example of a SAR image from 2 January 2020 that is free from cloud cover with the bright dots showing high backscattering while the dark dots represent low backscattering, while Figure 2 is the optical image from the same date showing cloud cover.

The radar system generally has a wavelength undisturbed by interference from water particles and water vapor in the air (clouds and rain). Because they are not dependent on illumination (irradiation) from the sun or other sources, the radar system can function day and night and in all weather. Synthetic Aperture Radar (SAR) works by detecting the phase-change of reflected signal caused by the movement of the platform to obtain the surface image with good resolution (i.e., visually discernible). The SAR system is generally divided into two wavelengths, namely short (C-band and X-band) and long (L-band and P-band) waves. Early SAR satellite systems use a single platform such as

Radarsat. Currently, the most commonly used are satellite constellation systems such as TerraSAR-X and TanDEM-X pair, four-satellites Cosmo-SkyMed in X-band, three-satellites Radarsat Constellation Mission, and Sentinel 1 satellite pair, which give shorter revisit time and higher temporal resolution [20,41–43]. Figure 3 shows the backscatter mechanism of shortwave radar (illustrated with black arrows) and long (illustrated arrows in blue) on various surfaces under normal conditions and during a flood. On the grass surface, there are surface reflections at both wavelengths due to the relative roughness of the surface. For short waves, the scattering is due to the thickness of the grass, while long waves can penetrate deeper [44].

Figure 1. SAR image of Jakarta on 2 January 2020.

Figure 2. An optical image from Landsat-8 of Jakarta on 2 January 2020.

When a flood occurs, specular reflection occurs in both types of waves. On objects in the form of trees or forests, the reflection is dominated by the scattering volume. For short waves, the scattering comes from the canopy (leaves) of the trees, while the longwave scattering by the branches and other tree structures is added by a double-bounce, which hits the ground surface and then the tree trunk or vice versa. When there is a flood, the double reflection will get more significant due to the specular reflection on the water surface (shown as a thick line of the direction of the reflection). In urban areas, the reflection on both waves is dominated by multiple reflections, although the surface will appear coarser on short waves. When there is a flood, this double reflection will be significantly strengthened due to the specular reflection on the water surface (shown as a thick line of the direction of the reflection).

Figure 3. Radar wave backscatter mechanism on the surface of the object (a) Grass/Land, (b) Vegetation, and (c) Urban in normal and flood conditions, for short wavelengths (C- and X-band) and long waves (L- and P-band).

In this study, the flood data collected came from Sentinel-1a remote sensing satellite. The data are downloaded through Google Earth Engine by the Copernicus catalog and selecting available dates from the archive. The selected mode is Interferometric Wide Swath (IW) with 250 km swath and $5 \text{ m} \times 20 \text{ m}$ spatial resolution [45]. For our model, the pixel resolution is preset at $10 \text{ m} \times 10 \text{ m}$. Remote sensing data combined with GIS data are integrated to create a flood hazard and potential map. Based on information obtained from remote sensing and GIS databases, the ML method can be applied for spatial modeling of flood vulnerability.

The data shown in Figure 4 are divided by the date of acquisition into three categories, namely: pre-event, consists of data from November to December 2019, which represent conditions before the major flood occurred; co-event is data taken on or near the 2 January 2020 flood, and the rest is categorized as post-event data. In Figure 5, the SAR images are set into a dataset, which contains 39 cross-polarized and 39 co-polarized images from Sentinel-1a between November 2019 and October 2020, with co-event images designated as the target image. The dataset SAR was collected using Google Earth Engine and consisted of Sentinel-1a VV and VH images between November 2019 and December 2020 as RGB composite TIF images. All images are resized into 946×681 pixels covering the Jakarta area and part of the Bekasi and Tangerang Regencies that flooded. The target image is further broken down as flood markers to make a 25×25 pixels-sized kernel. The previous individual images shown in Figure 5 are combined into three-dimensional data with a $946 \times 681 \times 78$ pixel dataset and $25 \times 25 \times 20$ pixel kernel.

2.2. Image Segmentation and Classification

In a digital image processing application, the primary process is segmentation to detect and identify objects and components within the image. The segmentation process divides the image into parts known as constituent objects. Automatic segmentation is generally the most challenging image processing [12]. With the development of image processing algorithms, image segmentation is also developed using region growing and merging, namely by expanding pixels so that the object becomes larger. In the end, some objects close to the same value will merge into one other, bigger object. This mathematical

algorithm is the basis for developing an image segmentation algorithm that carries out the unsupervised segmentation process without human intervention.

Figure 4. Sample images from co-polarization (VV) and cross-polarization (VH): (a,b) Pre-flood event; (c,d) Co-event during January 2020 flood; (e,f) Post-flood event.

Kwak et al. created a SAR satellite data processing algorithm to detect urban floods near-real-time using data before and after a flood event. Furthermore, the image is classified using a supervised classification to obtain the flood area based on building classes. The developed Probability Density Function (PDF) method can reduce the maximum backscatter intensity difference for rice fields and open areas by 35 dB; however, for urban areas, it has increased by 25 dB [9]. Further development of this method can reduce variance by 12 dB and increase urban areas by 15 dB [10]. In comparison, Liang et al. [46] used PDF to estimate the maximum similarity before thresholding by comparing the Otsu, Split-KI (Kettler and Illingworth), and Local Thresholding (LT) methods. The Overall Accuracy (OA) results obtained from the Sentinel-1a image classification in the Louisiana plain were 98.12% (Otsu), 98.55% (Split-KI), and 98.91% (LT), respectively.

Figure 5. Sentinel-1a images: (a) co-polarized (VV); (b) cross-polarized (VH); (c) flood markers.

Pelich et al. proposed the creation of a large-scale global database for flood inundation maps derived from the SAR dataset [28]. The method used is histogram thresholding to delineate quickly, then the level of flood distribution is extracted from the SAR backscatter using the Probability Density Function (PDF). Thresholding is performed using the Hierarchical Split-Based Approach (HSBA) to identify pixels with a bimodal distribution on the sub-pixels, which indicates that there is an immersion limit on these pixels [47]. The accuracy of the results obtained from flood detection in rural areas is 35%.

Another technique in flood detection is to utilize the polarization characteristics of radar signals, namely the Interferometric SAR (InSAR) method. The principle of stable scatterer or persistent scatterer is used to detect areas that do not experience changes in reflection characteristics, while changes in reflection characteristics result in low coherence between image data and are assumed to be flooded. The mapping is built by creating 20 interferometric pairs from 22 consecutive Sentinel-1a images with a composition of 17 pairs of pre-event images, a pair of images during a flood, and two pairs of post-event images [48]. Chini et al. also integrated intensity data using InSAR coherence, normalized cross-correlation to detect the presence of water in urban areas and mapping of double-bounce-producing objects using histogram thresholding and region growing. Pixels are categorized as floods when there is a decrease in coherence on the RGB composite channel [16].

In line with the development of the field of artificial intelligence, image processing methods also develop by making use of artificial intelligence functions. Several artificial intelligence methods that are widely used in image processing are Artificial Neural Networks or ANNs. The method that has recently begun to be applied in studies of mapping flood potential and vulnerability is to use machine learning. Some of the methods that were implemented include Adaptive Neuro-Fuzzy [34], Support Vector Machine (SVM) [35,36], Convolutional Neural Network (CNN) [36,38], and Swarm Intelligence [39,40,49]. Dasgupta et al. used Gamma Maximum A-Posteriori (GMAP) to filter out speckle noise from SAR images, then performed surface texture analysis using the Gray Level Co-Occurrence Matrix (GLCM) [34].

Although being the most common basic method on flood mapping, NDFI/NDWI as the most straightforward method tends to amplify noise greatly. Otsu thresholding suffers from high computational requirements since it is an early optimization method. The SMUF, SVM, GRNN [34,35], and most recently CNN [50] still perform the classification process in a 2D plane and then perform the 1D multi-temporal process. Due to the complexity of the factors that influence the occurrence of floods in urban areas, the most effective and efficient classification method is needed. As a classification technology developed based on feature matching, the ML method produces a more accurate recognition than feature matching. However, it has limited extraction features that can cause errors in the computation process. This study proposes three-dimensional CNN in mapping flood areas to filter and weigh neurons and map the potential flood areas in urban areas with fewer images compared to the previous study [36,51]. CNN features unsupervised feature extraction compared to Artificial Neural Network (ANN), in which the process is achieved through the training phase to recognize flood areas. In ANN, all neurons of a layer are fully connected to every neuron from other layers, whereas in CNN, only the last layer of neurons is fully connected due to the parameter-sharing nature of the CNN, therefore the computational load of CNN is less than ANN.

2.3. Deep Learning Neural Network

Recent developments in the Deep Learning Neural Network (DNN) are increasingly opening up great opportunities in flood mapping research. Deep Learning as one of the Machine Learning models has shown promising results in image processing and pattern recognition. Therefore, this research will propose mapping the potential flood areas using the DNN algorithm. DNN is based on Artificial Neural Network and generally consists of an input layer, with more than one hidden layer and one output layer [52]. Figure 6 shows the conceptual structure of the DNN model used for flood vulnerability mapping. The input layer is the factor that affects flood ($F1-Fn$). The information is processed and analyzed in the hidden layer to determine the weight and classification of each pixel. The final result of the classification is an indication that there is a flood in the output layer with two possible labels: Flood (positive class) and Others (negative class).

Figure 6. DNN structure concept for mapping flood potential.

DNN is a feed-forward network and is trained using the back-propagation method. However, more hidden layers will make the network challenging to train because of the different adjustment speeds in the hidden layer. DNN was implemented successfully in various applications, especially in automatic image recognition, speech recognition, language processing, and some applications in remote sensing. There is no rule of thumb about the number of hidden layers and neurons in each layer since it depends on the complexity of the problem and the conditions of the dataset.

The number of hidden layers in DNN has the advantage of representing a very complex relationship between factors. The hidden layer on DNN has neurons that are activated with the Rectified Linear Unit (ReLU) function as an alternative whose computation is more straightforward when compared to the sigmoid. Because DNN is trained on the principle of back-propagation, ReLU can minimize the decrease in learning gradient, hindering the learning process. Mathematically, the ReLU activation function can be expressed as $h'(x)$ in Equation (1).

$$h'(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \quad (1)$$

Hidden layers in DNN perform increasingly complex feature transformations to produce a more discriminatory feature abstraction. The classification results displayed in the output layer are based on the most abstract features obtained in the last hidden layer. During the DNN learning phase, the connection weights between layers are adjusted to reduce the difference between observed and predicted results. The back-propagation process trains DNN by providing feedback on the error results to the hidden layer. The deviation between the observed and predicted results is expressed in the loss function between entropies, as expressed in Equation (2).

$$L = -\frac{1}{N_D} \sum_{n=1}^{N_D} T \ln(Y) + (1 - T) \ln(1 - Y) \quad (2)$$

where N_D is the number of training data points, T represents the observed output, and Y represents the predicted output. The back-propagation learning gradient used for the training sample of m is formulated in Equation (3):

$$g = \frac{1}{m} \sum_{i=1}^C \frac{\partial L}{\partial w} \quad (3)$$

where L is the loss function, w represents the network weight, and $C = 2$ represents the number of output classes used (flood and others).

2.4. Convolutional Neural Network (CNN)

CNN is one type of DNN that uses the convolutional principle in its data processing. The basic concept of CNN architecture is to utilize a convolutional layer to detect the relationship between the features of objects and a pooling layer to similar group features. The CNN architecture consists of a series of layers, namely the Convolutional Layer (CL), which functions to transform a set of activations with a differential function, a Pooling Layer, and the final result is a Fully Connected Layer (FCL). Unlike other neural networks where all neurons are fully connected with every other neuron of the next layer, CNN disregards zero-valued parameters and makes fewer connections between layers. The non-zero parameters can be shared to be used by more than one connection in the layer to reduce the number of connections. This characteristic is useful for recognizing features.

The pooling layer function is used to reduce the size of an image by downsampling it and summarizing the features. The common pooling methods to achieve grouping are average pooling, where the summary is the dominant feature, and maximum pooling by summarizing the strongest feature [53]. Average pooling produces a smooth feature that is useful to extract the most relevant value, such as the color of a surface, where a small variation in isolated points within a region does not affect the overall value. On the other hand, max-pooling extracts high contrast data, such as edges or points.

The problem with a sampling matrix (and an image) in CL is that pixels at and near the edge are sampled less than pixels farther from the edge. This sometimes results in sampling inaccuracy. To prevent this, the kernel filter is padded, with extra rows and columns to allow for more information to be collected from the edge pixels. For two-dimensional data, there are two types of padding: same padding and valid padding. Same padding maintains the sample size at the same as the original matrix; basically, it resamples the image. Valid padding considers all pixels valid, so the model considers the value. This is useful for keeping the information from corner pixels since the simple model considers it invalid due to being less sampled compared to other pixels.

The extracted features compose the feature map that the FCL will use to classify the result. This approach makes CNN a method with fewer computational requirements than the fully connected ANN structure. The CL calculation is formulated in Equation (4):

$$h_{i,j} = \sum_{k=1}^m \sum_{l=1}^m w_{(k,l)} x_{(i+k-1),(j+l-1)} \quad (4)$$

and the pooling layer (max pooling) is stated in Equation (5):

$$h_{i,j} = \max \left\{ x_{(i+k-1),(j+l-1)} \mid \forall l \leq k \text{ and } 1 \leq l \leq m \right\} \quad (5)$$

with fully connected layer h formulated in Equation (6):

$$h = \sum_i w_i x_i \quad (6)$$

where $h_{i,j}$ is the output at point (i, j) on the layer with input x and filter w , and m denotes the width and height of the filter. Non-linear functions are used in CL and FCL to convert negative values to zero, including Sigmoid, Hyperbolic Tangent (*Tanh*), and Rectified Linear Unit (ReLU).

Three-dimensional CNN is a CNN structure whose input is a set of square matrices, $s \times (n \times n)$, so it is a suitable method for image segmentation and classification. In this study, the dataset used is multi-sensor, multi-temporal data derived from SAR and optical images, rainfall data, and ground surface contour data, as shown in Figure 7.

Figure 7. Representation of the multi-temporal 2D dataset into 3D data.

2.5. Proposed Method

Segmentation and classification of flooded areas using 3D CNN for the SAR image dataset and the flood factor consists of three-dimensional dataset segmentation stages using three-dimensional CNN to get initial segmentation results. These results are used to weight neuron connections to perform n -dimensional optimization so that we get the classification of pixels into flood or other categories.

In the three-dimensional CNN shown in Figure 8, several CLs with dimensions $a \times a \times a$ are used to filter the input data to obtain a feature map. The input data used are shown in Table 1. The images are down sampled using a pooling layer by summarizing the features present in the images. In this model, the pooling layer uses max pooling, which summarizes the most dominant value in the sample. To prevent edge and corner pixels from being omitted by the model, valid padding is used on the input layer and the CL. The padding basically left the image unchanged but allows edge and corner pixels to be more sampled as it is now placed further from the edge. Furthermore, the pooling layer measuring $b \times b \times b$ is used to reduce the map, so those neuron connections are formed to compile the information obtained, which is then formed into FCL. FCL stored the different feature values and compiled them into a feature map with two output categories, namely flood pixels and non-flood pixels.

Figure 8. Representation of 3D-CNN process.

Table 1. Input data used for the 3D-CNN.

Type	Source	Resolution/Scale	Acquisition Date
Co-polarization (VV) SAR data	Sentinel-1a	10 m	21 November 2019 to 20 October 2020
Cross-polarization (VH) SAR data	Sentinel-1a	10 m	21 November 2019 to 20 October 2020

The stages carried out in this study began with an inventory of the data used for classification, namely the SAR image dataset. The pre-processing stage is comprised of registering the image data to ensure that the coordinates are consistent between different images. As the images are in RGB TIF format with $r \times c \times 3$ dimensions, they must be converted first into grayscale images, and then samples of sub-images were selected that represent flood and non-flood targets. The data are then divided into training and validation sets. The Feature Learning stage, or training, provides training data for the model to store known flood data. The commonly used proportion between the two sets is 70:30 [54], but we also include 80:20 and 90:10 for comparisons. Training data are used to train 3D CNN [36] in determining the parameters' optimized values. The next stage is to conduct training on the classification by three-dimensional CNN to detect the presence of water surfaces and differentiate them from other surfaces by the variance of the pixel values since dry land and permanent water bodies have consistent values. The ReLU plays a significant part in this phase since flood areas tend to change values, the possibility of dry land changes to a water surface and then back to dry land will result in a negative value. The ReLU rectifies this problem and prevents the neuron with a negative output from being contributed to the network. The Classification stage presents the system to other data for recognizing if there are flood features present in the images using feedback from the results of the Training stage. The overall process in the research is shown in Figure 9.

Figure 9. Workflow of the research.

3. Results and Discussion

The three-dimensional CNN model is trained with two main hyperparameters, namely: epoch, which is the complete iteration of convolution feed-forward before starting over the next iteration; and validation-split, which is the proportion of the training data used for validating the result of the training. In this research, we use the combination of training/validation split of 70/30, 80/20, and 90/10 with epochs of 100 and 150 iterations. The elapsed time and resulting accuracy for each combination are shown in Table 2 and the graphic plot in Figure 10. Accuracy is defined as the percentage of correct predictions for the test data calculated by dividing the number of correct predictions by the number of total predictions, while elapsed time counts the total time needed to perform the training with the corresponding proportion and epochs.

Table 2. The elapsed time, accuracy, and RMSE of the 3D-CNN model.

No. of Epochs	Performance Metrics	Training/Validation Split		
		70/30	80/20	90/10
100	Time	163 min	188 min	140 min
	Accuracy	0.667	0.659	0.685
	RMSE	0.284	0.282	0.203
150	Time	243 min	257 min	302 min
	Accuracy	0.672	0.692	0.674
	RMSE	0.288	0.314	0.296

Figure 10. Graphic plot from the testing and validation of the 3D-CNN model: (a) 100 epochs, 70/30 split; (b) 100 epochs, 80/20 split; (c) 100 epochs, 90/10 split; (d) 150 epochs, 70/30 split; (e) 150 epochs, 80/20 split; and (f) 150 epochs 90/10 split.

During model testing with 100 epochs, the algorithm quickly reaches 100% training accuracy under 40 epochs, and in the first half of the epoch, the testing accuracy increases, but in the second half, it does not rise significantly, being around 0.667; 0.672; and 0.685 for 70/30; 80/20; and 90/10 data split. The overall accuracy achieved by the model is between 0.667 to 0.692 for 100 epochs completed between 140 and 183 min. Root Mean Square Error (RMSE) for 70/30 and 80/20 is around 0.28, while for 90/10 is lower at 0.2, which is consistent with higher accuracy. For 150 epochs, the accuracy of 0.672; 0.692; and 0.674 with RMSE of 0.288; 0.314; and 0.296 for the corresponding data split in 70/30; 80/20; and 90/10 ratios, respectively. The process was completed in 4 h and 3 min. Figure 10 shows that the validation accuracy quickly becomes stable after 20 to 25 epochs while the training accuracy is still increasing until it reaches 100%. This condition indicates that the model was overfitting during testing. Overfitting resulted from a vast set of neural connections, which often reduces the system fitness due to non-common cases included in learning data [55].

We readjusted the model to eliminate and reduce overfitting and then tested it with similar hyperparameters. Overfitting correction is performed by randomly deactivating some neurons on each layer, so they are not used during forward- and back-propagation training. This causes the learning process to spread out connection weights without

focusing on specific neurons. In this research, the deactivation probability is set at 0.5, which means there is an equal chance of each unused neuron in the learning process. Low deactivation probability will not reduce overfitting, while high probability will cause the system to underachieve. Reducing neurons results in a smaller, simpler, and more regulated connection network, which means outlying or widely different results will be disregarded. In this manner, the overall error could be reduced by averaging errors from different connections.

The adjusted model yields the result shown in Table 3 and Figure 11. The results indicate that computation time takes 50 min longer for an 80/20 and 90/10 split, with the resulting accuracy reaching over 0.7 than the previous test. The most significant increase in accuracy is for 150 epochs with a 90/10 split of testing and validation data, which shows an increase of 0.045 for accuracy of 0.719, the lowest RMSE achieved by 70/30 split with 100 epochs with a drop of RMSE value from 0.284 to 0.024. The fastest computing time of 165 min is achieved with 100 epochs and 70/30 split data. This result is consistent that fewer training data corresponds with faster computing but lower accuracy, while a higher percentage of training data took longer but with higher accuracy.

Table 3. The elapsed time, accuracy, and RMSE of the adjusted 3D-CNN model.

No. of Epochs	Performance Metrics	Testing/Validation Split		
		70/30	80/20	90/10
100	Time	165 min	235 min	195 min
	Accuracy	0.691	0.708	0.705
	RMSE	0.024	0.051	0.078
150	Time	172 min	283 min	302 min
	Accuracy	0.699	0.709	0.719
	RMSE	0.082	0.093	0.112

Figure 11. The testing and validation accuracy of the tuned 3D-CNN models: (a) 100 epochs, 70/30 split; (b) 100 epochs, 80/20 split; (c) 100 epochs, 90/10 split; (d) 150 epochs, 70/30 split; (e) 150 epochs, 80/20 split; and (f) 150 epochs, 90/10 split.

Further testing with 140, 145, 155, and 160 epochs to investigate the optimum combination of accuracy with shorter time yields the results shown in Figure 12. Since the testing accuracy is greater with 150 epochs than 100 epochs, we assume that accuracy will improve within 150 ± 10 epochs. Testing with a 70/30 split confirms that as epochs increased from 140 to 160, accuracy gradually improved by 13.6% from 0.567 to 0.703, while computing time increased by 12% from 240 min to 269 min. A similar trend is also observed during testing using an 80/20 data split with an accuracy increase by 13.5% from 0.577 to 0.712 but with a much longer computing time from 243 min to 304 min, representing an increase of 25.1%. The more significant increase is due to the additional time needed to perform more training for the 80/20 than the 70/30. As for the testing with a 90/10 data split, the peak accuracy performance is achieved at 150 epochs with 0.719. Testing with 140, 145, 155, and 160 epochs gives lower results.

In Table 4, the three-dimensional CNN without any combinations with other methods results in higher accuracy than what was achieved by Wang et al. at 0.685 [36]. It is comparable to Grimaldi et al. [11] on open trees flood accuracy at a range of 0.55 to 0.70, which is similar in conditions to flooded areas in Jakarta. Figure 12 shows the flood map of the proposed model compared to the SAR image, as shown in Figure 2, where the model could detect most of the dark areas of the flood while leaving out the similarly dark Jakarta Bay. Compared to the sub-district-level flood map publicly released by the government [2], it is also shown that the flood has occurred in the reported sub-districts. There are discrepancies between the detected and reported areas since the report classifies floods as a whole sub-district coverage.

Table 4. Comparison between models.

KERRYPNX	Methods		
	3D-CNN	CNN-SVM	Fuzzy Logic
Accuracy	0.719	0.685	0.669
Location	Dense Urban	Rural	Rural
Characteristic	Flat	Hill	Flat

(a)

Figure 12. Cont.

Figure 12. Result map compared to (a) 2 January 2020 SAR image, and (b) Jakarta Flood Map released by the government report.

4. Conclusions

In this study, an application of a three-dimensional Convolutional Neural Network for flood mapping is proposed. The deactivation factor minimizes the overfitting problem to reduce the number of neurons and simplify the connections. The research results are that the 3D-CNN method enables the analysis of multi-temporal images for flood detection and classification instead of using multiple image pairs with multiple classification levels. For three combinations of splitting training/test data, the highest overall accuracy of 0.72 was achieved for a split of 90/10 and 150 epochs in 302 min. Regarding computation time, the best performance is achieved with an 80/20 split and 150 epochs with an accuracy of 0.71 in 283 min. Another test with epochs other than 150 showed that accuracy gradually decreases with a 90/10 split, but with a lower training function, the accuracy improves as the number of epochs increases.

Author Contributions: The research was led by D.S. and M.R.; R.A. provided satellite data and insights on satellite data processing; I.R. was responsible for developing methods and analysis. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Doctoral Program Research Indexed Publication Grant of Universitas Indonesia (PUTI Doktor UI) 2020 under Grant NKB-3321/UN2.RST/HKP.05.00/2020.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank Rokhmatuloh and Ardiansyah of the Faculty of Mathematics and Natural Sciences, Universitas Indonesia, for satellite data processing support.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. National Development Planning Agency. *JABODETABEK February 2007 Post-Flood Damage and Loss Estimation Report*; Ministry of National Development Planning: Jakarta, Indonesia, 2018.
2. Jakarta Disaster Mitigation Agency. *Jakarta Historical Flood Map, in Jakarta Historical Flood Map 2013–2017*; DKI Jakarta Disaster Mitigation Agency: Jakarta, Indonesia, 2017.
3. Vanama, V.S.K.; Rao, Y.S. Change Detection Based Flood Mapping of 2015 Flood Event of Chennai City Using Sentinel-1 SAR Images. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2019; pp. 9729–9732.
4. Eini, M.; Kaboli, H.S.; Rashidian, M.; Hedayat, H. Hazard and vulnerability in urban flood risk mapping: Machine learning techniques and considering the role of urban districts. *Int. J. Disaster Risk Reduct.* **2020**, *50*, 101687. [\[CrossRef\]](#)
5. Wu, Z.; Shen, Y.; Wang, H.; Wu, M. Urban flood disaster risk evaluation based on ontology and Bayesian Network. *J. Hydrol.* **2020**, *583*, 124596. [\[CrossRef\]](#)
6. Ciecholewski, M. River channel segmentation in polarimetric SAR images: Watershed transform combined with average contrast maximisation. *Expert Syst. Appl.* **2017**, *82*, 196–215. [\[CrossRef\]](#)
7. Martinis, S.; Kersten, J.; Twele, A. A fully automated TerraSAR-X based flood service. *ISPRS J. Photogramm. Remote Sens.* **2015**, *104*, 203–212. [\[CrossRef\]](#)
8. Schumann, G.J.P.; Moller, D.K. Microwave remote sensing of flood inundation. *Phys. Chem. Earth Parts A/B/C* **2015**, *83–84*, 84–95. [\[CrossRef\]](#)
9. Kwak, Y.; Yun, S.; Iwami, Y. A new approach for rapid urban flood mapping using ALOS-2/PALSAR-2 in 2015 Kinu River Flood, Japan. In Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Ft. Worth, TX, USA, 23–28 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1880–1883.
10. Kwak, Y.; Natsuaki, R.; Yun, S. Effect of Building Orientation on Urban Flood Mapping Using Alos-2 Amplitude Images. In Proceedings of the IGARSS 2018—2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 2350–2353.
11. Grimaldi, S.; Xu, J.; Li, Y.; Pauwels, V.; Walker, J. Flood mapping under vegetation using single SAR acquisitions. *Remote Sens. Environ.* **2020**, *237*, 111582. [\[CrossRef\]](#)
12. Tanguy, M.; Chokmani, K.; Bernier, M.; Poulin, J.; Raymond, S. River flood mapping in urban areas combining Radarsat-2 data and flood return period data. *Remote Sens. Environ.* **2017**, *198*, 442–459. [\[CrossRef\]](#)
13. Rahman, M.R.; Thakur, P.K. Detecting, mapping and analysing of flood water propagation using synthetic aperture radar (SAR) satellite data and GIS: A case study from the Kendrapara District of Orissa State of India. *Egypt. J. Remote Sens. Space Sci.* **2018**, *21*, S37–S41. [\[CrossRef\]](#)
14. Jo, M.; Osmanoglu, B. Rapid Generation of Flood Maps Using Dual-Polarimetric Synthetic Aperture Radar Imagery. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 9764–9767.
15. Pulvirenti, L.; Chini, M.; Pierdicca, N.; Boni, G. Detection of flooded urban areas using sar: An approach based on the coherence of stable scatterers. In Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Fort Worth, TX, USA, 23–28 July 2017; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2017; pp. 5701–5704.
16. Chini, M.; Pulvirenti, L.; Pelich, R.; Pierdicca, N.; Hostache, R.; Matgen, P. Monitoring Urban Floods Using SAR Interferometric Observations. In Proceedings of the IGARSS 2018—2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2018; pp. 8785–8788.
17. Chini, M.; Hostache, R.; Pelich, R.-M.; Matgen, P.; Pulvirenti, L.; Pierdicca, N. Probabilistic Urban Flood Mapping Using SAR Data. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2019; pp. 4643–4645.
18. Refice, A.; D’Addabbo, A.; Pasquariello, G.; Lovergine, F.P.; Capolongo, D.; Manfreda, S. Towards high-precision flood mapping: Multi-temporal SAR/InSAR data, Bayesian inference, and hydrologic modeling. In Proceedings of the 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Milan, Italy, 26–31 July 2015; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2015; pp. 1381–1384.
19. Mason, D.C.; Trigg, M.; Garcia-Pintado, J.; Cloke, H.; Neal, J.; Bates, P. Improving the TanDEM-X Digital Elevation Model for flood modelling using flood extents from Synthetic Aperture Radar images. *Remote Sens. Environ.* **2016**, *173*, 15–28. [\[CrossRef\]](#)
20. Guimarães, U.S.; Narvaes, I.S.; Galo, M.L.B.T.; da Silva, A.Q.; Camargo, P.O. Radargrammetric approaches to the flat relief of the amazon coast using COSMO-SkyMed and TerraSAR-X datasets. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 284–296. [\[CrossRef\]](#)
21. Yang, J.; He, Y.; Caspersen, J. A multi-band watershed segmentation method for individual tree crown delineation from high resolution multispectral aerial image. In Proceedings of the 2014 IEEE Geoscience and Remote Sensing Symposium, Quebec City, QC, Canada, 13–18 July 2014; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2014; pp. 1588–1591.
22. Boni, G.; Pulvirenti, L.; Silvestro, F.; Squicciarino, G.; Pagliara, P.; Onori, R.; Proietti, C.; Candela, L.; Pisani, A.R.; Zoffoli, S. User oriented multidisciplinary approach to flood mapping: The experience of the Italian Civil Protection System. In Proceedings of the 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Milan, Italy, 26–31 July 2015; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2015; pp. 834–837.

23. Xie, J.; Yu, W.; Li, G. An inter-agency collaborative computing framework for fast flood mapping using distributed remote sensing data. In Proceedings of the 2016 Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics) 2016, Tianjin, China, 18–20 July 2016; 2016; pp. 1–5. [\[CrossRef\]](#)
24. Yang, J.; He, Y.; Caspersen, J.P.; Jones, T.A. Delineating Individual Tree Crowns in an Uneven-Aged, Mixed Broadleaf Forest Using Multispectral Watershed Segmentation and Multiscale Fitting. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *10*, 1390–1401. [\[CrossRef\]](#)
25. Chini, M.; Papastergios, A.; Pulvirenti, L.; Pierdicca, N.; Matgen, P.; Parcharidis, I. SAR coherence and polarimetric information for improving flood mapping. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2016; pp. 7577–7580.
26. Duan, Y.; Fang, L.; Licheng, J.; Peng, Z.; Lu, Z. SAR Image segmentation based on convolutional-wavelet neural network and Markov random field. *Pattern Recognit.* **2017**, *64*, 255–267. [\[CrossRef\]](#)
27. Burle, S. FloodMap.net. Available online: <https://www.floodmap.net/Elevation/ElevationMap/?gi=1642911> (accessed on 1 August 2020).
28. Pelich, R.; Chini, M.; Hostache, R.; Matgen, P.; Delgado, J.M.; Sabatino, G. Towards a global flood frequency map from SAR data. In Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Fort Worth, TX, USA, 23–28 July 2017; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2017; pp. 4024–4027.
29. Lee, J.-Y.; Kim, J.-S. Detecting Areas Vulnerable to Flooding Using Hydrological-Topographic Factors and Logistic Regression. *Appl. Sci.* **2021**, *11*, 5652. [\[CrossRef\]](#)
30. Sidek, L.M.; Chua, L.H.C.; Azizi, A.S.M.; Basri, H.; Jaafar, A.S.; Moon, W.C. Application of PCSWMM for the 1-D and 1-D–2-D Modeling of Urban Flooding in Damansara Catchment, Malaysia. *Appl. Sci.* **2021**, *11*, 9300. [\[CrossRef\]](#)
31. Cian, F.; Marconcini, M.; Ceccato, P. Normalized Difference Flood Index for rapid flood mapping: Taking advantage of EO big data. *Remote Sens. Environ.* **2018**, *209*, 712–730. [\[CrossRef\]](#)
32. Cian, F.; Marconcini, M.; Ceccato, P.; Giupponi, C. Flood Depth Estimation by Means of High-Resolution SAR Images and LiDAR_Data_ResearchGate. Available online: https://www.researchgate.net/publication/326067701_Flood_depth_estimation_by_means_of_high-resolution_SAR_images_and_LiDAR_data (accessed on 16 September 2020).
33. Iglesias, R.; Garcia-Boadas, E.; Vicente-Guijalba, F.; Centolanza, G.; Duro, J. Towards Unsupervised Flood Mapping Generation Using Automatic Thresholding and Classification Approaches. In Proceedings of the IGARSS 2018—2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 4927–4930.
34. Dasgupta, A.; Grimaldi, S.; Ramsankaran, R.; Pauwels, V.; Walker, J. Towards operational SAR-based flood mapping using neuro-fuzzy texture-based approaches. *Remote Sens. Environ.* **2018**, *215*, 313–329. [\[CrossRef\]](#)
35. Li, L.; Chen, Y.; Xu, T.; Shi, K.; Huang, C.; Liu, R.; Lu, B.; Meng, L. Enhanced Super-Resolution Mapping of Urban Floods Based on the Fusion of Support Vector Machine and General Regression Neural Network. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1269–1273. [\[CrossRef\]](#)
36. Wang, Y.; Fang, Z.; Hong, H.; Peng, L. Flood susceptibility mapping using convolutional neural network frameworks. *J. Hydrol.* **2020**, *582*, 124482. [\[CrossRef\]](#)
37. Li, Y.; Martinis, S.; Wieland, M. Urban flood mapping with an active self-learning convolutional neural network based on TerraSAR-X intensity and interferometric coherence. *ISPRS J. Photogramm. Remote Sens.* **2019**, *152*, 178–191. [\[CrossRef\]](#)
38. Sameen, M.I.; Pradhan, B. Landslide Detection Using Residual Networks and the Fusion of Spectral and Topographic Information. *IEEE Access* **2019**, *7*, 114363–114373. [\[CrossRef\]](#)
39. Bui, Q.-T.; Nguyen, Q.-H.; Nguyen, X.L.; Pham, V.D.; Nguyen, H.D.; Pham, V.-M. Verification of novel integrations of swarm intelligence algorithms into deep learning neural network for flood susceptibility mapping. *J. Hydrol.* **2020**, *581*, 124379. [\[CrossRef\]](#)
40. Bui, D.T.; Hoang, N.-D.; Martínez-Álvarez, F.; Ngo, P.-T.T.; Hoa, P.V.; Pham, T.D.; Samui, P.; Costache, R. A novel deep learning neural network approach for predicting flash flood susceptibility: A case study at a high frequency tropical storm area. *Sci. Total Environ.* **2020**, *701*, 134413. [\[CrossRef\]](#)
41. Shen, X.; Anagnostou, E.N.; Allen, G.H.; Brakenridge, G.R.; Kettner, A. Near-real-time non-obstructed flood inundation mapping using synthetic aperture radar. *Remote Sens. Environ.* **2019**, *221*, 302–315. [\[CrossRef\]](#)
42. Ali, S.; Arief, R.; Dyatmika, H.S.; Maulana, R.; Rahayu, M.I.; Sondita, A.; Setiyoko, A.; Maryanto, A.; Budiono, M.E.; Sudiana, D. Digital Elevation Model (DEM) Generation with Repeat Pass Interferometry Method Using TerraSAR-X/Tandem-X (Study Case in Bandung Area). In *IOP Conference Series: Earth and Environmental Science*; IOP Publishing: Bristol, UK, 2019; Volume 280, p. 012019.
43. Landuyt, L.; Van Wesemael, A.; Schumann, G.J.-P.; Hostache, R.; Verhoest, N.E.C.; Van Coillie, F.M.B. Flood Mapping Based on Synthetic Aperture Radar: An Assessment of Established Approaches. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 722–739. [\[CrossRef\]](#)
44. Felegari, S.; Sharifi, A.; Moravej, K.; Amin, M.; Golchin, A.; Muzirafuti, A.; Tariq, A.; Zhao, N. Integration of Sentinel 1 and Sentinel 2 Satellite Images for Crop Mapping. *Appl. Sci.* **2021**, *11*, 10104. [\[CrossRef\]](#)
45. Schubert, A.; Miranda, N.; Geudtner, D.; Small, D. Sentinel-1A/B Combined Product Geolocation Accuracy. *Remote Sens.* **2017**, *9*, 607. [\[CrossRef\]](#)
46. Liang, J.; Liu, D. A local thresholding approach to flood water delineation using Sentinel-1 SAR imagery. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 53–62. [\[CrossRef\]](#)

47. Chini, M.; Hostache, R.; Giustarini, L.; Matgen, P. A Hierarchical Split-Based Approach for Parametric Thresholding of SAR Images: Flood Inundation as a Test Case. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 6975–6988. [[CrossRef](#)]
48. Pulvirenti, L.; Chini, M.; Pierdicca, N.; Boni, G. Flood Detection in Urban Areas: Analysis of Time Series of Coherence Data in Stable Scatterers. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2019; pp. 9745–9747.
49. Pham, V.D.; Nguyen, Q.-H.; Nguyen, H.-D.; Pham, V.-M.; Vu, V.M.; Bui, Q.-T. Convolutional Neural Network—Optimized Moth Flame Algorithm for Shallow Landslide Susceptible Analysis. *IEEE Access* **2020**, *8*, 32727–32736. [[CrossRef](#)]
50. Munawar, H.S.; Ullah, F.; Qayyum, S.; Heravi, A. Application of Deep Learning on UAV-Based Aerial Images for Flood Detection. *Smart Cities* **2021**, *4*, 1220–1242. [[CrossRef](#)]
51. Wang, Y.; Hong, H.; Chen, W.; Li, S.; Panahi, M.; Khosravi, K.; Shirzadi, A.; Shahabi, H.; Panahi, S.; Costache, R. Flood susceptibility mapping in Dingnan County (China) using adaptive neuro-fuzzy inference system with biogeography based optimization and imperialistic competitive algorithm. *J. Environ. Manag.* **2019**, *247*, 712–729. [[CrossRef](#)] [[PubMed](#)]
52. Muhadi, N.A.; Abdullah, A.F.; Bejo, S.K.; Mahadi, M.R.; Mijic, A. Deep Learning Semantic Segmentation for Water Level Estimation Using Surveillance Camera. *Appl. Sci.* **2021**, *11*, 9691. [[CrossRef](#)]
53. Scherer, D.; Müller, A.; Behnke, S. Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. In Proceedings of the International Conference on Artificial Neural Networks, Thessaloniki, Greece, 15–18 September 2010.
54. James, G.; Witten, D.; Hastie, T.; Tibshirani, R. *An Introduction to Statistical Learning*, 7th ed.; Springer: New York, NY, USA, 2017.
55. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.