Evolution in Electrical and Electronic Engineering Vol. 4 No. 1 (2023) 729-736

EEEE

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/eeee
e-ISSN : 2756-8458

Deep Learning Method for Flood Detection

Muhammad Zulhelmi Muhamad Zuraidi[1], Audrey Huong[1]*

[1]Department of Electronic Engineering, Faculty of Electrical and Electronic Engineering,
Universiti Tun Hussein Onn Malaysia, Batu Pahat, 86400, MALAYSIA.

*Corresponding Author Designation

DOI: https://doi.org/10.30880/eeee.2023.04.01.087
Received 04 February 2023; Accepted 28 March 2023; Available online 30 April 2023

Abstract: Floods, one of the problems, faced by low-land communities can occur practically everywhere. The impacts of the flood can be devastating, especially when it fills up an area in a short amount of time. However, some of the existing systems can be expensive and complex in their implementation in a real-world setting. The aim of this study is to propose a deep learning system for early flood detection using AlexNet. For this purpose, related images would undergo an image processing technique before they are used for model training. The pre-trained model is used to extract important information from these images. Based on the results, all the training accuracy with different parameters reached 100% except for hyperparameter 20 batch size and 10 epochs. while for the validation accuracy, the average value for 5 epochs is 95.74% and for 10 epochs is 95.03% The confusion matrix for binary classification has also been implemented for this system which the parameters are accuracy, specificity, sensitivity, and precision. The best result was the system with 30 batch size0 epochs and 157 sample images which the accuracy value is 0.9574, precision and sensitivity have the same value which is 0.9677 and the specificity is 0.9375. It is hypothesized that the epochs and mini-batch size affected the time taken to finish the classification process and the accuracy of the system.

Keywords: Flood, Deep Learning, Classification

1. Introduction

Natural disasters like hurricanes, earthquakes, and floods have caused havoc on the environment, the economy, and people's lives. The increased flood risk has a significant influence on the current and future economy, particularly in developing nations with inadequate resources for flood prevention mitigation. Furthermore, because the collecting data is also quite expensive, these monies are sometimes allocated without sufficient and acceptable information and high risk and flood-prone locations [1]. Flood, one of nature's most destructive tragedies, has destroyed houses, crops, and has been the focus of several studies [2]. Flood also is one of the mother natures that happen frequently in the entire world especially in Malaysia. At least once in a year, some of the regions with low-lying area are likely to receive flood twice or even more in a year. In some circumstances and during the monsoon seasons, flash may happen leading loss of human life and damage of settlements.

*Corresponding author: audrey@uthm.edu.my

The economic life and loss caused by the flood disaster is because the citizens and manufactures have not received warning on the disaster risk, and the government has not received information with urgency. It is imperative that government should identify efficient methods to overcome this problem to reduce the losses. However, some of the existing systems can be expensive and complex in the implementation when testing the simulation network in a real-world setting. The latest technology to detect flood is CubeSat which is mission that use PhiSat-1 that known as ϕ-Sat-1 from the European Space Agency (ESA) that uses Artificial Intelligence (AI) for Earth observation which enables the hardware performance of onboard processing by including a power-constrained machine learning accelerator and the software to execute customized applications, intends to allow the presentation of this concept. The PhiSat-1 will illustrate a flood segmentation algorithm that operating effectively on the accelerator aboard the PhiSat-1, generates flood masks to be transmitted rather than the raw photos [3]. The Phisat-1 was very expensive because a CubeSat's price is determined by its mass, size, and the launch vehicle it uses. It might be anything between 10,000 USD and 500,000 USD on average. The sizes of CubeSats range from 0.5U to 27U, with 1U being 10cm x 10cm x 10cm.

On the other hand, the machine learning and deep learning have made significant advances in image classification. However, there are just a handful approaches that use deep learning to detect and manage real-world disasters. This is primarily owing to the domain's scarcity of annotated data. Existing work often pulls data from the we or social media and manually annotates it. Hence, the variability of images in such datasets may not be enough to construct a robust model that can be employed in a variety of real-world scenarios. Nonetheless, there were lot of flood photographs on social media with media with the timeline were taking during flood seasons. Most of these images are of clear and noise-free [4]. It is the objective of this study to propose an effective flood detection system for early flood warning.

2. Methodology

The flow diagram on the development of this project is shown in Figure 1. To kick-start of this project, first, data were collected from the images been taken from the database which is Kaggle [5] and self-collect. All these photographs are then saved in the computer database which the address of the file stated in the deep learning coding. Then, all the collected photographs are grouped into two classes: Flood and non-flood and been organized into folders which to do deep learning and neural network for the initial recognition and identification procedure after going through the entire training set process. The design for this CNN model is the use of AlexNet as the type of deep learning with initial learn rate is 0.001. A 1e-3 first learning rate is typically regarded as a reasonably high initial learning rate, whilst a 1e-4 initial learning rate is typically seen as a relatively low initial learning rate. For the mini batch size, the values that have been chosen is 5 and 10 because too high and too small values might result in excessive noise levels. The value of epoch is 10, 20, 30 is to establish the ideal number of epochs for a particular problem which to combine early halting and cross validation procedures and for the data splitting, 70% of the image used for training and 30% for validation.

2.1 Software

In this project, to implement and develop the coding for picture classification of flood detection, MATLAB was chosen because it offers methods and functions for deeper learning and machine learning in addition to a variety of fields, such as robots, computer vision, and data processing, to input into these algorithms [5] and can produce good performance especially for this project such as image classification.

2.2 Data Collection

The use of public resources, such as Kaggle webpage, is very useful to collect flood image because it contains large dataset of flood image. The result from exploring Kaggle are about 10,000 of flood images in the database and they have been downloaded. For the non-flood images, the method that will be used is self-collect image based on the location experienced with flood. The downloaded image of

after the search of flood images in the Kaggle website. The output image has about 10,000 images in total and comes with various and almost similar in some output images from the dataset. After selection process had been done, there were total 178 image been chosen for flood image and for non-flood image contains 75 images from the self-collect. For this system, the ratio of data split that been set is 70 percent for training and 30 percent for validation.

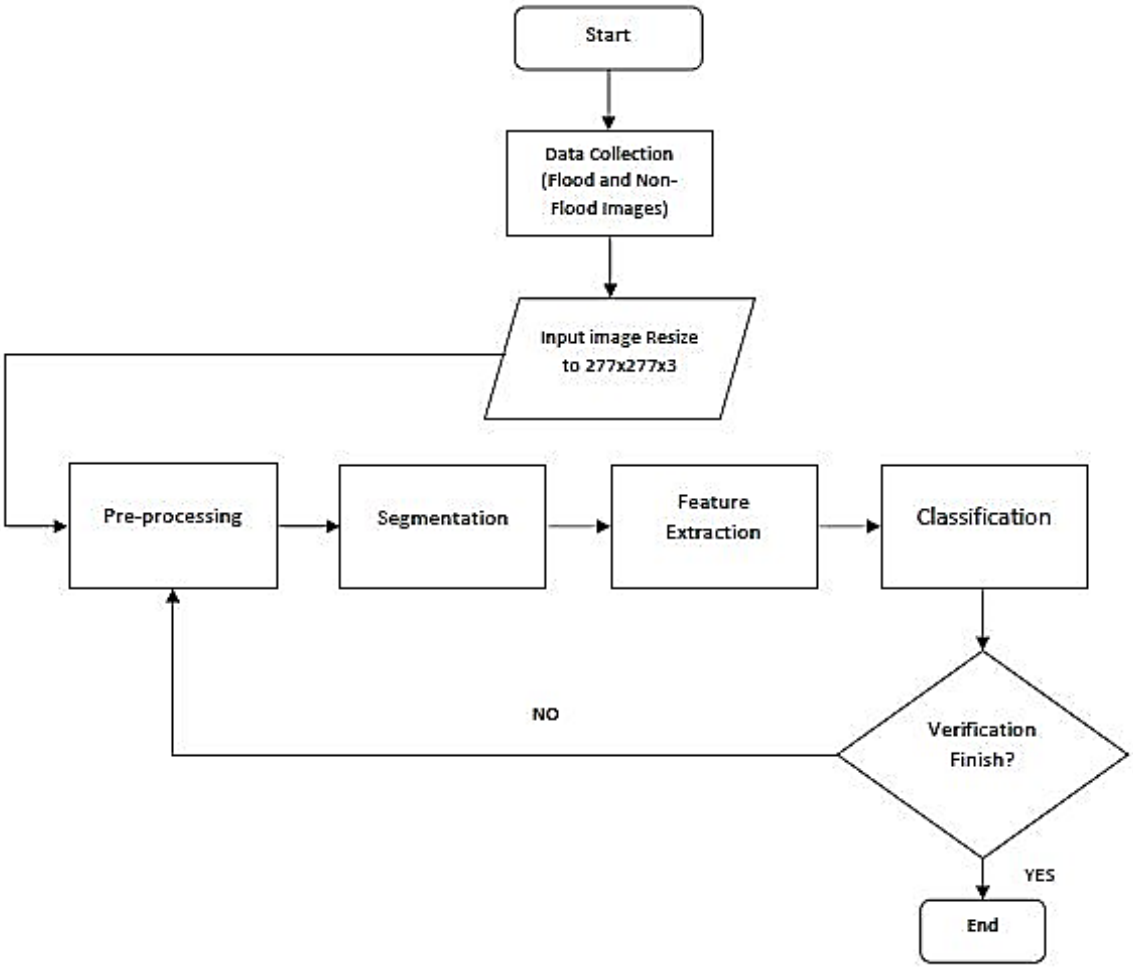


Figure 1: Flowchart of the project

3. Result and Discussion

3.1 Result

The results and discussion section presented data and analysis obtained from the simulation of the system. For the system with 5 epochs, there were three types of different batch size were implemented to the system which is 10, 20 and 30. Table 1 shows the number of batch size 10 required 871 seconds to complete the process while for 20 batch sizes required 1499 second and for 30 batch sizes, it required 2069 second. While for the system with 10 epochs, the parameters also same as previous batch. Referring to the tables 1, for the batch size 10, the time taken to complete the process is 430 second with 93.62% accuracy while for 20 batch size, it required 981 second with 95.74% accuracy and for 30 batch size, the accuracy is 95.74% with 1327 second to complete.

Table 1: Accuracy and Time Taken for Every Batch and Epoch

Number of Sample Image	Training		Validation Accuracy (%)	Training Accuracy (%)	Time taken to complete training(s)
	Batch Size	Epochs			
158	10	5	93.62	100	871
158	20	5	100	100	1499
158	30	5	93.62	100	2069
158	10	10	93.62	100	430
158	20	10	95.74	90	981
158	30	10	95.74	100	1327

There are few general image characteristic types available including specificity, sensitivity, accuracy, and precision measures given in Eq. 1- 4. The most common quantitative measures of diagnostic accuracy. The most common quantitative measures of diagnostic quality are specificity, sensitivity, and precision.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \qquad Eq.1$$

$$Sensitivity = \frac{TP}{TP + FP} \qquad Eq.2$$

$$Specificity = \frac{TN}{TN + FP} \qquad Eq.\ 3$$

$$Precision = \frac{TP}{TP + FP} \qquad Eq.4$$

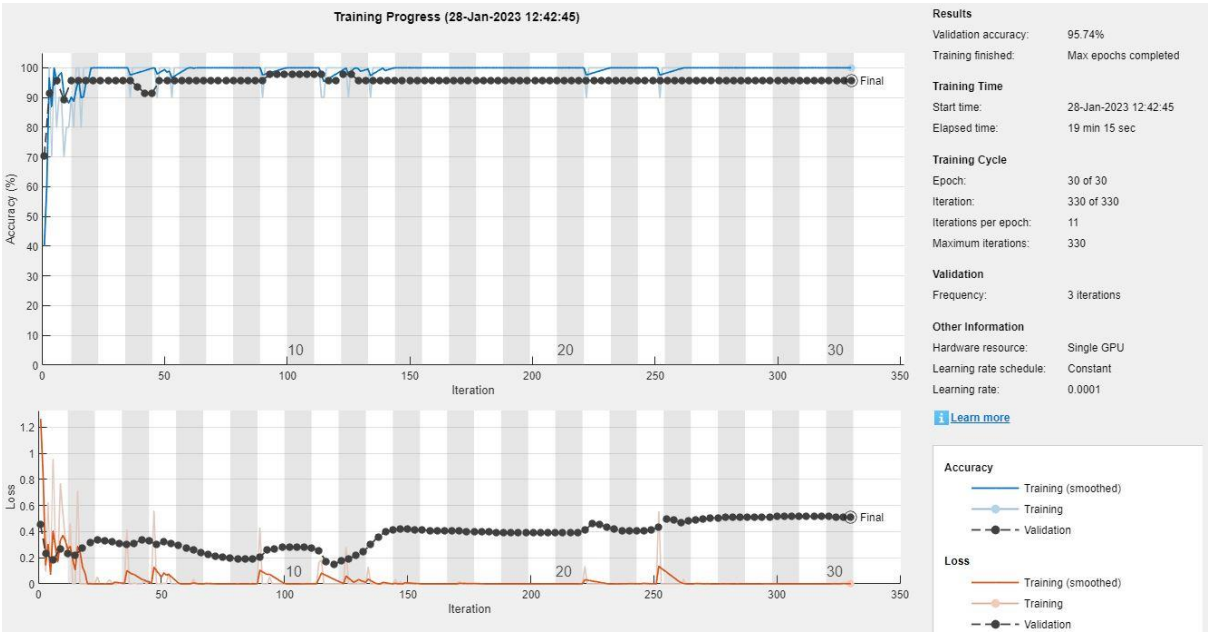
Accuracy measures the proportion of true predictions (both true positives and true negatives) among all predictions. A Sensitivity measures the proportion of true positives (i.e., cases where the model correctly predicted that the outcome would be positive) among all positives (i.e., all cases where the outcome is actually positive). A specificity estimates the share of all negatives that are true negatives, or instances when the model accurately anticipated that the outcome would be negative (i.e., all cases where the outcome is negative). Lastly, precision measures the proportion of true positives among all cases where the model predicted the outcome to be positive.

Table 2 shows the result from the binary classification of the system in terms of accuracy, precision, sensitivity, and specificity. The training with 20 batch size and 5 epochs demonstrated the best performance for 5 epochs because of the value of accuracy, precision, sensitivity, and specificity are equal to 1 which no misclassification at all. For the system with 30 epochs, the best performance demonstrated by 30 batch size because the value of each parameter for binary majority higher than other two batch size which the accuracy=0.96, precision=0.97, sensitivity=0.97 and specificity=0.93.

Table 2: Accuracy, Precision, Sensitivity and Specificity for different batch size and epochs.

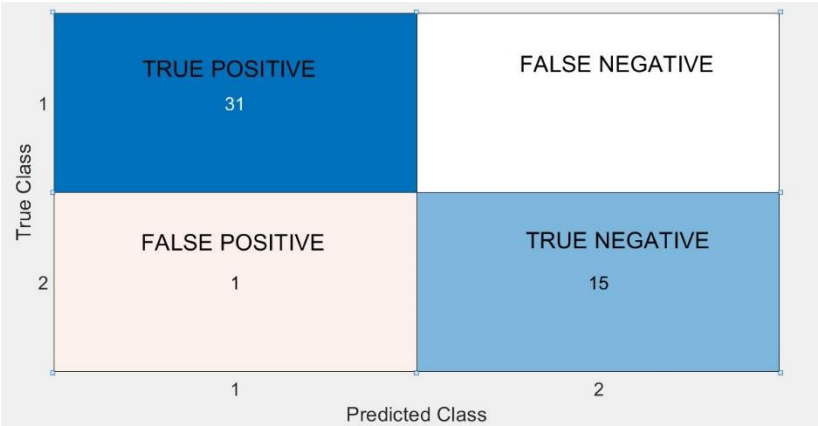
Training		Accuracy	Precision	Sensitivity	Specificity
Batch Size	**Epochs**				
10	5	0.96	0.94	0.97	0.86
20	5	1	1	1	1
30	5	0.93	0.91	1	0.81
10	10	0.93	0.94	0.97	0.86
20	10	0.96	0.94	1	0.86
30	10	0.96	0.97	0.97	0.93

Figure 2 shows a training system with output result of accuracy and image of flood and non-flood that had been classified. For the Figure 2 (a), the number epochs that be used is 30 which the number of iterations is only 330. The time taken to complete all the iteration is 1155 seconds. Referring to Figure 2(b), the picture with the circle shape where the image had been classified as flood while for picture with no shape classified as non-flood. In addition, there was a picture with the red rectangular with addition of three arrows was the indicator to show that image had been misclassified. Other than that, Figure 2(c) shows the result of matrix confusion of binary classification which the value of True Positive=30, False Positive=1, False Negative=1 and True Negative=16.



(a)

(b)



(c)

Figure 2: The result from the classification process using AlexNet, (a) Validation accuracy, (b) Image had been classified as Non-Flood and Flood, (c) Confusion matrix for binary classification

3.2 Discussion

Small batch sizes can have a negative impact on performance since they do not give the model enough data to appropriately update its weights. Additionally, when working with tiny batch sizes, the model can end up being more vulnerable to data's random noise, which might lead to a less-than-ideal outcome. The model does not have enough time to learn the underlying pattern in the data, hence a minimal number of epochs may also result in underfitting. To enhance the model's performance, more epochs and higher batch sizes are advised. In addition, the important thing that need to be always remember is that the appropriate batch size and epoch count will vary depending on the dataset and model setup.

A higher batch size can provide the model more data to update its parameters, which could result in a model that is more accurate. There is a transfer between batch size and computing resource because employing a larger batch size also calls for more memory and processing. The appropriate batch size and number of epochs will vary depending on the dataset and model architecture, although using bigger batch sizes and more epochs is advised to boost the model's performance. Additionally, it is important to prevent overfitting by using strategies like early stopping, cross-validation, and regularization. Table

1 shows the increased number of batch size affected the time taken to complete the process even though the time taken were increased. The percentage of accuracy also increase and with 20 batch size of 5 epoch, 100% of validation accuracy can be obtained, which shows the best accuracy among two others batch size. So, the best accuracy is batch size 30 because it shows the highest frequency and highest training accuracy.

Misclassification happened when a deep learning model had incorrectly predicted the class label for and input. There were some factors that lead to misclassification of image as example in Figure 2(b) which one of the images shown were misclassified. One of the reason is overfitting that occurs when a model grows very complex and matches the test data too closely, leading to poor performance on fresh, untrained data. Other than that, imbalanced dataset which occurs when the quantity of samples in one class is significantly more than in another, can have an impact on deep learning models. As a result, there may be misclassification due to the model's bias toward the dominant class.

Based on Table 2, the performance of batch size and epoch that provided low performance possibilities happened because of faster training time that led to inaccurate results. A classifier with a high sensitivity abled to properly identify many positive examples and have a low proportion of false negatives. Other than that, high specificity will have a low rate of false positives and be able to accurately identify many negative cases. A high accuracy indicates that the classifier properly detects a high percentage of the positive examples among those images has classified as positive, making a low number of false positive errors. The conclusion can be made is in the classification system, the hyperparameters chosen was important to have a balance value between specificity and sensitivity because the indicator has ability to correctly identify positive exampled while for negative examples, the value of sensitivity and specificity needed to balance so that the image can identified correctly.

4. Conclusion

The project on flood detection using deep learning is to classify the flood from the image dataset had been collected to categorize it its flood or non-flood. These images were trained by using Convolution Neural Network (CNN) for the classification process. In addition, all the output result had been analyzed and analysis to compare the difference and get the most accurate result. The neural network has shown their ability to process the image with high test accuracy even though there was misclassification image. Other than that, future work includes adding Graphical User Interface (GUI) to make the implementation more user friendly and to insert alert system to increase the effectiveness of flood detection system. The enlargement of well annotated dataset also can help to achieve better accuracy so that the system can learn and have potential in real application of flood detections. Lastly, a system with fast computing would also help to speed up the training and interference process.

Acknowledgement

The authors would like to thank the Faculty of Electrical and Electronics Engineering, Universiti Tun Hussein Onn Malaysia for the support.

References

[1] Chandrakant Chawan, A., K Kakade, V., & K Jadhav, J. (2020). Automatic Detection of Flood Using Remote Sensing Images. *Journal of Information Technology and Digital World*, *02*(01), 11–26. https://doi.org/10.36548/jitdw.2020.1.002

[2] Menon, R., Professor, A., Simon, S., Shaju, R., & Students, S. (2021). Detection of Flood Images using Different Classifiers. *International Journal of Innovative Science and Research Technology*, *6*(6). Retrieved from https://ijisrt.com/assets/upload/files/IJISRT21JUN524.pdf

[3] Mateo-Garcia, G., Veitch-Michaelis, J., Smith, L., Oprea, S. V., Schumann, G., Gal, Y., … Backes, D. (2021). Towards global flood mapping onboard low cost satellites with machine learning. *Scientific Reports*, *11*(1). https://doi.org/10.1038/s41598-021-86650-z

[4] S. Pouyanfar et al., "Unconstrained Flood Event Detection Using Adversarial Data Augmentation," 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 2019, pp. 155-159, doi: 10.1109/ICIP.2019.8802923.

[5] HHR@Clemson. (2023). *FloodDBS: Flood Image DataBase System*. Kaggle.com. https://www.kaggle.com/datasets/hhrclemson/flooding-image-dataset