ISSN 2278-3091

Volume 9 No.2, March - April 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse205922020.pdf

https://doi.org/10.30534/ijatcse/2020/205922020

An Inference Approach of Flood Level Detection and Alert System: Flood-Induced Simulated Environment

Alvin Sarraga Alon[1], Jennalyn N. Mindoro[2], Cherry D. Casuat[3], Mon Arjay F. Malbog[4], Julie Ann B. Susa[5]

[1]Technological Institute of the Philippines, Manila Philippines, aalon.cpe@tip.edu.ph

[2]Technological Institute of the Philippines, Manila Philippines, jnicolas.cpe@tip.edu.ph

[3]Technological Institute of the Philippines, Manila Philippines, ccasuat.cpe@tip.edu.ph

[4]Technological Institute of the Philippines, Manila Philippines, malbog.monarjay@gmail.com

[5]Technological Institute of the Philippines, Manila Philippines, jsusa.cpe@tip.edu.ph

ABSTRACT

Many kinds of research focused on the flood detection and monitoring, flood management, flood risk management and flood forecasting in urban areas, wherein a large number of populations lies chaos in mobility is high. Owing to natural disasters, flooding in these regions can lead to an increase in mortality rates. This project is primarily focused on the detection of a flood by installing a flood detector device with a camera beside the bridge column. The camera is facing the three lines with different colors. If one of the colors was tempered by the river water, the device will send an alarm to the community that the water level in the river is high. This aims to alert the community and the authorities to be aware and be ready for the approaching flood. Flood-Level Detection and Alert System proved 87.1%, 73.6%, and 95.69% testing accuracy of Green, Blue, and Red respectively. Overall, the accuracy of the whole system produced 85.46%.

Key words : Deep Learning, Flood Detection, Machine Vision, MobileNet SSD, Raspberry Pi 4B.

1. INTRODUCTION

Floods are among the most damaging natural hazards which regularly take several lives [1]. The scale of its damage is increased by population growth, acceleration of the urbanization cycle, rapid industrialization, especially unreasonable land, water supplies and forest degradation [2]. It has been recorded that flooding has impacted 3,046,770,000 people worldwide and 1,674,670,000 people, from 1900 to 2007, in China alone [3]. Regional flood control is a dynamic and exhausting concern and flood issues have drawn wide science attention [4]. Several methods of study focused on flood forecasting, emergency control, flood risk assessment and flood prediction [5].

In general, flood indicators are used for interpreting flooding from a management perspective. The metrics are built to understand the intensities of floods including their severity and clustering characteristics. In general, peak flood discharge or peak flood level is used to characterize the flood severity [1]. Based on [6], this paper was established with the goal of mitigating harm and damages incurred by flooding in urban centers, by means of a cheap flood alarm system for authorities and the public. The system tracks the water level of a runway or river and activates warning warnings through telephone calls and SMS messages to the authorities according to the danger of the level. There are light beacons for the community that identify the river's condition. The data is processed via the Unidos Server in the cloud computing and can be accessed in real time. These are all the same with [7]-[9] who also monitor the flood levels

In urban areas, where a large number of populations lies chaos in mobility is high, the flooding of these regions will lead to higher mortality levels due to natural catastrophes [10]. Highways and bridges link both the urban and rural areas. Bridges constructed over dams, seas and other channels are typically more vulnerable to catastrophes such as flooding and tsunamis. An early detection and alarm program is designed to inform people about the harm done by such disasters. The flood / overflow spectrum is monitored using the video streaming mechanism and if it reaches a certain level, alert information is received. The information regarding the flood is forwarded to the local population. It should enable them take the necessary steps to cope with the tragedy that has happened. The same study has been used by [11] – [13] that uses video as flood monitoring and by [14] – [18] using the power of machine vision deep learning.

Floods will last just a few hours and inflict serious harm and a potential life danger. Flood prediction includes comprehensive awareness of infiltration and the patterns of runoff, which is typically obtained locally. Compared to more technically sophisticated developing countries, This offers details regarding the location of the flood area and the date and period of the flood [14]. This device will minimize the negative impacts, often incidents caused by the collapsing of the bridge due to flooding.

The study primarily focused on the detection of the flood by installing a flood detector device with a camera beside the bridge's column. The camera is facing three lines with a different color if the red color was tempered by the river water the device will send an alarm to the community that the water level in the river is high. It aims to alert the community and the authorities to be aware and be ready for the approaching flood.

The objective of the study is to avoid great losses and prejudice in a community by developing a device that can give alert to authorities and also the people, to know the water level of the river, from low level to high and if it is time for them to evacuate. The project will be used by the local government in flood-prone areas that will enable them to easily detect the flood when there is a typhoon. The scope and limitation of the study would be determining the water level of the river.

2. METHODOLOGY

The study's primary function is to detect the flood and to produce an alarm according to the level it reaches. Figure 1 shows the block diagram of the study.

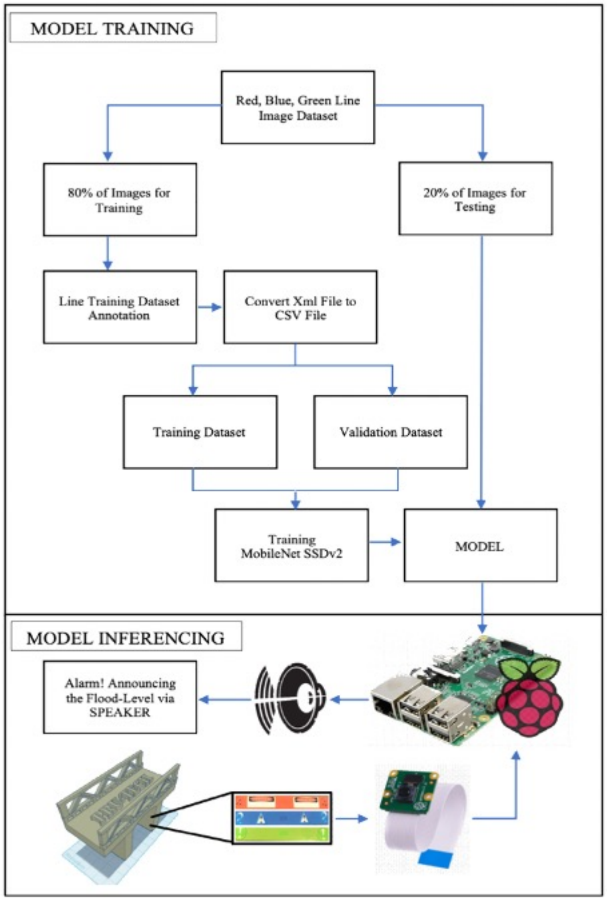
MODEL TRAINING
Red, Blue, Green Line Image Dataset
80% of Images for Training
20% of Images for Testing
Line Training Dataset Annotation
Convert Xml File to CSV File
Training Dataset
Validation Dataset
Training MobileNet SSDv2
MODEL
MODEL INFERENCING
Alarm! Announcing the Flood-Level via SPEAKER

Figure 1: Flood-Level Detection and Alert System

2.1 Red, Blue, Green Line Dataset

The dataset was composed of three colors: Red, Green, and Blue. These three colors indicate what would be the height of the water in that particular bridge as shown in Figure 2. Each color corresponds to a certain water level; Blue has the lowest water level or normal, Green means the water is rising and a flood is possible and Red indicates that there will be a flood and the water level is high. The images were gathered through a mobile phone by printing the three different colors to different bond papers. Throughout, the number of datasets reached 300 images.

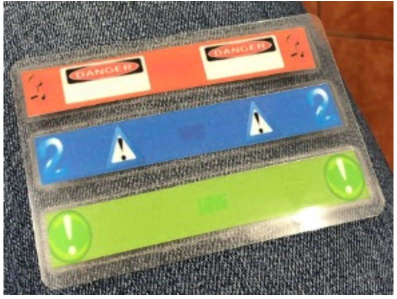

Figure 2: Red, Blue, Green Line Dataset

2.2 20% of images for Testing

The study used 60 random images from the data set to the testing folder because it is the best practice to have a high accuracy rate to put the 20 percent of random images into the test folder. If you have a limited dataset, it is not recommended to exceed 20 percent.

2.3 80% of Images for Training

Since 20 percent of the dataset goes to the test folder. Therefore the remaining 240 random images will go to the training folder. Putting 80 percent of your dataset into your train folder will give you high accuracy. It is not recommended to go way beyond 80 percent. Figure 3 shows the images used for the training.

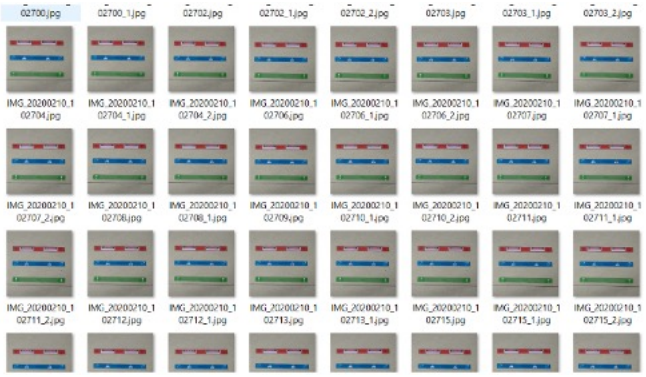
02700.jpg
02700_1.jpg
02702.jpg
02702_1.jpg
02702_2.jpg
02703.jpg
02703_1.jpg
02703_2.jpg
IMG_20200210_102704.jpg
IMG_20200210_102704_1.jpg
IMG_20200210_102704_2.jpg
IMG_20200210_102706.jpg
IMG_20200210_102706_1.jpg
IMG_20200210_102706_2.jpg
IMG_20200210_102707.jpg
IMG_20200210_102707_1.jpg
IMG_20200210_102707_2.jpg
IMG_20200210_102708.jpg
IMG_20200210_102708_1.jpg
IMG_20200210_102709.jpg
IMG_20200210_102710_1.jpg
IMG_20200210_102710_2.jpg
IMG_20200210_102711.jpg
IMG_20200210_102711_1.jpg
IMG_20200210_102711_2.jpg
IMG_20200210_102712.jpg
IMG_20200210_102712_1.jpg
IMG_20200210_102713.jpg
IMG_20200210_102713_1.jpg
IMG_20200210_102715.jpg
IMG_20200210_102715_1.jpg
IMG_20200210_102715_2.jpg

Figure 3: Images for Training

2.4 Annotating Images

After splitting the dataset and used 20% of images to the test folder and 80% of images to the training folder, the next step is to annotate and label the image. The study used the software LabelImg to put labels to the images that gathered as shown in Fig. 4. Deep learning requires large volumes of data to train, validate, and test. A model of deep learning learns to find patterns in the data that are being fed to it. The annotated image data needs to be highly accurate, so the model can learn and forecast the true scenarios accordingly.

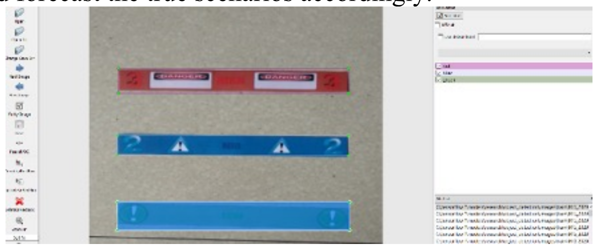

Figure 4: Annotation Images Dataset

2.5 Convert XML file to CSV file

After labeling and annotating the images the LableImg generated an XML file then will be converted the XML file to CSV file. One of the requirements upon making a custom dataset is having the XML file converted into a CSV file. This serves as the preparation for the training of customs datasets. Having many datasets improve its accuracy through its detection.

2.6 Training Dataset & Testing Dataset

After the process of labeling and annotating the images and converting it to CSV file the dataset is now ready for testing. The study used the MobileNet SSD V2 to generate the model.

2.7 Training using MobileNet SSD v2

The study used the MobileNet SSD V2 because it is the one recommended to be used in generating a model in a Raspberry Pi.

- Single Shot Multibox Detector — SSD is an object detector that can be used on real-time video, easily enough.
- An object detector can identify various types of objects in the image. The detections are represented by boundary boxes, and the model also predicts a class for each bounding box. SSD is used in many variants. The study used MobileNet V2 as the backbone, and for the SSD layers, also known as SSDLite, there are depth-separable convolutions.
- SSD: Single Shot MultiBox Detector — The study used SSD, and discrete the bounding box performance space into a collection of default boxes over specific aspect ratios and sizes per feature map location. Compared to methods involving object proposals, the SSD model is simple, as it fully eliminates the generation of proposals and corresponding pixels or the resampling stage of functions and encapsulates all computation in single network.

2.8 Model

From data gathering, image labeling and annotation the next step is training. The 80% image for training and 20% images for testing has been trained using the MobileNet SSD V2, then after some time depending on your computer's capacity the model will be the final output of the training. The generated model has been used for detecting the Red, Blue and Green Color for the alarm system.

2.9 Detect Line through Pi-Camera

Once the model is done and ready to use, the file has been used to raspberry and this is where various tests happen, after the testing of the model and made sure that is working, the pi camera used as a medium for detecting the line and made sure that is 90% accurate. The pi-camera has been mounted in front of the bridge to take real-time images and has been used to automatically detect lines according to the machine vision algorithm.

2.10 Alarm System

The prototype's main objective is to monitor the flood under the bridge and will give an alert signal in nearby villages especially to those in low lying areas. The prototype is trained to detect lines using pi camera and trained as well to give signal whether the line that is detected is in low which green indicator, yellow for alert and lastly for red line indicator for the danger, when the pie camera detects that flood reaches the red line indicator, the alarm system will be activated and that is the sign that all residents residing in that area must be evacuated.

3. HARDWARE DESIGN

The study created a design of the miniature bridge prototype using tinkercad.com. Upon making the design of the bridge, we need to consider the dimensions of each part of the bridge according to the 3D machine that the study used to generate the actual bridge as seen in Figure 5.

The study used filament to create the bridge and a mini aquarium for the case of the whole prototype as shown in Figure 6. The dimension of the bridge has a height of 7.125 inches, a width of 4.125 inches, and a length of 10 inches.

The raspberry pi and the pi-camera are placed outside the aquarium facing the foundation column of the bridge that indicates the level of the flood as shown in Figure 7.

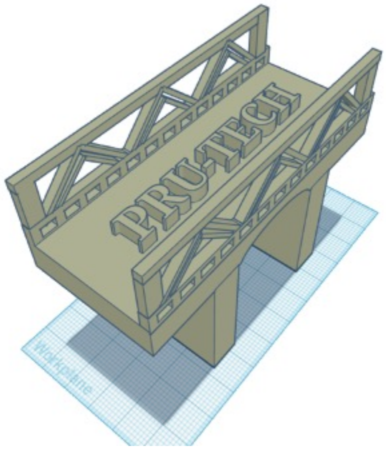

Figure 5: 3D Design of the Simulated Bridge-Column Environment

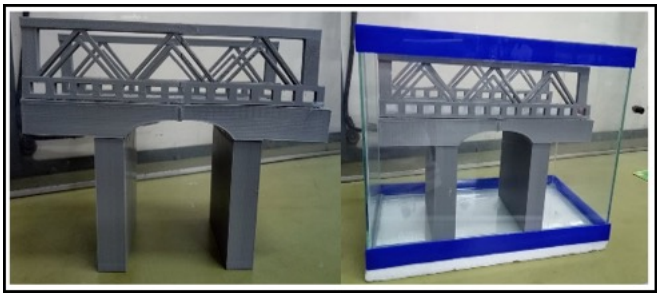

Figure 6: Actual Simulated Bridge-Column Environment

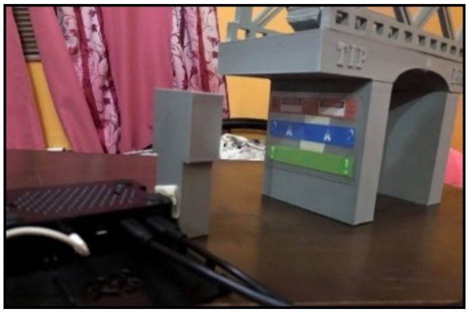

Figure 7: Actual Prototype Setup with the Raspberry-Pi and Pi-Camera

4. RESULTS AND DISCUSSIONS

A. Training Results

The loss function for classification is used in machine learning that represents the inaccuracy of predictions in the system and device' classification problems.

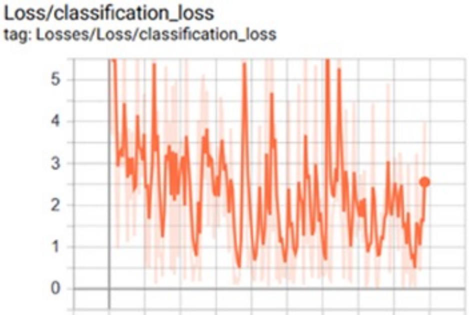

Figure 8: Classification Loss

Figure 8 shows the classification loss where the Y-axis represents the number of classification loss and the X-axis represents the steps used which are 9.8 thousand steps.

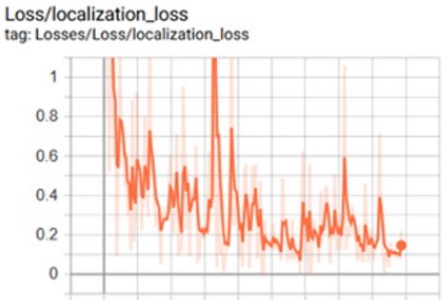

Figure 9: Localization Loss

Figure 9 shows the localization loss where the Y-axis represents the number of localization loss which are the result of loss functions and represent the "price paid for the inaccuracy of predictions" in the classification/localization problems and the X-axis represents the steps used which are 9.8 thousand steps.

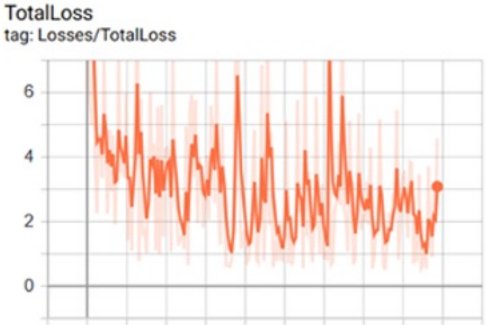

Figure 10: Total Loss

Figure 10 denotes the total loss where the Y-axis reflects the total loss to assess the degree to which the expected values deviate from the actual training results and the X-axis describes the 9.8 thousand steps used.

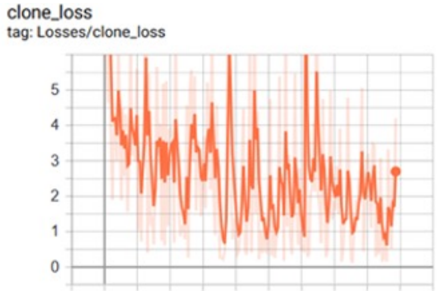

Figure 11: Clone Loss

Figure 11 shows the clone loss where the Y-axis represents the clone loss; a TensorFlow creates a clone of the model for the training of each GPU where it reports the loss in each clone and the X-axis represents the steps used which are 9.8 thousand steps.

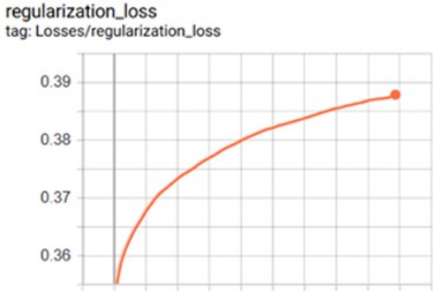

Figure 12: Regularization Loss

Figure 12 shows the regularization loss where the Y-axis represents the regularization loss; this method helps the optimization to be generalized and the X-axis represents the steps used which are 9.8 thousand steps.

B. Testing Results

Table 1: Actual Testing Results

Test No.	Test Case	Expected Output			Actual Output (%Detection)		
		Green	Blue	Red	Green	Blue	Red
1.	B, R	-	-	✓	-	-	98%
2.	B	-	✓	-	-	64%	-
3.	R	-	-	✓	-	-	99%
4.	B, R	-	✓	✓	-	67%	99%
5.	B, R	-	✓	✓	-	99%	73%
6.	R	-	-	✓	-	-	84%
7.	G,B,R	✓	✓	✓	91%	73%	99%
8.	G,B,R	✓	✓	✓	64%	84%	99%
9.	R	-	-	✓	-	-	99%
10.	G,B,R	✓	✓	✓	98%	94%	99%
11.	G,B,R	✓	✓	✓	76%	73%	99%
12.	R	-	-	✓	-	-	99%
13.	G,B,R	✓	✓	✓	96%	66%	99%
14.	B, R	-	✓	✓	-	85%	99%
15.	B, R	-	✓	✓	-	88%	99%
16.	B, R	-	✓	✓	-	73%	95%
17.	R	-	-	✓	-	-	99%
18.	R	-	-	✓	-	-	99%
19.	B, R	-	✓	✓	-	66%	94%
20.	B, R	-	✓	✓	-	67%	99%
21.	G,B,R	✓	✓	✓	94%	66%	93%
22.	G,B,R	✓	✓	✓	66%	73%	73%
23.	B, R	-	✓	✓	-	64%	99%
24.	R	-	-	✓	-	-	99%
25.	G,B,R	✓	✓	✓	99%	64%	98%
26.	R	-	-	✓	-	-	99%
27.	G,B,R	✓	✓	✓	96%	66%	98%
28.	R	-	-	✓	-	-	88%
29.	G,B,R	✓	✓	✓	91%	73%	99%
30.	B, R	-	✓	✓	-	67%	99%
				Mean Average	87.1%	73.6%	95.69

As shown in Table 1, the line color with the highest accuracy is the color Red followed by Green and Blue. Figure 13 shows the test cases that were used in the testing table.

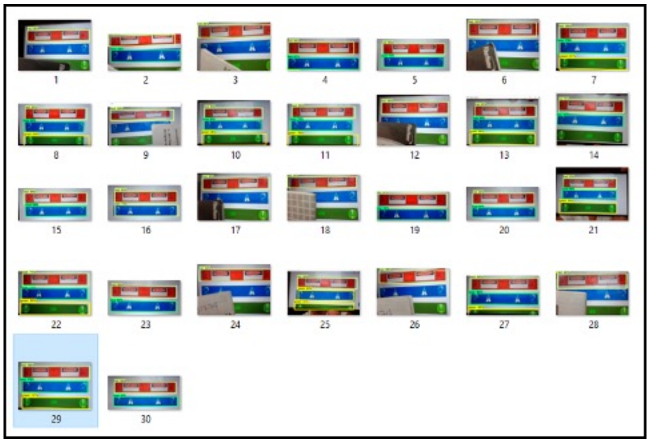

Figure 13: Test Cases

5. CONCLUSION

The study created a system with an automated flood warning system, where it improves the environment from a manual alarm system to an automated alarm system with the use of Raspberry Pi 4 and Camera. The study conducted research where the system has an alarm system that can recognize the water level if it is on below, average, or high level of water streaming the river/sea. The study proved an 87.1% testing accuracy of Green. For Blue, gathered a testing accuracy of 73.6%, while for its Red, it produced a 95.69% testing accuracy. Overall, the accuracy of the whole system produced 85.46%.

ACKNOWLEDGEMENT

The author would like to thank particularly his Microprocessor's Class of 2nd SEM SY19-20. Additionally, to MR-SUAVE Lab, Microprocessors Lab, and 3D Printing Lab of the Technological Institute of the Philippines.

REFERENCES

1. L. Wang, X. Chen, Q. Shao and Y. Li, **Flood indicators and their clustering features in Wujiang River, South China,** *Ecological Engineering*, vol. 76, pp. 66-74, 2015. https://doi.org/10.1016/j.ecoleng.2014.03.018
2. C. Glass, **Dangers from Floods**, Interpreting Aerial Photographs to Identify Natural Hazards, pp. 111-122, 2013.
3. Z. Kundzewicz, B. Su, Y. Wang, J. Xia, J. Huang and T. Jiang, **Flood risk and its reduction in China**, *Advances in Water Resources*, vol. 130, pp. 37-45, 2019. https://doi.org/10.1016/j.advwatres.2019.05.020
4. K. Hansson, M. Danielson and L. Ekenberg, **A framework for evaluation of flood management strategies**, *Journal of Environmental Management*, vol. 86, no. 3, pp. 465-480, 2008. https://doi.org/10.1016/j.jenvman.2006.12.037
5. R. Samu and A. Kentel, **An analysis of the flood management and mitigation measures in Zimbabwe for a sustainable future**, *International Journal of Disaster Risk Reduction*, vol. 31, pp. 691-697, 2018.
6. A. Silva Souza, A. de Lima Curvello, F. de Souza and H. da Silva, **A flood warning system to critical region**, *Procedia Computer Science*, vol. 109, pp. 1104-1109, 2017. https://doi.org/10.1016/j.procs.2017.05.453
7. A. N. Yumang et al., **Real-time flood water level monitoring system with SMS notification**, 2017IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), Manila, 2017, pp. 1-3.
8. S. N. Sakib, T. Ane, N. Matin and M. S. Kaiser, **An intelligent flood monitoring system for Bangladesh using wireless sensor network,** 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV), Dhaka, 2016, pp. 979-984.
9. K. Hiroi and N. Kawaguchi, **FloodEye: Real-time flash flood prediction system for urban complex water flow**, 2016 IEEE SENSORS, Orlando, FL, 2016, pp. 1-3. https://doi.org/10.1109/ICSENS.2016.7808626
10. T. Tingsanchali, **Urban flood disaster management**, *Procedia Engineering*, vol. 32, pp. 25-37, 2012.
11. Feng Xu and Wentao Wang, **Design of flood control and video surveillance system of water resources**, 2010 6th International Conference on Advanced

Information Management and Service (IMS), Seoul, 2010, pp. 432-435.

12. L. Lopez-Fuentes, C. Rossi and H. Skinnemoen, **River segmentation for flood monitoring,** 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, 2017, pp. 3746-3749.
13. K. P. Menon and L. Kala, **Video surveillance system for realtime flood detection and mobile app for flood alert**, 2017 International Conference on Computing Methodologies and Communication (ICCMC), Erode, 2017, pp. 515-519. https://doi.org/10.1109/ICCMC.2017.8282518
14. C. Damle and A. Yalcin, **Flood prediction using Time Series Data Mining**, *Journal of Hydrology*, vol. 333, no. 2-4, pp. 305-316, 2007.
15. C. Ng, **Training of a deep learning algorithm for quadcopter gesture recognition**, *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 1, pp. 211-216, 2020. https://doi.org/10.30534/ijatcse/2020/32912020
16. K. R, **Forecasting of the Air Pollution Based on Meteorological Data and Air Pollutants using Deep Learning: A Novel Review**, *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 1, pp. 801-807, 2020. https://doi.org/10.30534/ijatcse/2020/115912020
17. A. Alon, **SmaCk: Smart Knock Security Drawer Based on Knock-Pattern using Piezo-electric Effect**, *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 2, pp. 339-343, 2020. https://doi.org/10.30534/ijeter/2020/16822020
18. A. Alqudah, **Brain Tumor Classification Using Deep Learning Technique - A Comparison between Cropped, Uncropped, and Segmented Lesion Images with Different Sizes**, *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 6, pp. 3684-3691, 2019. https://doi.org/10.30534/ijatcse/2019/155862019