IoT-based Automatic Flood Warning and Monitoring System using Image Processing

Genrei Paul Aquino
College of Engineering Education
Electronics Engineering Program
Davao City, Philippines
g.aquino.488266@umindanao.edu.ph

Ethan Rigel Cartagena
College of Engineering Education
Electronics Engineering Program
Davao City, Philippines,
e.cartagena.488239@umindanao.edu.ph

Johnwell Caesar Condez
College of Engineering Education
Electronics Engineering Program
Davao City, Philippines
j.condez.488265@umindanao.edu.ph

Abstract - Floods threaten people worldwide, which calls for developing effective warning and monitoring systems. This study proposes using Image Processing and Internet of Things (IoT) technology to flood monitoring systems. This study aims to develop a flood monitoring system that can oversee the water level accurately. It also aims to alert the people around the area through the Telegram application quickly and efficiently. The flood monitoring system used a KNN algorithm for the image processing and telegram API for the output channel. A solar panel setup powers the device. For the data gathering, the researchers performed a simulation by building a mini canal with a flood gauge. The study had a 92% overall accuracy for the expected and predicted color. The study had a 91% overall accuracy for the live video and telegram output. The solar power gains more energy than the energy used by the device, providing more than enough energy for more than 24 hours. Additionally, the IoT-based Automatic Flood Warning And Monitoring System Using Image Processing functioned satisfactorily, fulfilling its primary functions. Overall, the developed device proved efficient and suitable for general applications. Future work includes deploying the system in multiple flood-prone areas under centralized server control, exploring alternative models or algorithms to complement the KNN algorithm, and integrating predictive capabilities for forecasting flood occurrences within specific regions.

Keywords - Telegram API, IP-Cam, Flood Gauge, Raspberry Pi 4, Solar Power, Color Recognition, Machine Learning

I. INTRODUCTION

The Philippines is situated close to the equator and surrounded by various bodies of water, which includes the Pacific Ocean. Water in this particular ocean is warm, and typhoons frequently occur there. Due to the earth's westward-moving wind, these typhoons frequently hit the Philippines. Flooding is one of the natural calamities that frequently result in significant loss of life. Flooding often occurs in conjunction with abrupt, heavy rain, and the water level rises quickly simultaneously. Monitoring is a method to determine the condition of the river to maintain public safety and minimize losses, making it preferable for people to prepare for this catastrophe beforehand [1]. The effects of floods depend on the location, amount, severity, and rainfall duration after a catastrophic typhoon [2].

Most detection systems require human focus and attention and must be fully automated. Automating these systems can help you save time, money, and even lives [3]. This study aims to be beneficial to communities that are prone to flooding. In the study of [4], the researchers used a more realistic test of the image segmentation techniques conducted outside. The images used were divided into the foreground (water) and background using thresholding, region growing, and hybrid techniques. Another study by [5] focuses on Smart IoT Flood Monitoring Systems using the microcontroller ebbed NXP LPC1768; if a public member has internet access, they can keep track of events and foresee the possibility of a flood from the web server. The suggested system is simple to maintain and has a minimal design cost. Their project will update the water level at the web server, and the system will inform the public to evacuate so that the system may take quick action. In the study of [6], it was Created as a computer vision-based automated water level monitoring system based on visual appearance. The performance test led to taking five photos in a 24 24-hour period. Due to the lack of awareness, the river's waterways overflowed, flooding several nearby communities and causing significant damage to both human life and property [7]. The labeling of utility poles, the use of closed-circuit television (CCTV) to monitor rivers and dams, and other typical flood level monitoring techniques are some of the methods used. As a result, manual monitoring's effectiveness is questionable because manual monitoring alone cannot transfer the data it collects immediately. Also, the data may not be accurate due to the inherent limitations of manual monitoring. The study can employ image processing to reduce the number of fatalities during these natural occurrences. This will instantly catch the deluge and detect its level, making information dissemination easier [8]. The study [9] uses the Internet of Things with a telegram application and computer vision to detect people's faces. In their research, the researchers used a computer the size of a credit card with a camera board for the security system, like the Raspberry Pi 3. The Raspberry PI also has a passive infrared sensor (PIR) to identify movement. So, it helps to monitor things and get alerts when motion is detected. It takes a picture and looks for faces, then sends the picture to a smartphone or any other device with a telegram app. The researchers used the Telegram application in the Internet of Things to see what was happening and get alerts when the sensor detected movement. A related study used RPi Hat in their research. Incorporating Raspberry Pi 4 & LTE Base HAT into the study is necessary to make their study accessible remotely. The features include a 3G/4G USB Modem or GSM/GPRS shields that can be used as GSM/GPRS communication modules to ensure an Internet connection. Another study [10] checks it often to predict dangerous water levels at this dam. Their research intends to design a prototype system that can substitute humans in monitoring dam water levels at any time by integrating hardware and software using IoT technology and social media (Twitter and Telegram). The watergate has a height sensor, microprocessor, and wifi module. This system sends real-time water level data to the server. System testing demonstrates that under typical conditions, the system sends data to the server every minute and updates the water level status on Twitter every 5 minutes. The technology sends data to the server every 5 seconds and alerts registered telegram contacts if the water level exceeds a predefined limit. In the study of [11], Image processing is used to suggest and build a smart, automated system model for detecting vehicles to keep traffic moving. This system will use live streaming video to find vehicles that break traffic rules automatically. The study [12] uses the language Python and installs the library OpenCV to do color detection of the RGB images. Real-time data collection is done frame by frame using a camera and processed by the computer to identify the color using the K-Nearest Neighbors (KNN) algorithm [13]. For the best result of using the KNN algorithm, for this research, they used to recognize 12 different colors, and the best K value to have the best result is 5 [14].

This study focuses on developing a more accurate and reliable flood monitoring system using Image Processing and automatically sending the current flood situation to the community via a telegram channel. In the study of [5], their flood monitoring typically consists of a sensor placed in any body of water. However, it could damage their prototype during a flood or any calamity. In this study, the image processing will capture the current flood situation without damaging the devices due to the camera position and flood gauge. The study [8] uses labeling poles and CCTV cameras to monitor the flood level manually, most likely through image processing monitoring. However, the data may not be accurate and reliable due to manual monitoring. Hence, developing real-time automatic flood monitoring using image processing will provide accurate data to the community.

The study aims to develop an IoT-based Automatic Flood Warning and Monitoring System using Image Processing for flood-prone areas to identify the flood level at a specific time and give the community quick advice. It mainly aims (1) to develop a flood monitoring device that will be placed in the flood-prone area, (2) to accurately determine the flood level using an image processing algorithm, and (3) to send warning alerts to the telegram app.

This study will create an automated monitoring system that uses image processing to observe the water level in Bailey Rd, Talomo, Davao City. The device will be used as a warning system to alert the people living in the area should the water rise to a threatening level. This will give the people enough time to prepare before the flood occurs.

This study focuses on developing an automated flood warning and monitoring system that uses image processing to determine the water level of an area and whether the water level is increasing or subsiding. The study serves as a precaution for the people within the given area. The researchers' system is best used in communities near rivers overflowing when heavy rain occurs. The limitation of our study is monitoring the flow rate of rise and fall of the water. It is also limited to real-time monitoring and has no prediction features.

II. MATERIALS AND METHODS

This section includes the conceptual framework, materials, resources, methods, statistical tools, and functionality tests. It covers the flowchart, diagrams, and design, presenting the present study's relevant processes.

A. Conceptual Framework

Figure 1 illustrates how the system operates by installing a camera that takes live images of the flood gauge to monitor the water level. The data gathered will undergo image processing to determine the current water level. Live images and water level updates will be sent to the Telegram application in time intervals via WiFi connection. Furthermore, this comprised the IoT Flood monitoring system.

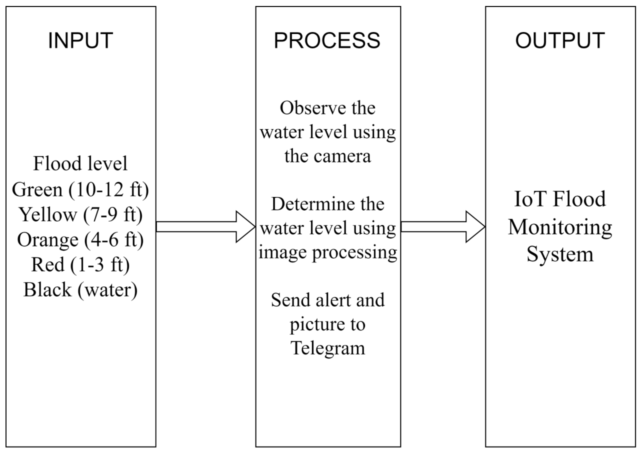

Fig.1 Conceptual Framework Design

B. Materials and Resources

The materials used in the research and how they are operated are as follows: Each material's specifications are also described based on its usage.

Raspberry Pi 4 is the system's main component and is a portable mini-computer for research wherein the algorithm or system is running. An IP camera collects inputs, while the Telegram application is an output of this research. Moreover, the flood gauge is the significant input processed by the system to send predicted output to Telegram. Also, the system uses home prepaid Wi-Fi to connect the Raspberry Pi to the internet through a LAN cable while the IP cam is connected through Wi-Fi. Since the system is running 24/7, at night, the flood gauge is not emitting the correct color because of darkness; therefore, an automatic solar light that turns on at night is the place to light up the flood gauge. Lastly, the system's power source is a Solar Panel, which converts sunlight into electricity. It connects to a solar charge controller, which controls the flow of electricity between the panel and the controller. The controller then charges a 12V battery, and an inverter converts the battery's direct current (DC) electricity into the alternating current (AC) needed to power 220V AC devices. This makes it possible to use solar energy for various purposes, specifically powering the IP camera, Raspberry Pi, and Home Prepaid Wi-Fi.

Fig. 2 Illustrates the CCTV camera dimensions for the flood monitoring CCTV camera, which measures approximately 7.9 inches in length, 3.9 inches in width, and 3.1 inches in height. It's compact enough for discreet placement in flood-prone areas. At the same time, it has features such as a 2MP resolution, 30m (98ft) night vision, multiple lens options, and advanced image processing for real-time flood detection.

Fig. 2 CCTV Camera Dimensions

Fig. 3 Illustrates the prototype dimensions of the Junction Box for the flood monitoring system. The flood monitoring system's junction box is designed precisely, optimizing space and protection for electronic components. Its dimensions—2 1/4" by 1' 1/4" on top, 1'8" in length, 10" in width, and 2'4 1/4" in height—balance compactness with ample room for installation and maintenance. This design ensures efficient operation and durability in challenging conditions, making it essential for reliable flood monitoring.

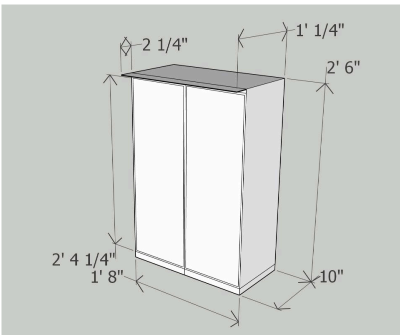

Fig. 3 Junction Box for the Flood Monitoring System

Fig. 4 shows the inside of the Junction Box built to keep the electronic components of the Flood Monitoring system. Inside the junction box, a lead acid battery (12V, 100Ah) is the core power source, supported by a solar charge controller regulating solar panel input. A 500W power inverter facilitates DC to AC conversion for compatibility with standard devices. Circuit protection is ensured through a sequence of DC breakers—16A DC breaker from the solar panel to the charge controller, 32A DC breaker from the controller to the battery (or in series), and 63A DC breaker from the battery to the inverter—and an AC breaker (16A) safeguarding the inverter's output. An Integrated Wi-Fi modem enables remote connectivity, while an SSD stores system data and logs. The Raspberry Pi is the central monitoring unit, overseeing flood levels and relaying updates via Telegram, collectively forming a robust flood monitoring system within a compact enclosure.

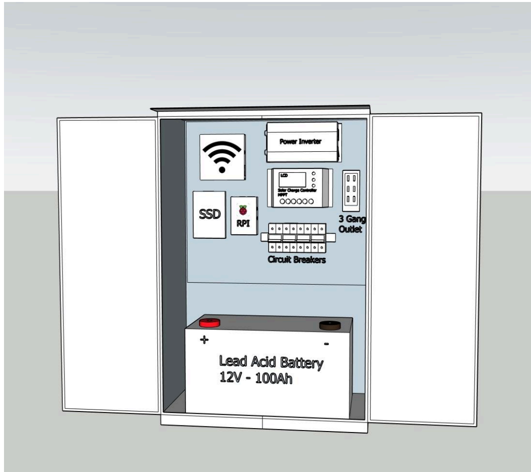

Fig. 4 Junction Box for Flood Monitoring System

Fig. 5 shows the design that will be used, which situates the system in the area's bridge overlooking the flood gauge. The prototype design is a crucial element of the system's development; it visually represents how it will function in the real world. The prototype's dimensions are based on the specific needs of the river underneath the bridge in Bailey Rd, Talomo, Davao City, including the size, shape of the area, types of hazards present, and potential risks to people and property.

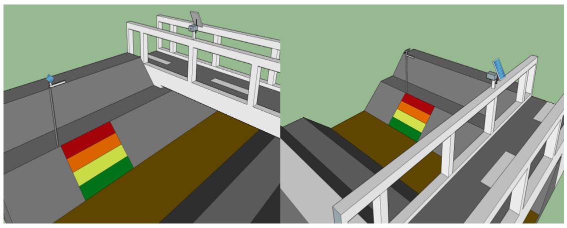

Fig. 5 Design 1 Attached to the Bridge

C. K-Nearest Neighbors Machine Learning Classifier

Fig. 6 shows extracting features to obtain the Red, Green, and Blue Color Histogram values from training images [15]. A Color Histogram essentially shows how colors are distributed within an image. It counts the number of pixels with colors falling within predefined ranges across the image's color space, covering all potential colors. The research uses green, yellow, orange, and red colors. Also, the color of the flood is either black or brown.

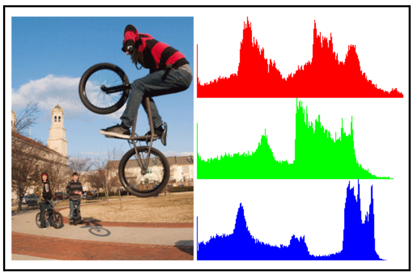

Fig. 6 Feature Extraction

Fig. 7 shows the data points of the trained datasets of the color, and the system uses a standard of 5 as K [15]. The system predicts the processed inputted frame to check the nearest neighbor to the x. Then, from the classifications, the system predicts the output color based on the nearest neighbor.

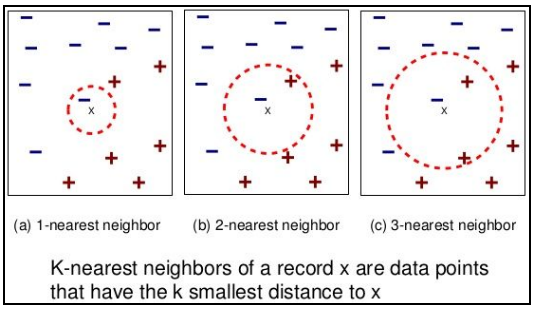

Fig. 7 KNN Classification

Fig. 8 shows the predicted output of the system, the systems using the live camera to collect inputs in real-time [16]. Predicted outputs vary from the data collected frame by frame by the camera. Also, the accuracy of the predicted color depends on the number of datasets the system trains.

Fig. 8 Color Prediction

D. *Telegram API*

Fig. 9 shows the Telegram BotFather in creating the API. that will be used in the system. BotFather streamlines the bot creation and maintenance process, providing users with the tools to customize their bots and integrate them seamlessly into the Telegram ecosystem.

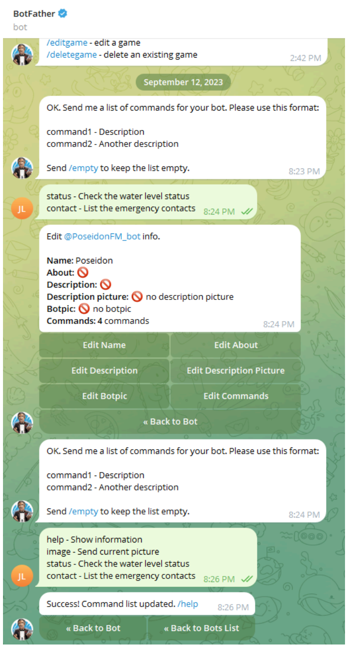

Fig.9 Telegram Bot

Fig. 10 shows the integration of Telegram API to the system by installing the python-telegram-bot library, importing the necessary libraries, setting up the picture processing algorithm, sending alerts to users, handling user input, and adding error handling. This system can help users get real-time warnings about flooded areas so they can take the proper measures and better deal with the flood. This sets up the API key the system will use to connect to the Python code for Python and Telegram to send or request info. Overall, it is a powerful tool for improving flood tracking and response.

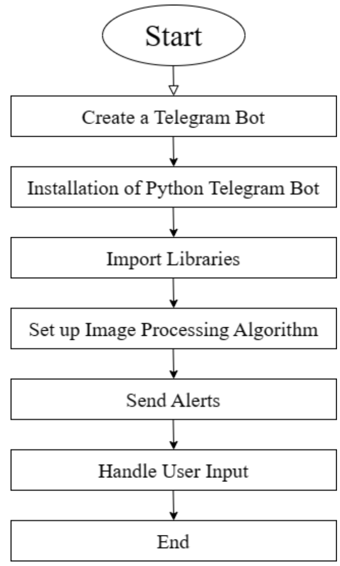

Fig. 10 Structure of the Telegram API

Fig. 11 illustrates the system block diagram for the Flood Monitoring System, which consists of several devices connected to a processing unit. The Raspberry Pi implements the Python code, which is the system's brain. Powered by a power supply that is solar generated to run the connected external devices. The SSD serves as the storage for the system connected to the RPi through USB and the Home Prepaid Wi-Fi via LAN cable to connect the RPi to the internet. The camera captures and gathers data and transmits it to the RPi via WiFi. The processed data is transferred from the RPi to the Telegram channel via the Internet via the cloud.

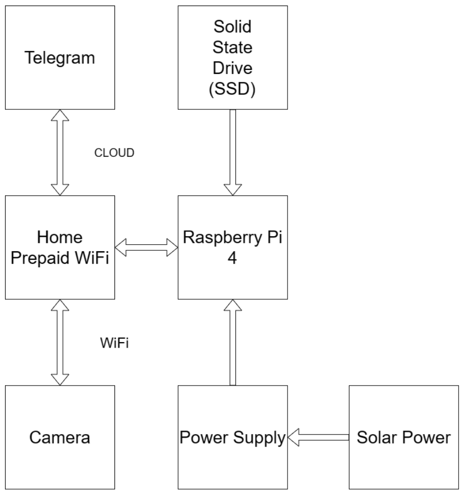

Fig. 11 System Block Diagram

E. System Flow

Fig. 12 shows the system flow of the device that starts from the camera by extracting the images to be image processed. If the water level detected is above the threshold, then it will take a photo of the camera's current time. Furthermore, the system will send that alert level and photo online to the telegram community group.

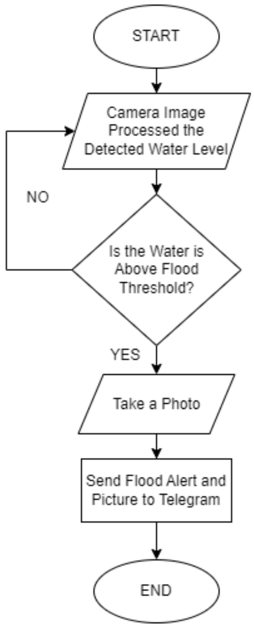

Fig. 12 System Flow

F. Dataset Gathering

Gathering of datasets for the specific colors of the research. The researchers will collect the colors green, yellow, orange, and red from the flood gauge while the colors are black or brown from the water or flood. These collected color pictures are placed in the same filename as the color before training it.

G. Solar Power

This research aims to optimize the solar power consumption within the system to ensure an uninterrupted electricity supply for 24 hours. It involves meticulously calculating wattage based on ampere usage to maximize energy storage and utilization efficiency. Factors such as weather conditions and solar panel efficiency will be rigorously analyzed to enhance the charging capacity of the system.

H. Statistical Analysis

The camera will take live pictures with five classifications to test. A confusion matrix was used to check the system's general accuracy. This validates the authenticity of the KNN algorithm. To use a confusion matrix in the study, researchers will first gather data and use machine learning techniques to build an image classification model. The confusion matrix is then made by comparing the labels the model thought would be used with those used. Metrics like accuracy, precision, and recall from the confusion matrix are used to judge the model's performance, and the model is tweaked as needed to improve its performance. Lastly, the model is put into the system to find flooded areas automatically in real-time. This makes flood tracking and response efforts more effective.

I. Functionality Test

To test the study's functionality, the researchers will conduct different field tests by creating a flood gauge simulation. As the researcher's statistician and adviser suggested, the researchers made a miniature canal to imitate the riverside flood gauge. Due to the problem, the researchers cannot manipulate the water in the river to conduct proper testing. Outlines a system for warning people about floods [17]. It has four warning codes corresponding to different flood level ranges and recommended actions. The height from base to flood line is divided into 4. In this flood gauge, 12 feet is divided into four. In contrast, the lowest range of 1-3 feet is marked with a "Red" warning code, indicating that critical action is required and assisted evacuation is recommended. The next 4-6 feet range is marked with an "Orange" warning code, indicating that pre-emptive evacuation is recommended. The 7-9 feet range is marked with a "Yellow" warning code, indicating that people should be alert and ready. Finally, the 10-12 feet range is marked with a "Green" warning code, indicating that normal monitoring is sufficient. The system will also show if the flood level is increasing or subsiding. The alert update on Telegram is posted every 30 minutes for the green warning code, 10 minutes for the yellow warning code, 3 minutes for the orange warning code, and 1 minute for the red warning code.

Table 1. Functionality Test

Flood Level (ft)	**Warning Code**	**Output**
1 - 3 Feet	*Red*	*Critical (Assisted Evacuation)*
4 - 6 Feet	*Orange*	*Warning (Pre - Emptive Evacuation)*
7 - 9 Feet	*Yellow*	*Normal (Alert or Ready)*
10 - 12 Feet	*Green*	*Baseline (Normal Monitoring)*

III. RESULTS AND DISCUSSIONS

A. Automated Flood Monitoring System

Fig. 13 shows the Automated Flood Monitoring System deployed in Bailey Rd, Talomo, Davao City. Inside the junction box, the following components are used: a lead acid battery, solar charge controller, power inverter, 16A DC, 32A DC breaker, 63A DC breaker, Wi-Fi modem, and SSD.

Fig. 13 Automated Flood Monitoring System (Junction Box)

Fig. 14 and 15 show the Automated Flood Monitoring System deployed in Bailey Rd, Talomo, Davao City. Outside the junction box with the following components used: placed above the bridge is the solar panel, and attached to a 5-foot pole is the CCTV camera and the solar-powered light facing the flood gauge.

Fig. 14 Automated Flood Monitoring System (External)

Fig. 15 Automated Flood Monitoring System (External) + Flood Gauge

The following devices, as shown in Table 2, use electricity to be able to use. Since the area of implantation lacks a power source, the researchers use solar energy to power the devices. Based on the table, the current usage is far below the maximum current specification. Furthermore, all devices' current usage ranges from 1.23 to 1.32 ampere.

Table 2. Current Specification Vs. Current Usage

			Usage	
	Specification		**Current**	
Devices	**Voltage**	**Ampere**	**Min**	**Max**
IP Cam	12	1	0.29	0.31
Home Prepaid Wi-Fi	12	1	0.26	0.28
Raspberry Pi 4	5	3	0.68	0.73
Total Current Usage			**1.23A**	**1.32A**

Fig. 16 shows the line chart of the solar charging compared to the maximum current usage of the devices. The y-axis shows the current value, while the x-axis is the 24-hour observation of the current usage and solar charging. Moreover, the yellow line on the chart indicates the difference between the current of solar charging and the current usage. Then, the green line indicates the average value of the results of the yellow line. It concludes that the green line is located in the positive part of the line chart. Therefore, the charging time overwhelms the maximum usage of the devices.

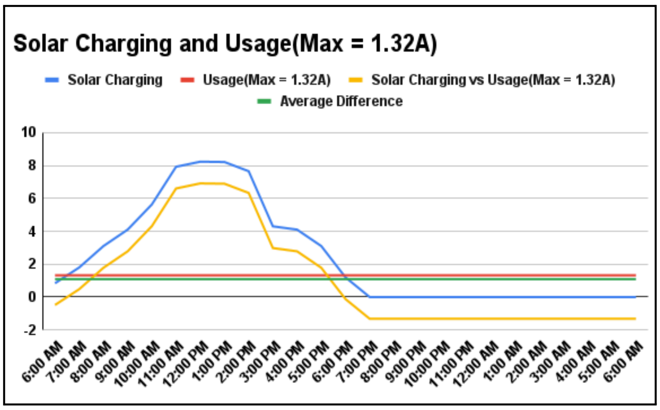

Fig. 16 Solar Charging Vs. Usage

B. *Accuracy Test Result*

The accuracy test results for the Flood monitoring system are shown in Table 3. To simulate flood detection, the researchers, recommended by their statistician and adviser, were advised to build a mini canal with a flood gauge. Ten trials were conducted to gather data for the Flood Monitoring System. Each trial was simulated with the flood level rising from the green level to the red level and then the water subsiding from the red level to the green level. Five trials were conducted during the day, while five were conducted at night.

Table 3 has five (5) colors, each representing its own flood gauge and black representing water. Green had a 20/20 result, and yellow had a 39/40 result, with one (1) showing incorrectly as orange. Orange had a 58/60 result with two (2) errors showing as red. Red had a 66/80 result, with fourteen (14) errors as orange. Black showed one (1) in green, one (1) in yellow, three (3) in orange, ten (10) in red, and one hundred eighty-five (185) in black. Moreover, the Overall Precision results stand at 93.83%, while the overall recall is 90.73%. This culminates in a 92% overall accuracy rating

Table 3. Confusion Matrix: Expected Color Prediction vs Actual Color

CONFUSION MATRIX								
COLOR GAUGE: EXPECTED COLOR PREDICTION VS ACTUAL COLOR PREDICTED (ALGORITHM)								
	ACTUAL							
	400	GREEN	YELLOW	ORANGE	RED	BLACK	Classfication Overall	User's Accuracy Precision
EXPECTED	GREEN	20					20	100.00%
	YELLOW		39	1			40	97.50%
	ORANGE			58	2		60	96.67%
	RED			14	66		80	82.50%
	BLACK	1	1	3	10	185	200	92.50%
	Truth Overall	21	40	76	78	185	400	
	Producer's Accuracy (Recall)	95.24%	97.50%	76.32%	84.62%	100.00%		
Overall Precision								93.83%
Overall Recall								90.73%
Overall Accuracy								92.00%

C. *Telegram Output Accuracy*

Table 4 shows the live video output of the Flood Monitoring System in contrast with the output shown in the Telegram Channel. Green and yellow had a perfect 40/40 and 20/20 results respectively. While orange had 13 expected and showed 7 red in telegram, resulting in a 13/20 result. Also, red has 18 correct colors with 2 orange colors, resulting in an 18/20 score. Moreover, the overall precision is 88.75%, and the overall recall is 89.67%. Therefore, the test had an overall 91% accuracy rating. Additionally, the average transmission time of the data from the system to the Telegram channel is 2.353179153 seconds."

Table 4. Confusion Matrix: Live Video Vs. Telegram Output

CONFUSION MATRIX							
LIVE VIDEO VS. TELEGRAM OUTPUT							
	ACTUAL						
	100	GREEN	YELLOW	ORANGE	RED	Classfication Overall	User's Accuracy Precision
EXPECTED	GREEN	40				40	100.00%
	YELLOW		20			20	100.00%
	ORANGE			13	7	20	65.00%
	RED			2	18	20	90.00%
	Truth Overall	40	20	15	25	100	
	Producer's Accuracy (Recall)	100.00%	100.00%	86.67%	72.00%		
Overall Precision							88.75%
Overall Recall							89.67%
Overall Accuracy							91.00%

IV. CONCLUSIONS AND FUTURE WORKS

An IoT-based Automatic Flood Warning and Monitoring System using Image Processing was successfully designed. Incorporating a flood monitoring system that sends alerts to a telegram application through Wifi. Solar power was used to power up the device. The current power usage ranges from 1.23 to 1.32 ampere across all devices. The solar power gains a more considerable amount of energy than the energy used by the device. The color recognition had a 92% overall accuracy rating for the accuracy testing. Also, the live video vs telegram output test had a 91% overall accuracy rating. Lastly, the average transmission time from the system to the telegram channel was 2.353179153 seconds, proving that the short delay is not noticeable, expecting there will be a delay for the system using an image of its file. Moreover, the statistical analysis provides substantial evidence that the system has high color recognition and KNN algorithm is good for flood monitoring.

For future works, the researchers advised that the same system will be placed in several flood-prone areas and controlled or monitored by one main server. Another model or algorithm to be used that can compare the result to the KNN algorithm. Additionally, implementing predictions of when floods will occur within an area.

REFERENCES

[1] Q. Zhang, N. Jindapetch, R. Duangsoithong, and D. Buranapanichkit, *"Investigation of Image Processing based Real-time Flood Monitoring,"* Nov-2018. Available: https://ieeexplore.ieee.org/document/8688775.

[2] B. Bennett, M. Leonard, Y. Deng, and S. Westa, *"An empirical investigation into the effect of antecedent precipitation on Flood Volume,"* Journal of Hydrology, 13-Oct-2018. Available: https://www.sciencedirect.com/science/article/abs/pii/S0022169418307868.

[3] H. B. Baydargil, S. Serdaroglu, J.-S. Park, K.-H. Park, and H.-S. Shin, *"Flood Detection and Control Using Deep Convolutional Encoder-decoder Architecture,"* IEEE Xplore, 2018. Available: https://ieeexplore.ieee.org/document/8549916.

[4] N. A. Muhadi, A. F. Abdullah, S. K. Bejo, M. R. Mahadi, and A. Mijic, *"Image segmentation methods for Flood Monitoring System,"* MDPI, 26-Jun-2020. Available: https://www.mdpi.com/2073-4441/12/6/1825.

[5] S. B. Zahir, P. Ehkan, T. Sabapathy, M. Jusoh, M. N. Osman, M. N. Yasin, Y. A. Wahab, N. A. M. Hambali, N. Ali, A. S. Bakhit, F. Husin, M. K. M. Kamil, and R. Jamaludin, *"Smart Iot Flood Monitoring System"* - IOP Science - Institute of Physics, 2019. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1339/1/012043.

[6] L. W. Kuswidiyanto, A. P. Nugroho, A. W. Jati, G. W. Wismoyo, M. Murtiningrum, and S. supadmo Arif, *"Automatic water level monitoring system based on computer vision technology for supporting the irrigation modernization,"* ResearchGate, Mar-2021. Available: https://www.researchgate.net/publication/350540520_Automatic_water_level_monitoring_system_based_on_computer_vision_technology_for_supporting_the_irrigation_modernization.

[7] J. L. Gan and W. Zailah, *"Water level classification for flood monitoring system using convolutional neural network,"* SpringerLink, 2018.Available: https://link.springer.com/chaptea0.1007/978-981-15-5281-6_21.

[8] L. K. S. Tolentino, R. E. Baron, C. A. C. Blacer, J. M. D. Aliswag, D. C. E. De Guzman, J. B. A. Fronda, R. C. Valeriano, J. F. C. Quijano, M. V. C. Padilla, G. A. M. Madrigal, I. Valenzuela, and E. Fernandez, *"Real time flood detection, alarm and monitoring system using image processing and multiple linear regression,"* SSRN, 2022. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4319789.

[9] I. Aydin and N. A. Othman, *"A new IoT combined face detection of people by using computer vision for security application,"* IEEE Xplore. Available: https://ieeexplore.ieee.org/document/8090171/.

[10] J. E. de la Cruz, C. A. Goyzueta, and C. Cahuana, *"Intrusion Detection and Prevention System for Production Supervision in Small Businesses Based on Raspberry Pi and Snort,"* IEEE Xplore, Sep-2020. Available: https://origin-ieeexplore.ieee.org/Xplore/home.jsp.

[11] F. M. J. M. Shamrat, I. Mahmud, A. K. M. S. Rahman, A. Majumder, Z. Tasnim, and N. I. Nobel, *"A Smart Automated System Model For Vehicles Detection To Maintain Traffic By Image Processing,"*

ResearchGate, Feb-2020. Available: https://www.researchgate.net/publication/339292442_A_Smart_Automated_System_Model_For_Vehicles_Detection_To_Maintain_Traffic_By_Image_Processing.

[12] P. Raguraman, A. Meghana, Y. Navya, S. Karishma, and S. Iswarya, *"Color Detection of RGB Images Using Python and OpenCV"* ResearchGate, Jan-2021. Available: https://www.researchgate.net/publication/349355136_Color_Detection_of_RGB_Images_Using_Python_and_OpenCv.

[13] A. Borawake, N. Kulkarni, and A. Ghorse, *"Real-time object color identification,"* IJIRT, Jan-2021. Available: https://ijirt.org/master/publishedpaper/IJIRT150658_PAPER.pdf.

[14] R. Bayraktar, B. A. Akgul, and S. Kadir, *"Color Recognition Using Color Histogram Feature Extraction and K-Nearest Neighbor Classifier,"* ResearchGate, Jan-2020. Available: https://www.researchgate.net/publication/341788229_Color_Recognition_Using_Color_Histogram_Feature_Extraction_and_K-Nearest_Neighbor_Classifier

[15] D. Subramanian and T. A. Cohen, *"Machine Learning Systems,"* SpringerLink, Novewmber-2022. Available: https://link.springer.com/chapter/10.1007/978-3-031-09108-7_6

[16] K. Pooja, R. Shreya, B. Yashika, S. Lakshmi, and B. Rekha, *"Color Detection using K-Nearest Neighbors Classification Algorithm"* IRJET, 2021, Available: https://www.irjet.net/archives/V8/i8/IRJET-V8I8147.pdf

[17] *"Disaster risk reduction and management,"* DCDRRMO. Available: https://drrmo.davaocity.gov.ph/.