Computer Vision-based Water Level Measurement Using YOLO v8 Nano

Yingren Wang (Student), Tao Huang (Supervisor)

College of Science and Engineering, James Cook University, Cairns QLD,4878, AUSTRALIA

***Abstract*— Water level measurements are important for effective water resource management, disaster monitoring, and prevention, particularly in the context of flood monitoring. This study highlights the limitations of conventional water level measurement methods and the potential of computer vision-based methods for accurate and cost-effective water level measurement. The objective of this project is to present a computer vision-based water level measurement technique that is cost-effective, accurate, and easy to distribute. The proposed method utilizes a state-of-the-art object detection algorithm and provides an overview of its performance. The result of this method provides an error rate of 0.05% to 0.2% within the full scale under good lighting conditions which could potentially outperform conventional methods. Moreover, the error rate was less than 2% under bad lighting conditions. Overall, this method shows promising and accurate measurements, and has the potential to integrate already installed cameras or Internet of Thing platforms and distribute the method efficiently.**

***Index Terms*—water level measurement, computer vision, YOLOv8, object detection**

I. Introduction

Water level measurements are important for effective water resource management, disaster monitoring, and prevention, particularly in the context of flood monitoring. In recent years, the increasing frequency of flood events has led to significant increases in fatalities and massive economic losses. According to the United Nations [1], floods result in the highest number of fatalities among all natural disasters. Furthermore, climate change [2] has resulted in more frequent and severe flood events in urban areas, thereby increasing the risk of damage and loss. This emphasizes the importance of closely monitoring floods to facilitate prompt action and reduce adverse consequences. Flood management is critical for maintaining and controlling a region's risks and for protecting lives. It comprises of four phases: prevention, preparedness, response, and recovery [3]. Flood prediction is a key component in flood prevention. One of the indicators for predicting a flood event is the rapid movement of water level [4]. Therefore, measurement of river water levels can provide robust information for flood prediction.

Employing conventional water level measurement methods to achieve optimal monitoring objectives can be costly and requires extensive maintenance [5]. Remote sensing technologies such as satellite altimetry can measure a wide geographic area with reasonable accuracy [6, 7]. However, processing these data can be time consuming and expensive, hindering real-time water level measurements. Ground sensors, including pressure, capacitance, and ultrasonic sensors, can provide high accuracy at lower cost [8]. In 2017, Duraibabu et al. [9] developed an underwater pressure and temperature sensor using fibre optic technology. This sensor was subsequently tested in seawater and attained a sensitivity of 15 nm/kPa. Furthermore, capacitance-based sensors are known for their cost efficiency [8], while ultrasonic and Lidar sensors can achieve a measurement accuracy of 0.01m within the 0-10m range [10-12]. However, these measurement devices must be calibrated periodically to meet hydrological standards. Moreover, because some of these sensors are situated in water, flash floods can easily damage them, resulting in a greater need for maintenance [3]. In addition, conventional methods typically struggle to comprehend conditions at sites where flooding occurs. As a result, additional cameras are usually installed at locations to monitor the hydrological situation [5]. In recent years, advancements in computer processing units have led to significant improvements in the speed and affordability of image processing algorithms. Consequently, the utilization of computer vision for water level measurements has become increasingly practical and cost-effective [3].

The conventional method of applying computer vision to water level measurement is to apply edge detection to extract the Region of Interest (ROI). In 2018, Lin et al. [13] proposed a method that used a single-lens reflex (SLR) camera to capture multiple photos within 25 hours in an urban area. They applied least-squares matching (LSM) to detect camera movement and adopted normalized cross-correlation (NCC) to perform movement correction. Subsequently, a Gaussian filter was used to optimize the ambient noise, and a Canny edge detector was employed to extract the ROI. After detecting the water line using the Hough transform, the method automatically calculated the water level based on pixels from the reference points to the water line. Kuo and Tai [14] presented a similar approach in 2022. They identified the ROI using a histogram of the images and marked the reference points on the staff gauge for level calculation. In addition, the NCC technique was applied during image processing to minimize the image shift caused by the camera vibrations. Normally, these methods require reasonably good image quality to apply these algorithms step by step. To use the night-version application, a near-infrared (NIR) imaging video camera may be required. The NIR camera improves image contrast

and filters reflection noise from the water surface [15]. Although the NIR camera showed a significant improvement in low-visibility water level measurements, its deployment complexity, such as the installation of new cameras and the application of multiple algorithms, still hinders its distribution.

Several novel approaches incorporating deep learning algorithms have been introduced. For example, in 2022, Rizk et al. [16] proposed a drone-based system for flood damage assessment that utilized the deep learning object detection algorithms Region-Based Convolutional Neural Network (R-CNN) and Visual Geometry Group 16 (VGG-16). This system was capable of estimating floodwater levels by identifying objects, such as houses and cars. The system involves aerial object detection and water level classification. R-CNN was applied for object detection, whereas VGG-16 was used for water level estimation. In 2020, Lin et al. [17] proposed a computer vision-based method for water level measurement involving the use of the semantic segmentation network DeeplabV3 + to identify the water gauge, followed by the application of K-means clustering to divide the crop water gauge into sections. Finally, a VGG-8 convolutional neural network was utilized to recognize each region. This method was unique in that it allowed for the measurement of water levels without the need to consider the installation location of the camera. Similarly, in 2021, Jafari et al. [18] proposed a real-time water-level monitoring technique utilizing live street cameras. This technique employs a novel approach called object-based image analysis (OBIA), which uses a pre-measured object as a reference. The approach adopted a Fully Convolutional Network (FCN) as an image segmentation and object-labelling algorithm. This technique successfully detected floodwater levels in downtown Houston during Hurricane Harvey. These methods allow the use of cameras that have already been installed. Compared to the conventional computer vision method, it significantly decreases the installation cost.

The objective of this project is to present a computer vision-based water-level measurement technique that uses deep learning algorithms to reduce the complexity of the previous method to achieve cost-effectiveness, acceptable accuracy, and easy distribution.

The remainder of this paper is organized as follows. Section II describes the methodology of the water-level measurement methods, provides an overview of our method, and introduces a state-of-the-art object detection algorithm. Section III describes the results of our proposed method and discusses further improvements.

II. Methodology

A. Method Overview

An overview of the proposed method is presented in Fig. 1. It starts with image transformation, which aims to orthogonally transform the image; some simple steps of image processing, such as cropping and resizing, are needed, but are not discussed in this paper. Second, a custom trained YOLOv8n model is used to detect the water gauge. Finally, the water level was calculated based on the known variables.

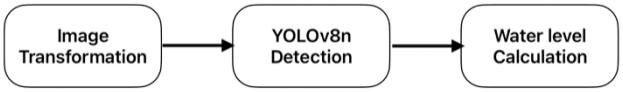

Fig. 1: Overview of the proposed method.

B. State of the art Object Detection algorithm

Traditional computer vision methods have demonstrated effectiveness; however, they face limitations in detecting water levels under adverse circumstances such as at night or during heavy rain. In contrast, deep learning is a type of supervised learning that requires labelling only once and can perform robustly without requiring human intervention. Consequently, integrating deep learning algorithms into water level measurements has the potential to overcome the limitations of the conventional methods. In recent years, object detection algorithms have been developed dramatically. In this section, we briefly review these algorithms.

Two-stage detectors were designed to enhance accuracy. In 2014, Girshick et al [19]. proposed a region convolutional neural network (R-CNN) for object detection, which is a two-stage method that integrates a regional proposal and the classification result of a (CNN) to detect predefined objects. To speed up the process, Girshick refined this method as Fast R-CNN in 2015 [20]. At the same year, Ren et al. [21] proposed a Faster R-CNN to improve speed by replacing the RPN in the previous region proposal stage. In 2017, He et al. [22] introduced a new segmentation algorithm based on Faster CNN, called Mask R-CNN, which significantly enhanced the segmentation speed. In 2017, Lin and Dollar [23] proposed a feature pyramid network that improved detection accuracy. Based on these pre-stage versions of two-stage methods, evolutionary methods continue to emerge [24].

One-stage detectors focus on the processing speed. One good example of a one-stage detector is YOLO. In 2016, Redmon presented You Only Look Once (YOLO), a single-stage detector designed for regression problems [25]. Redmon and Farhadi subsequently enhanced YOLO to YOLOv2 by employing unsupervised learning in 2017 [26].

In addition, they refined the original method by integrating the Residual Network (ResNet) model and the latest backbone network to create YOLOv3 [27], which exhibited considerable improvements in both speed and accuracy. Although Redmon stopped researching computer vision after YOLOv3, in recent years, this method has advanced rapidly by other scholars[28]. Fig. 2 shows the timeline of YOLO, the latest version upon writing this paper is YOLOv8, meanwhile, it has outperformed all other previous versions as Fig.4 shows.

The architecture of YOLOv8 as shown in Fig. 3-4 shows, The Backbone of the model comprises a series of convolutional layers that extract pertinent features from the input image. The SPPF layer and subsequent convolution

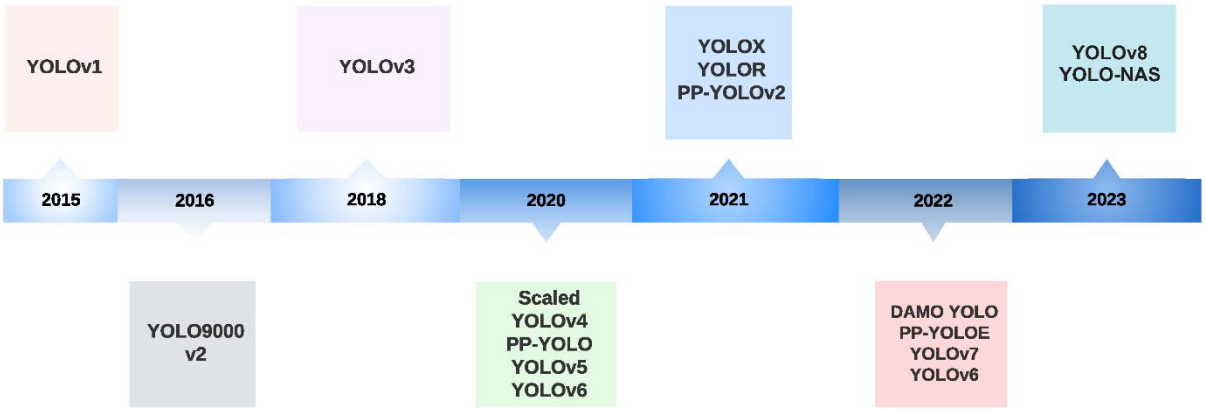

Fig. 2: A timeline of YOLO [28].

layers process features at various scales, whereas the Upsample layers augment the resolution of the feature maps. The C2f module integrates high-level features with contextual information to enhance detection accuracy. Ultimately, the Detection module utilizes a set of convolutional and linear layers to map high-dimensional features to the output bounding boxes and object classes. The architecture is aimed at speed and efficiency; however, it still achieves a high accuracy.

Fig 3. Backbone of YOLOv8 P5[29]

In this project, our application using an object detection algorithm is simple, which is focused solely on detecting the water line. Therefore, we aim to achieve straightforward results with minimal complexity. Thus, we selected a single-stage detector that could efficiently fulfil our requirements, as opposed to a two-stage detector that may have been more complicated and offered unnecessary features. In addition, Fig. 5 shows that YOLO v8n has smaller complexity and minimum latency. This has the greatest potential to align with our project objective of easy distribution. Therefore, we will implement a version of YOLO v8, YOLOv8n, where n represents Nano, which is the smallest version of YOLOv8.

C. Photogrammetric Method for Transformation

The primary objective of this step is to transform three-dimensional real-world objects into two-dimensional images. We require a method that can be applied to any real-time monitoring camera because the camera's location may differ. Several methods have been used in previous computer vision-based water level measurement methods. Zheng proposed a straightforward approach based on the pinhole camera model. Ideally, this method only requires measurement of the camera angle and its corresponding coordinate system for calibration. Yu and Hann [30] utilized four known reference points near to support image transformation. This method requires an understanding of the distance between the camera and object. However, in some cases, it may be difficult to measure the distance between the object and the camera and typically requires additional distance measurement sensors, such as lidar or ultrasonic sensors. Di Zhang [31] utilized a method which requires pre-calculation of the T matrix as Fig. 6 shows. Under certain conditions, when the camera is fixed, the pre-calculation of the T matrix requires only the real height and width of the water gauge. Under these conditions, the T matrix can be used for any further transformation at the same location. In this project, we adopted the T-matrix method for our proposed method.

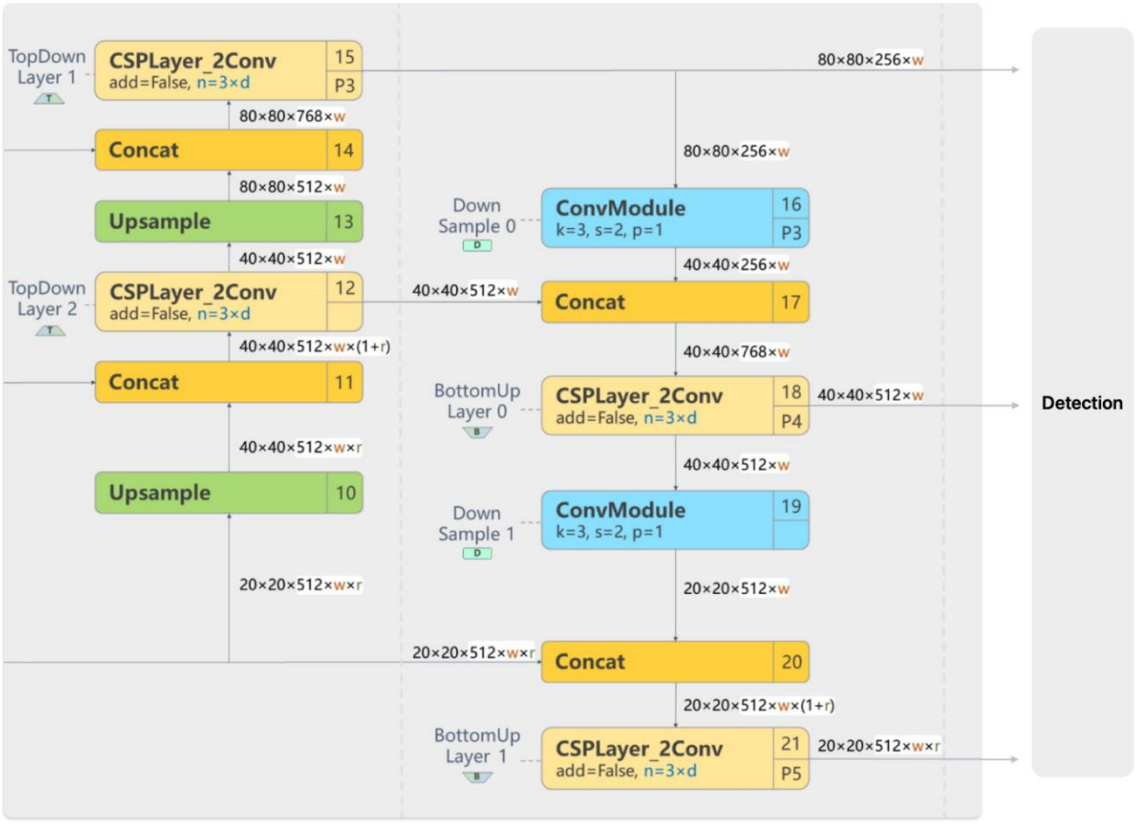

Fig. 4: Neck and head of YOLOv8[29]

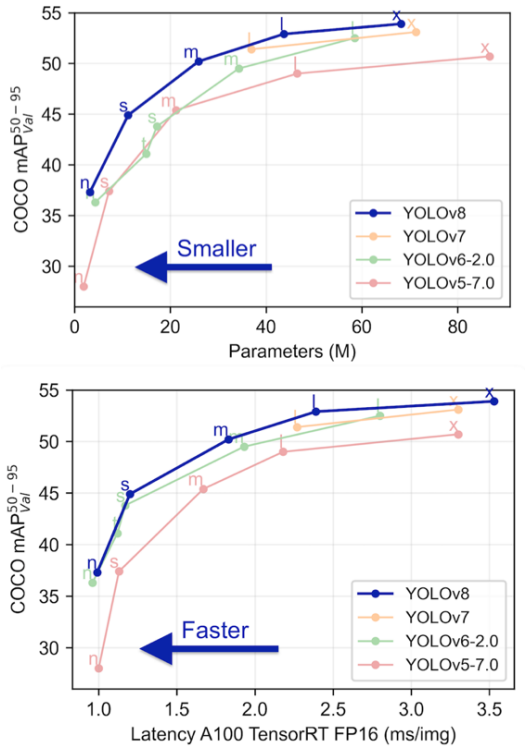

Fig. 5: Performance of YOLOv8, it outperforms other previous in all models, the model from left to right is nano to extra (n → x). The performance tests were conducted on the COCO dataset. The graph on the top shows the complexity of the model in terms of accuracy. The graph on the bottom shows the speed of the model in terms of accuracy.

Normally, in most application cases, the camera and water gauge are fixed; therefore, the T-matrix remains unchanged throughout the entire life cycle. Pre-calculation was performed only once for each location, and the pre-calculated T-matrix was used to transform all the following images. In real applications, even with camera movement over time, using this method, we can calibrate the T matrix without physically attending to the site. Therefore, it significantly decreases the maintenance costs. The calculation process for T matrix is shown in Equation (1).

$$\begin{bmatrix} x_i \\ y_i \\ w \end{bmatrix} = \begin{bmatrix} t_1 & t_2 & t_3 \\ t_4 & t_5 & t_6 \\ t_7 & t_8 & t_9 \end{bmatrix} \begin{bmatrix} W \\ H \\ 1 \end{bmatrix} \quad (1)$$

D. Model Building and Data

1) Data Collection

The project was conducted in a workshop owing to financial and timing constraints. Fig. 7 shows the data-collection process.

It is worth noting that our proposed method includes the T-matrix transformation; however, we assumed that if training the model from transformation images, the result would not be significantly affected. Therefore, we skipped this step during the experiment. In addition, this position setting allowed us to control lighting more precisely. We employed an adjustable smart light bulb to alter the

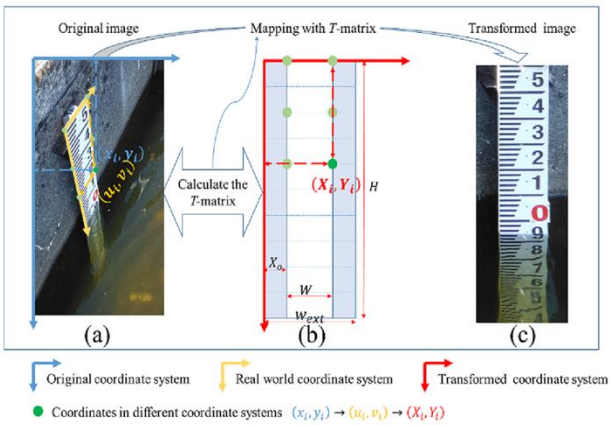

Fig. 6: A overview of T-matrix transformation [33]

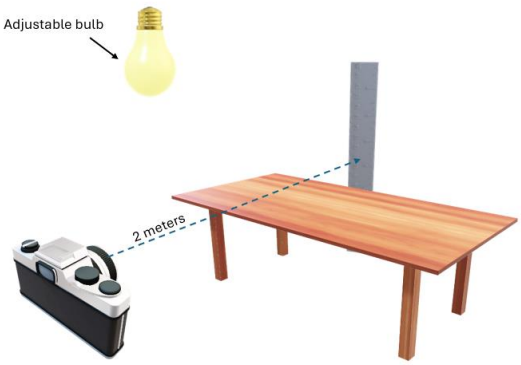

Fig. 7: Data collection design. The camera was positioned in front of a ruler (a simulation of the water level gauge). The light bulb and the table were in the same vertical plane.

brightness of the environment. We collected two datasets: model training (50 images for labeling and 10 for validation) and measurement testing (20 images). A total of 80 images were captured under various conditions in these datasets, including the blurring and dimming of light. Additionally, images were taken from Sony A72 with an FE 24–70 mm f/4 ZA OSS Lens, and we used an M-mode manual control ISO and shutter speed and kept them constant when we changed the environmental data.

One of the most difficult tasks in this project was to accurately determine the water line between the gauge and the water using this method. To achieve the objective of this project, we utilized YOLOv8n to detect the waterline. To train the model, we divided it into the labelling and training processes.

2) *Model Labelling*

Labelling images is undoubtedly the most laborious and time-consuming step of this project, demanding meticulous attention to detail. However, it is a task that only needs to be performed once per location throughout the project lifecycle, unless there are significant changes in environmental conditions. Our approach to image labelling involved leveraging the free version of Roboflow, an

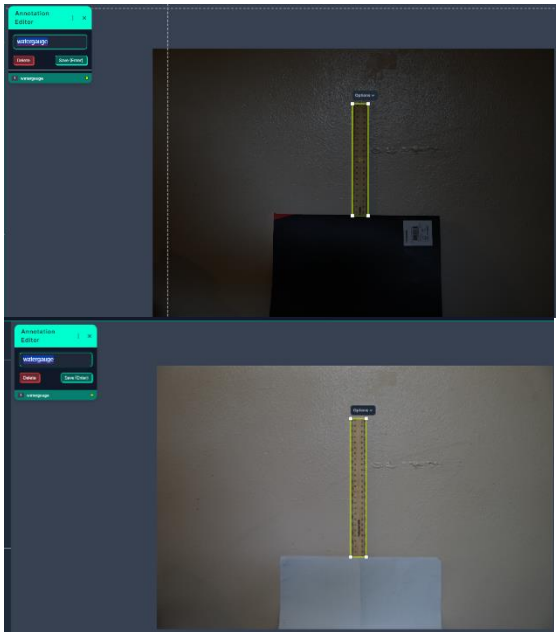

Fig. 8: Examples of Image annotation. All target images must be undertaken T-matrix transformation before annotation.

annotation website. While this choice only provided a functional solution, it is worth noting that more advanced software options exist that could potentially enhance the accuracy and efficiency of the labelling process. Unfortunately, because of budget and time constraints inherent in the project, accessing these advanced tools is not feasible.

One of the critical challenges in image labelling, as depicted in Fig. 8, is the accurate annotation of water gauges. The precision of these annotations directly impacts the model's ability to accurately draw a bounding box around the water level gauge, and subsequently affects the precision of the water level calculation. To mitigate these challenges and improve the accuracy of the labelling process, we employed several strategies. First, increasing the number of annotated images can help minimize the impact of individual mistakes. In addition, incorporating feedback loops into the labelling process can help identify and rectify errors more efficiently. We randomly reviewed the annotated images to ensure that the labelling process was aligned with the project objectives.

3) *Model Training*

The model's training was performed within the Windows operating system environment, utilizing the PyTorch framework [32] and YOLOv8n model architecture, and running on an NVIDIA GeForce GTX1650 GPU. The training process was fine-tuned by adjusting crucial parameters, such as batch size, learning rate, and epochs. Batch size refers to the number of samples selected at the start of each iteration. A smaller batch size reduces the likelihood of convergence, but a larger batch size can result in falling into a local optimum, which can slow the training process. When the learning rate is set too low, the training process becomes extremely slow. Conversely, if the

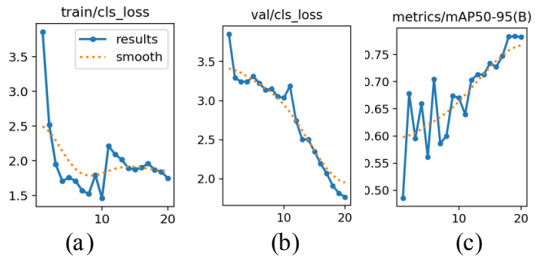

Fig.9: Key indicators of the training results, (a) training loss vs epochs. (b) validation loss vs epochs, (c) mAP vs epochs.

learning rate is set too high, it will skip the optimum model. Epoch refers to each complete process during the training process. The algorithm adjusts other parameters according to the accuracy of the previous epoch. To minimize the training process, the epoch should be set when the accuracy of the model stops increasing. These parameters were carefully calibrated and fed into the pre-training model along with the dataset to optimize the training process.

After careful tuning of these parameters during the training process, our final configuration comprised 20 epochs, a batch size of 8, a learning rate of 0.004, and an image size of 640 pixels. Each of these settings was carefully chosen to make a trade-off between model accuracy, training duration, and risk of overfitting. As shown in Fig. 9, throughout the training process, as the number of epochs increased, the mean average precision (mAP) exhibited a corresponding increase Fig. 9(c)., whereas simultaneously, the validation and training loss trended downward Fig. 9(a) and (b). This observation underscores the iterative nature of model training, where continued exposure to the dataset enables the model to refine its understanding and improve its predictive capability.

The process of selecting an optimal model requires comprehensive assessment of performance indicators. The highest mean average precision (mAP) was selected as the ideal model for deployment in object detection. We choose mAP50-95 as the indicator Fig. 9(c). This means that the mAP is computed with an intersection over union (IoU) threshold ranging from 0.5 to 0.95. In our training process, the best mAP (50-95) in the training was approximately 0.72 in the 18th epoch. This threshold indicates how similar features the model recognizes the item. For example, if we lower the threshold to 0.5, so as long as the model detect 0.5 of structural similarity, the model will recognize it as the object.

E. Water Level Calculation

Ultimately, the water level can be determined by calculating the pixel distance between the top edge of the previously defined pixel location and the detected waterline location, as shown in Fig. 10.

The water level calculation as Equation (2) shows. First, we need a good quality image that can be perfectly detected in both the water gauge and water line using the model. This is the purpose of obtaining the reference line pixel location

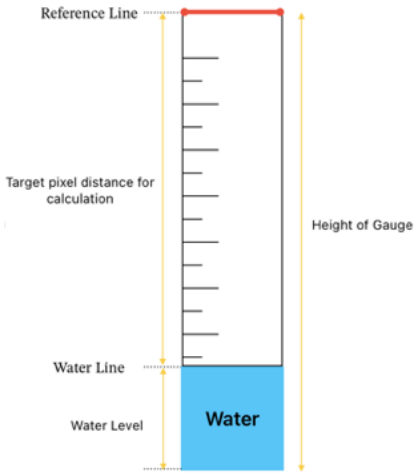

Fig. 10: A sketch of water level gauge. The goal of this project was to accurately detect waterlines. The pixel location of the reference line and the gauge height were fixed in each site.

$y_{reference}$ and the slop β of Equation 2. Because all other values are known, we only need to detect the pixel y location of the water line $y_{water\ line}$ to calculate the water level.

$$Level = H_{water\ gauge} - (y_{reference} - y_{water\ line}) \times \beta \quad (2)$$

Our method relies heavily on the YOLOv8n algorithm; therefore, its accuracy in detecting objects under various conditions is important. To assess its effectiveness, we conducted tests on images captured under extreme conditions, such as blurriness and a low-light environment.

III. Result and Discussions

A. Result

Examples of detection results as Fig. 11-15 shows. During the measurement test process, we set four levels and captured images under three different conditions. The results are shown in Table 1-3 shown. It is worth noting that to simplify the experiment, we took reference readings directly from the ruler using eye observation. And we made sure the waterline is exactly lie on the scale mark.

The results show that when the lighting and visibility of the water level gauge are good, the model can accurately detect the water level with an error of less than 0.5% of the full-scale (400 mm). The error increased when the brightness was reduced, and the images were blurred. The results also showed that the lighting condition had the greatest influence on the measurement accuracy. However, because waterline detection still works under extreme conditions, even though it is not perfect, we can say that the algorithm is feasible in most cases. In real-life applications, an adaptive threshold can be introduced to address changes in the lighting and location. For example, we set the threshold to lower at night while setting it higher during the daytime. However, as our project is still in the experimental

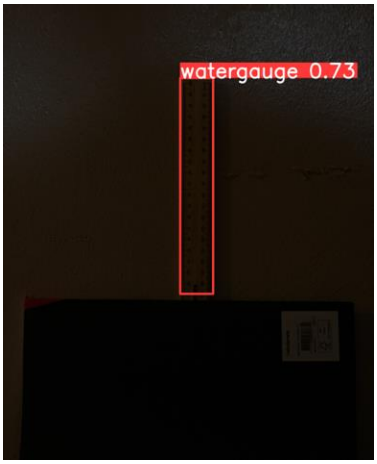

Fig. 11: Brightness: 15%. Although the brightness is low, with a good quality photo, the model can accurately locate the water gauge with a high threshold. However, the waterline drifted slightly.

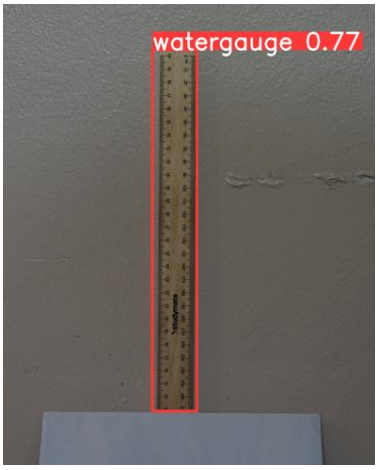

Fig. 12: Brightness: 60%. With the good quality of the photo and good lighting conditions, the model accurately drew a line between the water and the rule.

Fig. 13: Brightness: 60%. The photo was purposely out of focus; this aim was to simulate some climate events such as rain and fog. Although the model could easily detect the ruler, the waterline was not at the correct location.

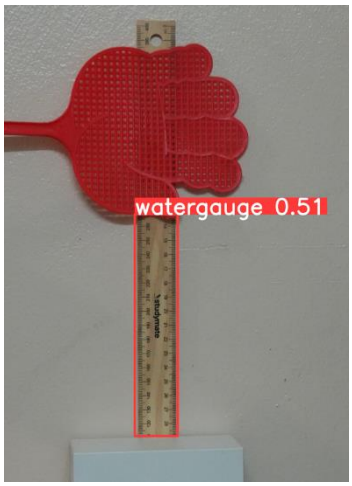

Fig. 14: Brightness: 100%. We simulated an occlusion condition by using an object to block the ruler. Although the water gauge was not fully detected, the water line was perfectly identified, which did not affect the results.

Fig. 15: Brightness: 5%, This is an extreme example; with very bad lighting and poor photo quality, we adjust the threshold to a very low value to detect the gauge. Such a low threshold is not advisable because unwanted objects can be detected in real-world applications.

TABLE 1: 60% BRIGHTNESS

Actual Level (mm)	Measurement	Error (%)
146	146.22	0.05%
130	130.19	0.05%
90	90.89	0.22%
50	50.2	0.05%

TABLE 2: 15% BRIGHTNESS

Actual Level (mm)	Measurement	Error (%)
146	151.1	1.28%
130	133.9	0.97%
90	97	1.75%
50	57	1.75%

TABLE 3: 60 % BRIGHTNESS WITH BLURRING IMAGES

Actual Level (mm)	Measurement	Error (%)
146	148.72	0.68%
130	135.69	1.42%
90	92.8	0.70%
50	51.7	0.42%

phase, to establish a comparison with others, we assumed that our target river has a water gauge of 10 m full scale, so our uncertainty could be in the range of 0.005m to 0.175m. In this situation, when lighting conditions are acceptable, it can outperform other traditional water level methods [8, 9, 12]. Additionally, compared to the training process, the speed of each measurement was very fast, every single measurement only cost 0.5 seconds, this is very useful when deploying in real time applications. Moreover, the measurement result tended to be higher than the actual result, which might be the result of the design nature of our method. However, in real applications, as the environment becomes more complex, other possibilities may be encountered.

The results of our experiment demonstrate that the proposed method can potentially replace conventional water level measurement methods and reduce the complexity of traditional computer vision-based water level measurements. It can provide instantaneous and wide-ranging water level measurements for flood prediction.

B. Discussion

This study exploited the benefits of deep learning for water level detection. After employing photogrammetric transformation, the water level was determined based on the pixel information detected by the training model. The case study illustrates that this method can effectively monitor the water level and track changes throughout the day and partly at night when light conditions are above a certain level.

The proposed method had several advantages. First, unlike other camera-based water level measurement techniques that require numerous image processing steps, such as binary processing, edge detection to detect waterlines, order-statistic filtering for adaptive thresholding, and NCC and median filtering to remove random noise [13-15]. It only requires the YOLOv8n object detection algorithm to train a reliable model in the first stage. This model can be used repeatedly in subsequent applications, thereby saving considerable time and costs. Secondly, the proposed measurement method has several benefits over ultrasonic and capacitor sensors, which require higher equipment costs and are more sensitive to environmental background data changes such as temperature and humidity. In addition, major on-site maintenance of this method only includes cleaning the surroundings of the water gauge and camera lens. Other calibration processes can easily be conducted indoors. Therefore, the maintenance process is simple and less laborious. Third, because the detection process of the proposed method is simple and fast, it does not require significant computing power. Therefore, the detection process can run in microcontrollers with small CPU, such as the Raspberry PI, which can potentially replace some Internet of Things (IoT) applications that require installing a large number of sensors for different measurements using a potentially wide range of computer vision applications [5]. Fourth, because the method is algorithm-based, with the advancement of deep learning algorithms, a substantially improved version of the algorithm can be easily applied to the future development of this method. Consequently, this method demonstrates an excellent potential for future use in measuring water levels.

However, this method also has several disadvantages. One significant disadvantage of this method is that it relies strongly on the deep learning algorithm; if detection fails, it will not be able to identify the water level or miscalculate the level. To improve this, we can first increase the number of labelled images and ensure that they cover all environmental conditions. In addition, exploring alternative labelling techniques such as active learning or semi-supervised learning [33] could offer opportunities for optimizing the efficiency of the labelling process. Furthermore, we present additional algorithms for this method. For instance, in our formula for calculating the water level gauge, we assumed that it is linear; however, the transformation may not be fully accurate in some cases, such as the movement of the water level gauge. A study using algorithms like K-mean clustering, fuzzy recognition and Artificial Neural Network to improve detection accuracy in force measurement [34]; therefore, we can explore a similar method to improve the accuracy of our measurement.

IV. Conclusion

A. Summary

Our study demonstrated the application of the deep learning algorithm YOLO for computer vision-based water level measurements. The objective is to present a cost-efficient, accurate, and easy distribution method that can make use of an already installed site monitoring camera to detect the water level in most conditions. In the model training process, we labelled approximately 50 images to train the model, and selected the best model according to the mAP. During the measurement process, only two major steps were conducted: T Matrix transformation, water gauge detection, and water level calculation. The results of this method show promising and accurate measurements of water levels in challenging environments. This method can potentially provide real time water level monitoring in hydrological sites, such as rivers, lakes, and reservoirs. As it is computed vision-based, we can integrate this method with already installed real-time monitoring platforms, which do not require additional installation costs.

B. Limitation

However, this approach has several limitations. Labelling and calibration must be performed for each new location, which may require a large amount of time and data in a complex environment. The performance of the object detection algorithm relies on image quality and lighting conditions. Approaches to improve night-version image capture are required. Further research can be conducted on

night version improvements, such as the application of infrared cameras, implementation of error correction algorithms. Moreover, in our study, we explored changes in lighting and camera blurring. Environmental conditions can be more complex, and future studies can explore more environmental variables, such as reflection on water and wildlife activities.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] "HUMAN DEVELOPMENT REPORT 2021/2022," The United Nation, 2022. [Online]. Available: https://hdr.undp.org/content/human-development-report-2021-22 (accessed on 01 April 2024)

[2] S. Y. Schreider, D. Smith, and A. Jakeman, "Climate change impacts on urban flooding," *Climatic Change,* vol. 47, pp. 91-115, 2000.

[3] U. Iqbal, P. Perez, W. Li, and J. Barthelemy, "How computer vision can facilitate flood management: A systematic review," *International Journal of Disaster Risk Reduction,* vol. 53, p. 102030, 2021/02/01/ 2021, doi: https://doi.org/10.1016/j.ijdrr.2020.102030.

[4] H. S. Munawar, A. W. A. Hammad, and S. T. Waller, "Remote Sensing Methods for Flood Prediction: A Review," *Sensors,* vol. 22, no. 3, p. 960, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/3/960.

[5] B. Arshad, R. Ogie, J. Barthelemy, B. Pradhan, N. Verstaevel, and P. Perez, "Computer Vision and IoT-Based Sensors in Flood Monitoring and Mapping: A Systematic Review," *Sensors,* vol. 19, no. 22, p. 5012, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/22/5012.

[6] M. Tourian *et al.*, "Spatiotemporal densification of river water level time series by multimission satellite altimetry," *Water Resources Research,* vol. 52, no. 2, pp. 1140-1159, 2016.

[7] C. Yuan, P. Gong, and Y. Bai, "Performance Assessment of ICESat-2 Laser Altimeter Data for Water-Level Measurement over Lakes and Reservoirs in China," *Remote Sensing,* vol. 12, no. 5, p. 770, 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/5/770.

[8] K. Loizou and E. Koutroulis, "Water level sensing: State of the art review and performance evaluation of a low-cost measurement system," *Measurement,* vol. 89, pp. 204-214, 2016.

[9] D. B. Duraibabu, G. Leen, D. Toal, T. Newe, E. Lewis, and G. Dooly, "Underwater Depth and Temperature Sensing Based on Fiber Optic Technology for Marine and Fresh Water Applications," *Sensors,* vol. 17, no. 6, p. 1228, 2017. [Online]. Available: https://www.mdpi.com/1424-8220/17/6/1228.

[10] A. Kruger, W. F. Krajewski, J. J. Niemeier, D. L. Ceynar, and R. Goska, "Bridge-Mounted River Stage Sensors (BMRSS)," *IEEE Access,* vol. 4, pp. 8948-8966, 2016, doi: 10.1109/ACCESS.2016.2631172.

[11] I. Bae and U. Ji, "Outlier detection and smoothing process for water level data measured by ultrasonic sensor in stream flows," *Water,* vol. 11, no. 5, p. 951, 2019.

[12] J. D. Paul, W. Buytaert, and N. Sah, "A technical evaluation of lidar-based measurement of river water levels," *Water Resources Research,* vol. 56, no. 4, p. e2019WR026810, 2020.

[13] Y.-T. Lin, Y.-C. Lin, and J.-Y. Han, "Automatic water-level detection using single-camera images with varied poses," *Measurement,* vol. 127, pp. 167-174, 2018/10/01/ 2018, doi: https://doi.org/10.1016/j.measurement.2018.05.100.

[14] L.-C. Kuo and C.-C. Tai, "Robust Image-Based Water-Level Estimation Using Single-Camera Monitoring," *IEEE transactions on instrumentation and measurement,* vol. 71, pp. 1-11, 2022, doi: 10.1109/TIM.2022.3161691.

[15] Z. Zhang, Y. Zhou, H. Liu, and H. Gao, "In-situ water level measurement using NIR-imaging video camera," *Flow Measurement and Instrumentation,* vol. 67, pp. 95-106, 2019/06/01/ 2019, doi: https://doi.org/10.1016/j.flowmeasinst.2019.04.004.

[16] H. Rizk, Y. Nishimur, H. Yamaguchi, and T. Higashino, "Drone-Based Water Level Detection in Flood Disasters," *International Journal of Environmental Research and Public Health,* vol. 19, no. 1, p. 237, 2022. [Online]. Available: https://www.mdpi.com/1660-4601/19/1/237.

[17] F. Lin, Z. Yu, Q. Jin, and A. You, "Semantic segmentation and scale recognition–based water-level monitoring algorithm," *Journal of Coastal Research,* vol. 105, no. SI, pp. 185-189, 2020.

[18] N. H. Jafari, X. Li, Q. Chen, C.-Y. Le, L. P. Betzer, and Y. Liang, "Real-time water level monitoring using live cameras and computer vision techniques," *Computers & Geosciences,* vol. 147, p. 104642, 2021/02/01/ 2021, doi: https://doi.org/10.1016/j.cageo.2020.104642.

[19] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580-587.

[20] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440-1448.

[21] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems,* vol. 28, 2015.

[22] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961-2969.

[23] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117-2125.

[24] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE,* vol. 111, no. 3, pp. 257-276, 2023.

[25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779-788.

[26] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263-7271.

[27] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767,* 2018.

[28] J. Terven and D. Cordova-Esparza, "A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond," *arXiv preprint arXiv:2304.00501,* 2023.

[29] "YOLOv8—Ultralytics YOLOv8 Documentation." Ultralytics, 2024. [Online]. Available: https://docs.ultralytics.com/models/yolov8/ (accessed on 30 April 2024).

[30] J. Yu and H. Hahn, "Remote Detection and Monitoring of a Water Level Using Narrow Band Channel," *J. Inf. Sci. Eng.,* vol. 26, no. 1, pp. 71-82, 2010.

[31] D. Zhang and J. Tong, "Robust water level measurement method based on computer vision," *Journal of Hydrology,* vol. 620, p. 129456, 2023.

[32] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems,* vol. 32, 2019.

[33] M. Gao, Z. Zhang, G. Yu, S. Ö. Arık, L. S. Davis, and T. Pfister, "Consistency-based semi-supervised active learning: Towards minimizing labeling cost," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, 2020: Springer, pp. 510-526.

[34] J. Ye, Z. Lin, J. You, S. Huang, and H. Wu, "Inconsistency Calibrating Algorithms for Large Scale Piezoresistive Electronic Skin," *Micromachines,* vol. 11, no. 2, p. 162, 2020. [Online]. Available: https://www.mdpi.com/2072-666X/11/2/1