Low-Cost Visual Sensing of Stormwater Outflow

NCDOT Project 2022-04
FHWA/NC/2022-04
April 2025

François Birgand[1], Ph.D., Mohammad Motlagh Nooshzadi
Sierra Young[2], Ph.D.
[1]Department of Biological & Agricultural Engineering
North Carolina State University
[2]Civil and Environmental Engineering Department
Utah State University

NC STATE UNIVERSITY Bio&Ag ENGINEERING

STATE OF NORTH CAROLINA DEPARTMENT OF TRANSPORTATION

RESEARCH & DEVELOPMENT

Technical Report Documentation Page

1. Report No. **FHWA/NC/2022-04**	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle **Low-Cost Visual Sensing of Stormwater Outflow**		5. Report Date **April 30, 2025** 6. Performing Organization Code
7. Author(s) **François Birgand, Mohammad Motlagh Nooshzadi, Sierra Young**		8. Performing Organization Report No.
9. Performing Organization Name and Address **Department of Biological & Agricultural Engineering, NC State University 3110 Faucette Drive, Raleigh, NC 27695-7625** **Sierra Young, Ph.D.** **Department of Civil and Environmental Engineering** **Utah State University**		10. Work Unit No. (TRAIS) 11. Contract or Grant No.
12. Sponsoring Agency Name and Address **Research and Development Unit** **1549 Mail Service Center** **Raleigh, North Carolina 27699-1549**		13. Type of Report and Period Covered Final **Report** **Jan. 01, 2022 to Dec. 31, 2024** 14. Sponsoring Agency Code **RP2022-04**
Supplementary Notes: **François Birgand, Ph.D.** https://orcid.org/0000-0002-5366-1166		

16. Abstract

Stormwater systems serve as critical urban drainage infrastructure, maintaining municipal functionality and playing key hydrological and environmental roles. Yet, stormwater outlets are not designed for stormflow monitoring, making traditional contact-based sensors unsuitable for gauging highly variable and turbulent flows. To address this, we propose a computer vision approach that quantifies discharge from near-range images and videos of the outlet. Given the challenges of applying traditional computer vision in natural environments with variable lighting and environmental noise, our method employs a combination of machine learning (Mask R-CNN and YOLO8 models) and computer vision (CV-ML) techniques, leveraging the recognizable geometrical shape of culverts in otherwise very noisy images. Specifically, water stage is obtained through subtracting the extracted shape of the round culvert from the height of empty area above water. Subsequently, image-based measurements are transformed into real-world units by applying a homography transformation calibrated using a checkerboard reference object placed parallel to the outlet. Evaluation on a culvert of known dimensions demonstrated ± 1 cm accuracy on water stage for approximately 80% of general measurements. For water stages evaluated under calm flows, the method estimated more than 80% of stages within ± 0.5 cm, and under turbulent flows, the method estimated 63% of values within ± 1 cm (96% within ± 2 cm). These results show great promise in the use of image-based techniques in difficult conditions where no traditional techniques are applicable. They also are the prerequisite for estimations of discharge,which remains the focus of ongoing development under subsequent NCDOT-supported research. Metrology of the system conducted in the lab, showed that under well-lit conditions, the practical distance at which the majority of measurements had an error range within ±1 cm was found to be 8 meters for the Mask R-CNN model (corresponding to an object pixel resolution of 0.2 cm/px) and 6.5 meters for the YOLOv8 model (corresponding to an object pixel resolution of 0.15 cm/px). However, under dark conditions, practical distances were smaller. The YOLOv8 model also showed greater susceptibility to errors caused by the lighting pattern from the camera over the outlet edges.

17. Key Words *Stormflow monitoring; Image-based sensing; Computer vision; Machine Learning models*	18. Distribution Statement		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 59	22. Price

Form DOT F 1700.7 (8-72) **Reproduction of completed page authorized**

DISCLAIMER

The contents of this report reflect the views of the author(s) and not necessarily the views of the University. The author(s) are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of either the North Carolina Department of Transportation or the Federal Highway Administration at the time of publication. This report does not constitute a standard, specification, or regulation.

Acknowledgement

The writers of this report would like to sincerely thank John Kirby, Ryan Mullins, Curtis Bradley and Andy McDaniel from NC DOT for their support throughout this project.

They would also like to thank Mr. Nooshzadi's committee. In particular, Dr. Kenneth Chapman, for his consistent support and helpful comments throughout the project. They would also like to express their appreciation to Dr. Lirong Xiang, and Dr. Edgar Lobaton, for his contribution despite joining late in the committee as he brought valuable suggestions.

The writers would also like to express their gratitude to Dr. Roarke Horstmeyer and Dr. Juan Matias DiMartino, instructors in the Biomedical Engineering and Electrical and Computer Engineering Departments, at Duke University.

The writers would also like to thank Evelynn Wilcox, biological and agricultural engineering undergraduate student, for her invaluable contributions to the fieldwork and laboratory tests. Managing the fieldwork, lab tests, and extensive coding aspects of this project was a significant responsibility, and without her contributions, it would have been much more difficult to complete.

Lastly, they would like to thank the Biological and Agricultural Department staff, especially Heather Austin and Lacy Parrish, for their dedicated, high-quality work and unwavering support.

Table of Contents

1 Introduction 10

2 Methods 12

2.1 Deep learning models used 12

2.1.1 Reference deep learning model: Mask R-CNN 13

2.1.2 YOLOv8: a lightweight deep learning alternative 18

2.2 Model Evaluations 19

2.2.1 Coordinate Transformation 20

2.3 Ellipse Fitting of an empty outlet 23

2.4 Models' water stage measurement comparison in the field 25

2.4.1 Cameras used and field sites 25

2.4.2 Performance assessment of the system: measuring outlet diameter in the field 25

2.4.3 Performance assessment of the system: measuring water depth 26

2.5 System's metrology in the lab 28

3 Results 30

3.1 System's performance in the field: results on culvert diameters 30

3.2 System's performance in the field: results on water levels 35

3.3 Metrology of the system in the lab 38

3.3.1 Detection of culvert and accurate measurements stop when camera is too far 39

3.3.2 Relative measurement error on diameter increased with image distortion, distance, YOLOv8 model and dark images 41

4 Findings and Conclusion 45

5 Recommendations 46

6 Implementation and Technology Transfer 47

7 Cited References 47

List of Figures

Figure 1: Representative sample images, along with their annotations, from various sites at North Carolina State University (NCSU) and surrounding areas: (a) Softball Field site; (b) Centennial Campus site; (c) Edward Mills Road site; (d) Motorpool site.....13

Figure 2: The models capture the empty outlet (a) and the empty area of the culvert (b). Subtracting these two areas gives the flow area, which is marked in blue in (c) 14

Figure 3: Mask R-CNN architecture (reference: Sky Engine AI Developer Blog) 17

Figure 4: An example of a false positive detection: the Mask R-CNN model identifies both a culvert and its reflection in a pond.....21

Figure 5: Calibration setup with reference object (chessboard pattern) embedded and aligned with the stormwater outlet 21

Figure 6: An illustration of the camera's pinhole model used in this study (Zhang 2004; Tomasi 2017) 23

Figure 7: Horizontal perspective lines over the unoccupied area facilitating visual observation27

Figure 8: Laboratory setup for metrology testing: (a) prototype culvert with adjustable legs for mobility; (b) camera mounted on a pole attached to a rail above the setup. Arrows along the rail indicate the direction of camera movement to achieve different distances. Arrows on the pole indicate the direction of camera movement to obtain different vertical angles; (c) culvert positions marked on the floor, with numbers indicating wheel locations for each position and the arrow representing the direction of movement 29

Figure 9: Comparison of major and minor axis lengths in image coordinates (top) and real-world units (bottom). Red points represent Mask R-CNN model results, blue points represent YOLOv8 model results. 32

Figure 10: Models error range for the major and minor axes measurements. Mask R-CNN: a) and b). YOLOv8: c) and d) 34

Figure 11: Inference time comparison between the YOLOv8 and theMask R-CNN 34

Figure 12: Comparison of Mask R-CNN and YOLOv8 models with visual measurements for water stage estimation at the stormwater outlet. 36

Figure 13: Comparison of models error ranges relative to visually measured water stages.....37

Figure 14: The YOLOv8 model detected the unoccupied area of the culvert (left) compared to theMask R-CNN model (right) 37

Figure 15: Mean resolution of the culvert's representation at various distances 39

Figure 16: Detection of the culvert in the lab as a function of the distance for the Mask R-CNN (top row) and the Yolov8 model (bottom row), for images during daylight (left column) and night images (right columns) 40

Figure 17: Error on estimations of the culvert diameter via the size of the major axis of the ellipse appearing on images taken in the lab as a function of the distance of the camera to the culvert. Top row: Mask R-CNN model; Bottom row: Yolov8 model; Left column: daylight images; Right column: dark images.....42

Figure 18: Representative samples from the YOLOv8 model detections for dark images: (a) Edge detection bleeding (distance: 5.25 meters). (b) Model detected the outer edge instead of the inner edge (distance: 7 meters). 42

List of Tables

Table 1: Average precision for models' mask detections 19
Table 2: Mean relative error for major and minor axes measurements during daytime and nighttime. Values in red correspond to the Mask R-CNN model, and values in blue correspond to the YOLOv8 model. 32
Table 3: Practical distance limits and corresponding object pixel resolutions for both models under light and dark conditions, based on lab tests 43

List of Abbreviations

AP: Average Precision

CC: Creative Commons

CI/CD: continuous integration and continuous delivery

CNN: Convolutional Neural Network

DNN: Deep Neural Network

fps: frames per second

GPU: Graphics Processing Unit

IOU: Intersection over Union

mAP: mean Average Precision

Mask R-CNN: Mask Region-based Convolutional Neural Network

NC DOT: North Carolina Department of Transportation

NCSU: North Carolina State University

OS: Operating System

ReLU: Rectified Linear Unit

ROI: Region of Interest

RPN: Region Proposal Network

SOTA: State of the Art

US EPA: United State Environmental Protection Agency

YOLOv8: “You Look Only Once version 8”

1 Introduction

Increased urbanization around the world comes with less pervious surfaces and higher peaks of stormwater outflow following rainfall. Detrimental consequences include increased flooding, stream bank erosion, and pollutant loads among many others (Tsihrintzis and Hamid 1997; US EPA 2015; Winston R. J. and Hunt W. F. 2017; Müller et al. 2020; Kriech and Osborn 2022). Many stormwater control measures have been designed and implemented in the field to mitigate detrimental stormwater effects (reviewed by Prudencio and Null (2018)). In urban environments, there are numerous stormwater outlets where installing and maintaining traditional sensors to calculate flow is difficult and expensive. Image-based methods offer the possibility for a more accessible, cost-effective, and possibly more accurate alternative, although it comes with its own challenges.

In hydrology, computer vision has been used for measuring water level and water surface velocity (e.g., Birgand et al. 2013; Chakravarthy et al. 2002; Kaplan et al. 2019; Lin et al. 2018; Noto et al. 2022; Schoener Gerhard 2018; Takagi et al. 1998; Jeanbourquin et al. 2011; Jodeau et al. 2008; Kantoush et al. 2011; Fujita et al. 1998; Wu et al. 2019; Kim and Kim 2020; Fujita et al. 2019; Engelen et al. 2018; Holland et al. 2001; Bradley et al. 2002; Creutin et al. 2003; Hauet et al. 2008; Muste et al. 2008), and images are being used as an active monitoring tool (e.g., USGS 2022; Birgand et al. 2022). Image-based measurements are then used to estimate discharge, i.e., the volume of water passed by a point per unit time (e.g., Hauet Alexandre et al. 2008; Le Coz et al. 2010; Peña-Haro et al. 2021; Le Coz et al. 2021; Chahrour et al. 2021; Tsubaki et al. 2011; Bechle Adam J. et al. 59 2012; Ji et al. 2020; Zhao et al. 2021). Image-based methods offer additional benefits over traditional techniques, including non-contact sensing, access to the velocity field at the surface of the water, access to additional information about environmental conditions, visual verification, access to the 'raw' data, and openness to reanalyzing images using improved algorithms and developments (Hauet Alexandre et al. 2008; Birgand et al. 2022; Chapman et al. 2022; Zhang et al. 2019b; Eltner et al. 2021). Additionally, the development of communication networks has opened new possibilities to the field, such as the possibility of distant data interpretation or cloud computing (Pan et al. 2018; Yu and Hahn 2010). This subsequently obviates the need for field calibration and high-level maintenance at short periodicity, requiring fewer field maintenance visits from high-skill personnel (Pan et al. 2018).

Images have typically been used to measure water stages and videos to estimate surface velocity of the water. Traditional machine vision techniques (i.e., not using machine learning) have classically been used to measure water level in relatively calm waters (Fujita et al. 1998; Takagi et al. 1998; Bradley et al. 2002; Chakravarthy et al. 2002; Creutin et al. 2003; Fujita et al. 2007; Iwahashi and Udomsiri 2007;

Jodeau et al. 2008; Hauet Alexandre et al. 2008; Kim et al. 2008; Iwahashi et al. 2007). Most studies report values within ±10 mm (Nguyen et al. 2009; Kim et al. 2011; Hies et al. 2012; Lin et al. 2018; Pan et al. 2018; Zhang et al. 2019c; Hansen et al. 2017), although Birgand et al. (2022) reported uncertainty to be within ±3mm 70% and ±5mm 90% of the time in the field. Two categories of methods are currently used for measuring water surface velocity based on images: motion estimation and feature-point tracking (Jeanbourquin et al. 2011). Deep learning models have also been used to measure the velocity of water in coastal areas, based on principles similar to motion estimation methods (Kim and Kim 2020).

Given the highly variable contextual conditions in field images, deep learning approaches may be a more suitable choice for water stage monitoring, compared to classical computer vision techniques. Not surprisingly, deep learning approaches have been reported in image-based hydrological monitoring. Pan et al. (2018) reported on a deep learning system for water level detection and surveillance. Gupta et al. (2022) proposed a ranking system for stream stages based on convolutional neural networks (CNN).

Stormwater is often routed in circular pipes and culverts, until it is discharged into receiving bodies, i.e., often directly into the receiving streams. In large pipes, images and videos have been recorded from inside the pipes for monitoring (Nguyen et al. 2009; Jeanbourquin et al. 2011; Haurum et al. 2020; Ji et al. 2020). Most stormwater outlets are too small for such applications. However, because of their circular pattern, culverts and pipes have the potential to be automatically recognized using machine vision approaches, from images taken from cameras outside and facing the conduits in field conditions.

Deep learning models thus offer a great potential to address many of the theoretical difficulties of monitoring stormwater outflow. However, AI-based models have high computational intensities and requirements, which is not compatible with edge devices equipped with relatively low computational capabilities. Edge devices are associated with edge computing where raw data are analyzed *in situ* by mini-computers embedded in the sensors, as opposed to cloud computing where the raw data are sent to the cloud for analysis. Unless one can show that much lighter models can provide reliable results, the application of deep learning models for hydrological monitoring and other high-precision monitoring may be compromised on edge devices.

In this report, we explore the use of two deep learning methods to automatically measure the water stage at the mouth of stormwater outfalls from images taken by inexpensive time lapse cameras. Our objective is to assess how a computationally light model that could be run on edge devices, compares to a reference computationally intensive model. Our reference model is a deep learning model based on the Mask R-CNN architecture (He et al. 2017). Our light model is a deep learning model based on the YOLOv8 architecture (Jocher et al. 2023). We use indicators of performance, i.e., dimensions of culvert

diameter and water levels read manually on images in the field, and computational times for our assessment.

2 Methods

Deep learning models applied to images and videos require several steps to train. It is important to explain the process in simple terms, illustrating why these models are complex and computationally demanding. In our stormwater monitoring application, we train models for two cases: culverts with and without water. For the first case, we manually highlight the empty area between the water level and the culvert's top, and tell the model, "This is a culvert with water". In the second case, we highlight the entire empty culvert and tell the model, 'This is an empty culvert.' With enough images from various sites, angles, and lighting conditions, the deep learning models learn to automatically recognize culverts with and without water in most new images (outside of the training set). The models initially provide results in pixel coordinates, which can be converted into real-world measurements.

Below, we provide details for our Mask R-CNN and YOLOv8 models. We also outline our approach for comparing the performance of these two models.

2.1 Deep learning models used

The next step involving transforming measurements from the image coordinate system into a real-world coordinate system is presented afterwards. Given the utilization of deep learning structures in the models developed in this study, it is imperative to train them using a diverse dataset representing real-world scenes of stormwater structures and outflows. Generally, the training dataset includes images with detailed information on the location of the features the model aims to detect. Practically in our case, the training dataset consisted of images from cameras looking at stormwater outlets and the manually labeled 'culverts with and without water' information. This necessary process, which is referred to as labeling or annotation, is labor-intensive and requires human observers manually marking the features over images through annotation software. In this study, the image dataset used to train the network was collected from three different sites on NC State University campus in Raleigh, North Carolina. The choice was guided by the short distance from our laboratory for maintenance and observations. The dataset also includes images of a lab stormwater outlet prototype, as well as images downloaded through Google Images licensed under a Creative Commons (CC) agreement [1](Figure 1). The image annotation was carried out using the

[1] devoted to educational access and free of charge to the public

RectLabel software given its ease of use, robustness, and capability to provide the annotation files for various machine learning and deep learning architectures.

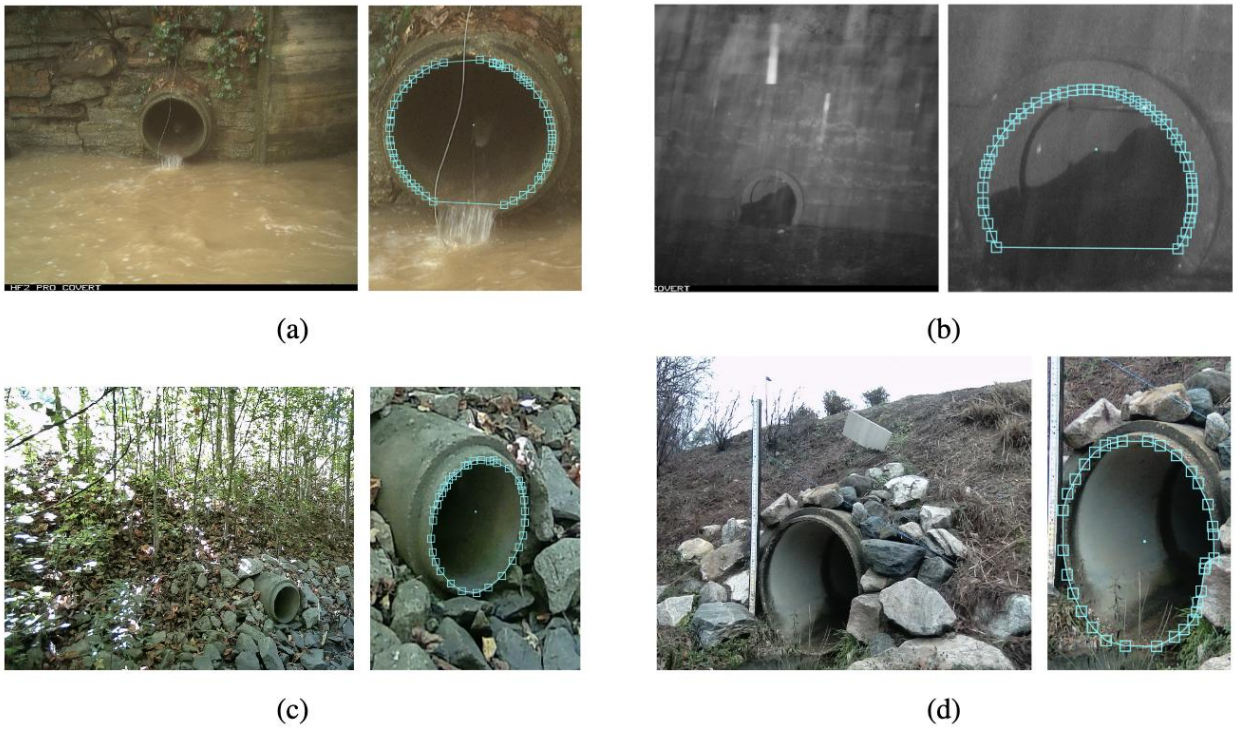


Figure 1: Representative sample images, along with their annotations, from various sites at North Carolina State University (NCSU) and surrounding areas: (a) Softball Field site; (b) Centennial Campus site; (c) Edward Mills Road site; (d) Motorpool site.

When training deep learning models, training data are used as inputs to optimization algorithms (like gradient descent variants) to tune model parameters. This aims to minimize the model's error in recognizing features, a concept known in machine learning terminology as empirical risk. Within this approach, the empirical risk relationship, which is typically defined as the mean of errors, or losses, for every training data, could be viewed as the signal generator which is dispatched through the network and changes the parameters to the extent of the received signal strength.

2.1.1 Reference deep learning model: Mask R-CNN

The reference model was developed using the Mask R-CNN architecture, an instance segmentation framework that identifies and outlines object boundaries as defined during the annotation process (He et al. 2017). Using this model, the training data was labeled so that the model could delineate

the unoccupied area of the culvert within its inner boundaries when there was an outflow from the system, and the entire inner boundary of the culvert when there was no flow (Figure 2). As a result, subtracting the mask during the flow event from the mask corresponding to the no-flow condition gave the area occupied by the flow at the outlet.

(a) (b) (c)

Figure 2: The models capture the empty outlet (a) and the empty area of the culvert (b). Subtracting these two areas gives the flow area, which is marked in blue in (c)

The Mask R-CNN architecture is a state-of-the-art (SOTA) instance segmentation and pose detection model and was originally introduced as an extension to the Faster R-CNN architecture (He et al. 2017; Ren et al. 2017). This structure is composed of a cascade of components and provides three types of outputs: object masks which delineate the boundaries of the objects present in the scene; localization information in the format of rectangular areas tightly encompassing the objects; and classification information for each of the detections. Upon unraveling the input data into a vector format, which is a common first step with computer vision deep learning structures, the Mask R-CNN architecture initially processes the data through a feature extractor network[2]. The feature extractor network is a deep neural network (DNN) that compresses the input image into something called a feature map. The initial layers of the feature extractor network handle low-level, general features, like the edges, while the deeper layers

[2] The feature extractor is a model which translates features, ranging from edges and textures to the objects present in the scene, into an abstract representation in a way which is comprehensible for the downstream components responsible for the designated target. This layered model processes images and videos primarily through filters, which can be thought of as windows sweeping across the image area. After each layer, the next layer window sweeps through points obtained from the previous layer's window locations. The actual representation of each window is called the receptive field, indicating the area of the original image considered when generating the output for that layer.

address more high-level features due to the increased receptive field of these layers (Redmon et al. 2016). To better understand the concept of receptive fields, we should focus on how the convolution mechanism operates over two-dimensional inputs.

A fully convolutional neural network (CNN) is composed of multiple layers, with each layer receiving input from the output of the previous layer. Within each layer, multiple filters (matrices of odd dimensionality) slide over the input, performing element-wise multiplication with the corresponding region of the input and summing the results. This result then replaces the central value of the filter's position over the input. Consequently, as an image passes through the layers, its dimensions shrink, but each value represents a larger region of the original input. In computer vision, this region is known as the receptive field.

The second component of the Mask R-CNN architecture is the region proposal network (RPN), which is responsible for generating bounding boxes that likely contain objects. The idea of using bounding boxes with the possibility to include objects is common among the object detection and segmentation models. The following components of the models refine these regions to accurately locate and segment the objects. Before the introduction of the RPN, networks relied on either external, non-integrated components for region proposals or a grid-based approach. While external bounding box generators were functional, they lacked adaptability to the specific characteristics of the training data. This often led to the generation of an excessive number of boxes, necessary to achieve sufficient precision in object detection. Consequently, the computational overhead increased, slowing down the entire process (Ren et al. 2017). The grid-based approach, while faster, lacked the precision necessary for accurate object detection, making it unsuitable for instance segmentation applications (Redmon and Farhadi 2017). In contrast, the RPN is trained alongside the rest of the network, allowing its bounding box proposals to adapt to the training data. This leads to more accurate proposals, enabling the network to achieve the desired precision with fewer boxes and less computation. As a result, this significantly improves the network's overall response time. The clear advantage of the RPN is evident in the performance comparison between Fast R-CNN and Faster R-CNN. While Fast R-CNN, relying on the selective search method with 2,000 region proposals, Ren et al. (2017) reported Fast R-CNN to achieve a mean average precision (mAP) of 70%, its frame rate was 0.5 frames per second (fps). In contrast, Faster R-CNN, leveraging the RPN for proposal generation, required only 300 proposals to achieve a mAP of 73.2%. Remarkably, it also achieved a

significantly faster frame rate of 17 fps (Ren et al. 2017). The concept of using a dedicated component for generating region proposals proved so influential that it was adopted by other architectures, like YOLO, due to its efficiency and precision (Redmon and Farhadi 2017).

The Mask R-CNN architecture's third and key innovation is the Region of Interest (ROI) Align layer. Its predecessor, Faster R-CNN, relied on pooling operations like max pooling to downsample region proposals. This involved dividing the ROIs into a grid and selecting the maximum value within each cell, introducing localization errors. While tolerable for object detection tasks focused on bounding boxes, these errors are detrimental to segmentation, which requires pixel-level accuracy. To address this, the ROI Align layer divides each grid cell into four quadrants and samples a point within each quadrant. The value at each sampled point is then calculated using bilinear interpolation through the four nearest neighboring feature map points. This method significantly reduces information loss and error compared to traditional pooling, leading to a measurable improvement in segmentation accuracy (He et al. 2017).

Lastly, the outputs of the Mask R-CNN models are provided through 3 distinct heads (Figure 3), namely the classification head, the localization head, and the mask head. While geometrically representing the shape of heads, they are actually fully connected neural network layers which provide the outputs in the requested format. By comparison, while convolutional layers use filters and process localized regions of the input data to generate output, the operation within the fully connected layers are carried out through matrix multiplication. Additionally, fully connected layers frequently incorporate a non-linear activation function, like the Rectified Linear Unit (ReLU), to introduce non-linearities essential for modeling complex relationships in the data. While the output class and the object bounding boxes appear to be designed for the output mask, the computations internally happen the opposite way. Masks are generated for each of the object classes, and detections with a score higher than a threshold are reported as the output. This is done after the non-maximum suppression process, which eliminates detections that overlap by more than a certain percentage.

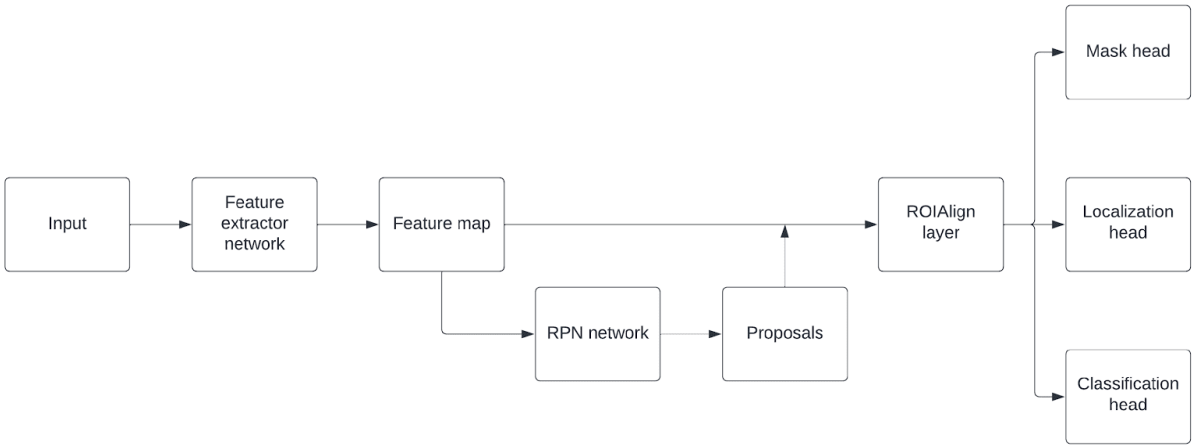


Figure 3: Mask R-CNN architecture (reference: Sky Engine AI Developer Blog - https://www.skyengine.ai/blog/what-is-mask-r-cnn.)

The Mask R-CNN model, despite its robust detection capabilities, is computationally intensive and requires the use of a Graphics Processing Unit (GPU) during inference. Furthermore, due to its specific structure, there are challenges in packaging it into formats compatible with edge devices' operating systems, such as the Raspberry Pi OS, as well as smartphone operating systems like Android and iOS. To make our model practical for real-world applications, in our case the stormwater flow monitoring, two options are available. First, the model could be deployed over a cloud service like Google Vertex, where scalability is assured, and access to computational resources is not a concern. In this scenario, cameras are responsible only for capturing images and videos, and transmitting them to the servers. The advantage of opting for this option is that the system allows for the use of more powerful models, given the available computational support. It also features a simpler design, making the detection of glitches and bottlenecks easier, and is capable of continuous training and improvement of the model, for example, through continuous integration and continuous delivery (CI/CD) pipelines, which are typical in industrial settings leveraging AI at scale (Garg et al. 2021; Garg and Garg 2019). Lastly, safety concerns with this option are minimized, given the centralized design of the computational facilities, which allows for enhanced physical safety measures. Additionally, proprietary connections with the desired level of security can be selected. However, such a design becomes expensive when deployed over third-party facilities. Moreover, domesticating

the facilities requires investments and technical knowledge, which may not be available to water and agricultural engineers.

Another option is to use a lighter model with a similar structure and functionality, which could be deployed individually on the cameras. In this scenario, aside from the typical task of capturing images and videos, the cameras would also be equipped with processing units capable of extracting the water level at the stormwater outlet and transmitting only those values. Using this system has the benefit of saving on communication costs compared to the previous approach and doesn't require an initial investment and high technical expertise. However, with this system, the training and improvement of the models become significantly harder, as the system is not designed for that purpose. Updating cameras with new models is significantly more time-consuming, and the cameras should be treated individually unless a remote updating system is specifically designed, which demands a high level of technical proficiency and can be impeding for the corresponding staff. Lastly, to the best of our knowledge, cameras with such capabilities, which could operate in the field, are not widely available in the market. Therefore, in the case of their widespread deployment, there needs to be an investment for the mass production of these types of cameras. There is a market for these, however, and there is little doubt that these smart cameras will eventually be available.

2.1.2 YOLOv8: a lightweight deep learning alternative

To accommodate the second option, an instance segmentation model based on the YOLOv8 structure was trained (Jocher et al. 2023). The YOLO architecture, with its fully convolutional integral design, offers the advantage of being faster. In the Mask R-CNN architecture, a 4-step alternate training approach serves as a speed bottleneck. From a macroscopic point of view, the issue with the alternate training approach is that it requires going back and forth between components during training, thereby elongating the training process. In contrast, the fully convolutional design of the YOLO structure allows for updating all parameters through a single pass of the gradient signal. Moreover, the lack of design optimization in the Mask R-CNN feature extractors presents a potential speed impediment. Since these extractors were not originally designed for object detection and segmentation, they may slow down the model's application. Another benefit of utilizing the YOLOv8 structure is that the trained model can be easily converted into formats like TFLite and CoreML (Apple Inc. 2021), which are

readable by typical smartphones as well as edge devices. Consequently, the design of cameras becomes easier, and it becomes feasible for the entire process to be implemented through a smartphone application. This enables staff and individuals to carry the model in their pocket and use it with just a few button presses.

2.2 Model Evaluations

After training, the models were evaluated on a test dataset containing images from a site located at NC State University Centennial Campus (image not shown).

Table 1: Average precision for models' mask detections. AP: number of positive (correct) detections divided by the total number of detections

	Intersection Over Union (IOU)	Average Precision (AP)
Mask R-CNN	0.50 : 0.95	0.841
YOLOv8	0.50 : 0.95	0.901

In Table 1 above, IOU is a metric used to quantify the overlap between the model detections and the ground truth. It is also computed through the following relationship:

$$IOU = \frac{A_{intersection}}{A_{union}} \tag{1}$$

The average precision (AP) is a standard metric for object detection and segmentation tasks an is defined as the number of positive (correct) detections divided by the total number of detections. As a result, AP values are indicators of how accurate model detections are.

In equation 1 above, $A_{intersection}$ refers to the area of intersection between the model's detection and the ground truth. Likewise, A_{union} denotes the union of the model's detection and ground truth region. The Precision and Recall are also typical metrics with object detection models, and they were first introduced by the COCO object detection challenge (Lin et al. 2014). To compute these two metrics, the following relationships are employed:

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP+FP} \\ \text{Recall (Sensitivity)} &= \frac{TP}{TP+FN} \end{aligned} \tag{2}$$

In equations 2, TP denotes true positive detections, FN represents false negative detections, and FP stands for false positive detections. It is important to note that true positives correspond to cases where the model's detections were correct for the desired object. In our application, true positive cases are model detections with an IOU higher than a threshold. Conversely, false positives indicate cases where the model made a detection, but the ground truth data indicates no object at that location. An example of a false positive detection in this study is the identification of the culvert's reflection in the ponded water in front of the culvert rather than the actual object (Figure 4). These false positives generate outliers our system automatically detected and flagged as unreliable. Finally, false negatives are cases where the model didn't make any detection, and the ground truth data confirms that choice. For this study, this refers to cases where there is no outlet in the image and the models make no detection. Here, the term "ground truth" refers to the annotations provided to the models, which in our application is either the empty outlet or the unoccupied area of an outlet with flow. Therefore, the recall denominator encompasses all correct cases, while the precision denominator aggregates the entirety of detections. Also, given the explanations, the average precision is computed by taking the mean of precision values at different recall levels, i.e., the area under the precision-recall curve.

2.2.1 Coordinate Transformation

Since image and video measurements exist in pixel coordinates, a geometrical method is needed to convert them to a real-world coordinate system. Projective geometry governs the relationship between real-world objects and their image representations. Specifically, the homography transformation allows us to move between real-world and image coordinates (Hartley and Zisserman 2004). To apply this transformation, the first step is to embed a reference object with known dimensions in the same plane as the object we want to measure. In this study, we used a 4×4 chessboard pattern as the reference object and its 9 inner corner points as known reference coordinates (Figure 5).

Figure 4: An example of a false positive detection: the Mask R-CNN model identifies both a culvert and its reflection in a pond.

Figure 5: Calibration setup with reference object (chessboard pattern) embedded and aligned with the stormwater outlet

Projective geometry extends Euclidean geometry, partially by providing a mathematical definition of infinity. It accomplishes this by adding an extra coordinate to each point. In two dimensions, Euclidean points reside on a plane within the projective space where the third coordinate equals 1. Therefore, to convert from projective coordinates (also referred to as homogeneous coordinates) to Euclidean coordinates, we divide the first two coordinates 73 of the projective representation by the third coordinate (assuming the third coordinate is non-zero and the points do not reside at infinity). This projects points from the general projective space onto the Euclidean plane. Consequently, the notation for the 2D Euclidean points in the project space is as (x, y, 1) (Hartley and Zisserman 2004).

Given these explanations, the next step in converting image coordinates to real-world coordinates involves expressing them in projective (homogeneous) format. This is essential because the homography matrix is computed under the assumption that points are represented in this way. Once the points are in projective coordinates, the following relationship holds between world coordinates and image coordinates[3]:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = H \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{3}$$

Here, u and v represent image coordinates, and X, Y, and Z denote real-world coordinates. The matrix H is also the homography matrix and is computed as follows:

$$H = A[r_1 r_2 r_3 \mid t] \tag{4}$$

Where A represents the camera's intrinsic matrix, including its focal length and optical center, and the second matrix is the extrinsic matrix with columns representing rotations with respect to the X, Y, and Z axes, as well as translation relative to the world-coordinate origin. The camera's intrinsic matrix also has the following representation.

$$A = \begin{pmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5}$$

[3] This is a projective geometry representation. In projective geometry coordinates are defined up to a scale a 1 is arbitrarily set. This can be replaced by any other value, as long as both sides of the equation are consistent

In matrix A, (u_0, v_0) is the coordinate of the principal point, α and β are the scale factors along the u and v axes, and γ the parameter representing the skew of the image axes (Zhang 2004).

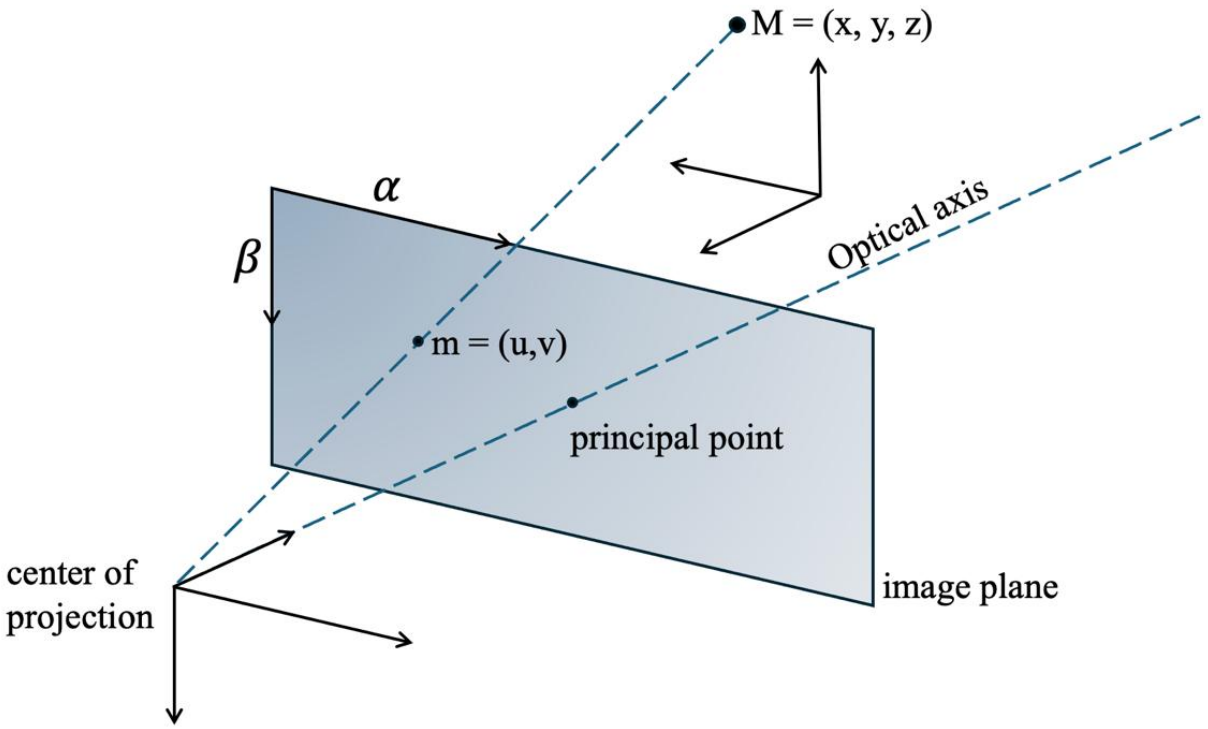


Figure 6: An illustration of the camera's pinhole model used in this study (Zhang 2004; Tomasi 2017)

Since our measurements are in the image coordinate system, based on equation (3), the corresponding real-world measurements can be obtained by multiplying the inverse of the homography matrix with the image measurements.

2.3 Ellipse Fitting of an empty outlet

Since the model outputs a binary image for the object mask, drawing the contour involves calculating the gradients over the binary image and applying a threshold to isolate non-zero values. To fit an ellipse to the contour points, several approaches can be considered, depending on the characteristics of the contour points. In this study, since the contour points closely align with the shape of an ellipse, a least-squares solution like the LIN algorithm (*in* Fitzgibbon and Fisher 1995), which minimizes the algebraic distance between the points and the fitted ellipse, or

the direct least-squares ellipse fitting method, should provide a good approximation of the shape (Fitzgibbon and Fisher 1995; Fitzgibbon et al. 1999).

Given that the fitted ellipse to the outlet mask determines the location of the center, the lengths of the axes, and the orientation of the ellipse in the image coordinate system, we can compute the coordinates for a representative number of points on the ellipse based on its parametric representation. The parametric representation of an ellipse centered at (0, 0) is given by:

$$\begin{aligned} x &= a \times \cos\ t \\ y &= b \times \sin\ t \end{aligned} \tag{6}$$

where the variables a and b are respectively the semi-major and the semi-minor axes, and the variable t ranges from zero to 2π. In Euclidean 2D space, rotation is a linear transformation and can be applied through the following matrix multiplication:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{7}$$

Assuming the ellipse orientation equal to θ, the parametric form of the ellipse will become:

$$\begin{aligned} x' &= x\cos\ \theta - y\sin\ \theta = a\cos\ t\cos\ \theta - b\sin\ t\sin\ \theta \\ y' &= x\sin\ \theta + y\cos\ \theta = a\cos\ t\sin\ \theta + b\sin\ t\cos\ \theta \end{aligned} \tag{8}$$

Finally, assuming the center of the ellipse is at (x_0, y_0), the parametric representation of the ellipse becomes:

$$\begin{aligned} x &= x_0 + a\cos\ t\cos\ \theta - b\sin\ t\sin\ \theta \\ y &= y_0 + a\cos\ t\sin\ \theta + b\sin\ t\cos\ \theta \end{aligned} \tag{9}$$

To generate points on the ellipse, we should divide the range of the variable t into the desired number of segments and plug them into equation (9). The generated points can then be mapped to their real-world coordinates using the inverse homography transformation. Having the ellipse points in their real-world coordinates, the height is calculated as the absolute difference between the maximum and minimum y-coordinates of the points.

To establish and compare the models' precision and performance under varying field lighting conditions, as well as to assess their robustness in detecting the outlet's shape, the models were compared in terms of their measurements for the major and minor axes. For this test, images spanning a full day were taken every 15 minutes. Each model was run on each image to obtain the object mask. Using the mask coordinates, a binary image was created. Gradients were then computed and thresholded to obtain the mask's contour. Next, an ellipse was fitted to the contour using a least-squares method, yielding ellipse parameters including its center, major and minor axes, and orientation. Next, the endpoints of the axes were identified in the image. After applying the homography transformation to these endpoints, their real-world distance was computed using the Euclidean norm (second norm) of the difference between corresponding points.

2.4 Models' water stage measurement comparison in the field

2.4.1 Cameras used and field sites

To train and evaluate the models' performance images from three different sites, a lab prototype, and Google Images licensed under Creative Commons (CC) were used. Two brands of game cameras were deployed in the field: the RECONYX HyperFire 2 Professional Covert IR Camera and Blazevideo A252 Trail Camera. The RECONYX camera has a resolution of 2048×1440 and uses an IR sensor and IR illumination to provide high-quality night images. The Blazevideo camera has a resolution of 3840×2160 and uses a color night vision sensor. Due to field constraints, the distances and angles between the camera and outlet varied for each camera. For the images used, the distance between the camera and the center of the outlet was approximately 6 meters, with a downward vertical angle of approximately 20 degrees from the horizontal.

2.4.2 Performance assessment of the system: measuring outlet diameter in the field

We used these images in our models to estimate the dimensions of a culvert outlet using images of an empty outlet, which appears as an ellipse on images. We estimated the dimensions of its major and minor axes, from a series of images captured over the period of a day. The axes' lengths gave two largely independent measurements of the same distances, i.e., the culvert

diameter. We then compared these estimates to actual measurements taken in the field to assess the model's accuracy and performance. The time series nature of the image data allows us to assess the model's performance in two key areas: its adaptability to varying lighting conditions across the day and its robustness in maintaining consistent measurements over time. Practically, for the culvert diameter assessment and water level (below) we report the performance for the 'Softball site' illustrated in Figure 1a.

2.4.3 Performance assessment of the system: measuring water depth

To assess the performance of the model, an initial attempt was made to use measurements from a flowmeter (Sontek IQ) mounted inside the culvert. However, this method did not yield robust results likely due to several reasons. One, stormwater in our culvert appeared to have a liquid and a gaseous phase because of the turbulence. The Ultrasonic Doppler method used by the Sontek-IQ requires aqueous phase only. Two, because of the turbulence, the stage and velocity could vary by several cm within seconds. The instrument takes measurements every second and averages them over a minimum of 30 seconds. Measurements taken every video frame and averages over 30 seconds are just not directly comparable. Three, the flowmeter's mechanism required a certain water level to start recording values, a condition not met in many cases considered in this study. In summary, due to highly variable and turbulent conditions, flowmeter readings proved inconsistent and were deemed unsuitable as reference measurements. Using a staff gauge, typically employed in stream water level monitoring, was also impractical due to the area's dimensional constraints and the turbulent condition of the water in the culvert. Therefore, the only viable option was to visually verify the water level in each image frame. To facilitate this, and assuming the coplanarity of the reference object and the outlet, a series of horizontal perspective lines were drawn as a guide to help the observer make an educated and potentially accurate judgment. Two people read the same images and the reference values were taken as the average.

To draw the perspective lines around the object mask, we first need to map the mask points to their real-world coordinates and extract the range of values in each direction. Allowing for a leeway around the mask edges, the horizontal perspective lines span between the maximum and minimum x-coordinates in the real-world system. By converting both ends of the lines from the real-world coordinate system to the image coordinate system using the inverse of the

homography transformation, the lines are obtained and could then be drawn on the image. Each yellow line was 2 cm appart from the next, and every fifth line was marked with a red color for better differentiation. Our assessment was that our readings were done within a ±1 cm. Additionally, to aid in the visual reading of the water level, the levels corresponding to the midpoint and the top point of the mask were highlighted with bluelines. A view of the outcome of these processes is shown in Figure 7.

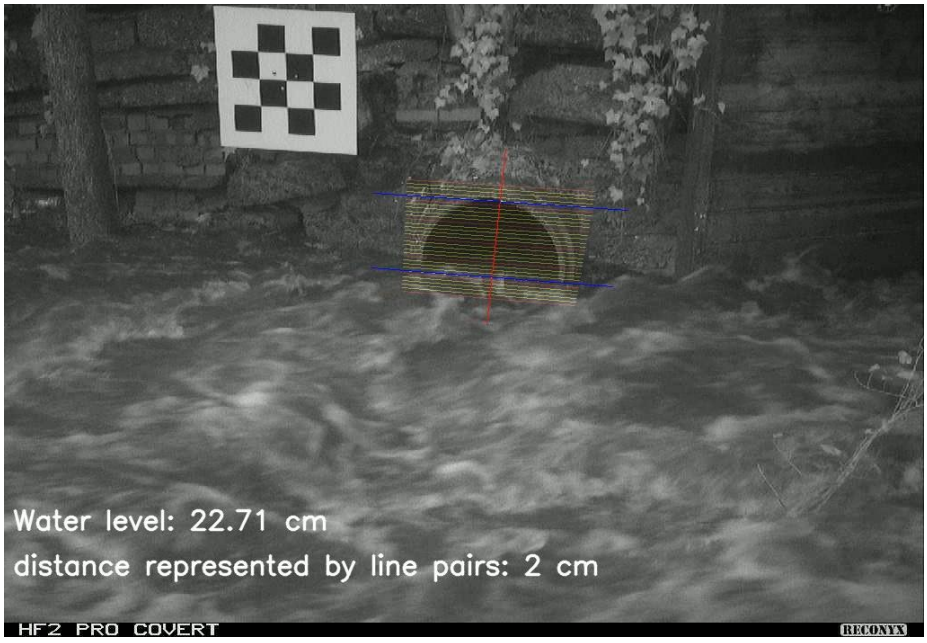


Figure 7: Horizontal perspective lines over the unoccupied area facilitating visual observation

In the processing pipeline, each frame is treated independently to avoid the resonance effect of errors across different frames. Accordingly, the homography transformation, as well as the mapping between the image coordinate system and the real-world coordinate system, is performed for each frame for both the object mask and the culvert's fitted ellipse.

For hydrologists, visual readings might appear at first to be less objective than more traditional sensing that involve pressure transducers or bubblers. However, in these extremely turbulent conditions, none of the traditional methods can be applied, as no stilling well can be installed in these conditions, nor any stream gauge. Although the ±1 cm uncertainty might sound large, it probably is the best that can be achieved in these conditions.

2.5 System's metrology in the lab

The models' performance assessment from the field do not fully reveal all the models' limitations. In practice, field conditions offer limited flexibility in camera placement due to obstructions such as streams or vegetation. Additionally, man-made structures can further restrict suitable mounting locations. Therefore, it is essential to determine more systematically the models' limitations beforehand to assess their suitability for various field conditions. Furthermore, the previous tests relied on a single perspective and thus cannot be used to study the effects of different camera settings, including lens distortion.

To address these situations, a lab setup was developed. A prototype culvert was built and placed on a desk with chair legs for easier mobility. The camera was mounted on a pole attached to a rail above the setup, allowing for easy adjustment of the distance between the camera and the culvert. Figure 8 shows a view of the lab setup. The camera used for these tests was a Blazevideo A252 Trail Camera described above. Notably, the culvert was not perfectly circular, and its actual diameter varied between 34.0 and 34.85 cm.

Figure 8: Laboratory setup for metrology testing: (a) prototype culvert with adjustable legs for mobility; (b) camera mounted on a pole attached to a rail above the setup. Arrows along the rail indicate the direction of camera movement to achieve different distances. Arrows on the pole indicate the direction of camera movement to obtain different vertical angles; (c) culvert positions marked on the floor, with numbers indicating wheel locations for each position and the arrow representing the direction of movement

During the tests, given the dimensions of both the culvert prototype and the chessboard pattern (used to establish the homography transformation between image and real-world coordinates), as well as the camera's depth of field, distances varied from 4.75 m to 11.25 m in 25 cm increments. Additionally, while maintaining the camera's perspective, it was moved vertically in 5.2 cm intervals, equal to the distance between the pole holes, starting at 141 cm from the floor, to capture different vertical angles. The culvert prototype was also moved between three locations along the direction perpendicular to the camera's rail, with the second

position being 30 cm away from the first, and the third position being another 19 cm further (49 cm total from the first position), to simulate varying horizontal angles. The non-uniform choice of distances for culvert movement was constrained by the lab environment. This entire process was repeated both with the lab lights on and off, using the camera's flash in the latter case, to assess the combined effects of lighting conditions and other factors. As a result, 648 images were recorded for each lighting condition, totaling 1296 images.

3 Results

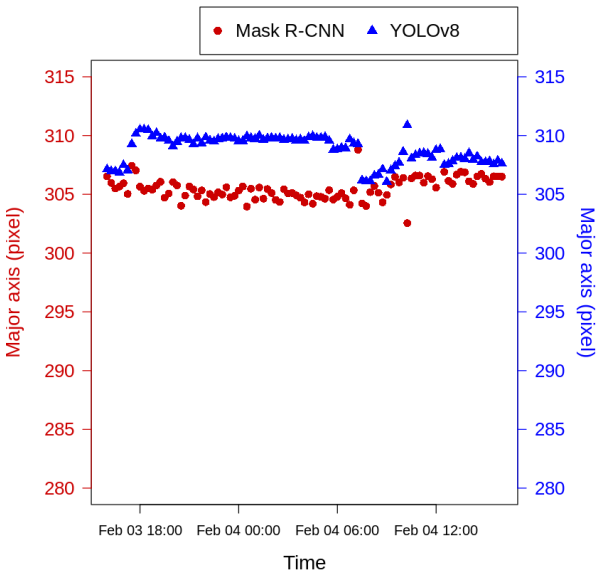
3.1 System's performance in the field: results on culvert diameters

The results obtained from the models' measurements are illustrated in Figure 9, with results expressed in pixel coordinates on the top row (Figure 9a and Figure 9b) and in real-world coordinates in the bottom row (Figure 9c and Figure 9d). Based on the results, one can see that both models were relatively consistent with their measurements throughout the day. However, while both models recorded similar overall performance for the minor axes, as reflected in the values recorded in both image and real-world coordinate systems, the difference between the models is greater for the major axis. The recorded values for each model exhibit greater dispersion, resulting in a wider range of real-world measurement changes, as demonstrated in Figure 9c.

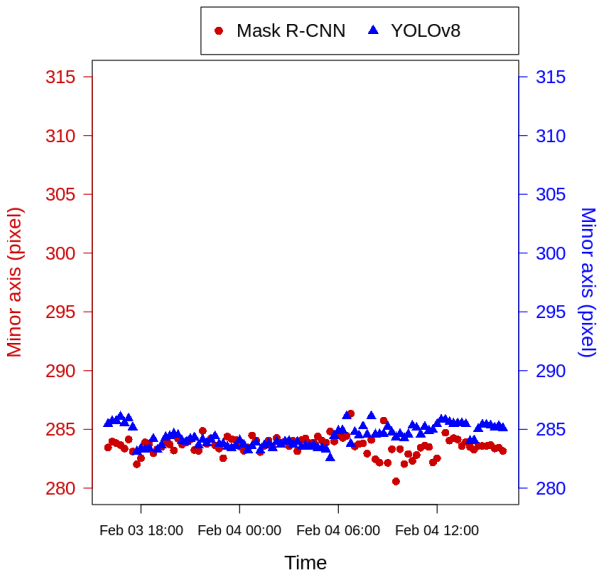
Based on the distribution of points around the actual measurement, no significant difference is observed between the models for the real-world measurements of the minor axis. For the major axis, the YOLOv8 model's data points are slightly more concentrated around the actual value than those of the Mask R-CNN model, suggesting marginally better performance in extracting the outlet's shape. To further establish the overall quality of measurements for each model, their mean relative errors for the axes were also computed. The Mask R-CNN model's mean relative error for the major axis is 0.015, while the YOLOv8 model's is 0.01. This further demonstrates the YOLOv8 model's marginally better performance in estimating the actual value of the major axis. For the minor axis, the mean relative error is 0.009 for the Mask R-CNN model and 0.008 for the YOLOv8 model, demonstrating marginally better performance from the YOLOv8 model compared to the Mask R-CNN model.

Additionally, closer examination of the plots reveals a slight change in measurement patterns between daytime and nighttime for both models. This change is more easily observed in Figure 9a and Figure 9b. To quantify this change in behavior, the mean relative error between daytime and nighttime was computed, as documented in Table 2. Based on the values in Table 2, the Mask R-CNN model's performance in measuring the major axis was slightly worse at night than during the day. However, the change in mean relative error between day and night for the YOLOv8 model is negligible. For the minor axis, both models demonstrated better precision at night than during the day. This could be due to the pattern of camera lighting on the outlet during nighttime image capture, potentially creating better contrast for capturing the edges around the minor axis region of the fitted ellipse compared to the major axis.

A



B



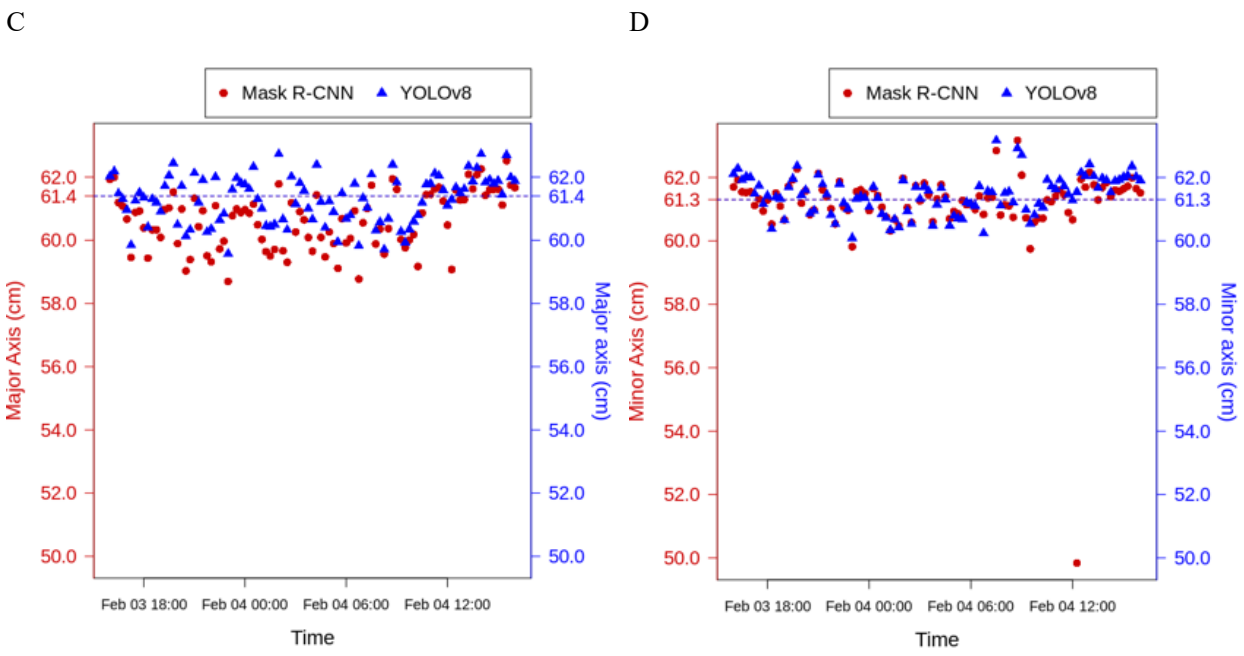


Figure 9: Comparison of major and minor axis lengths in image coordinates (respectively A and B) and real-world units (respectively C and D). Red points represent Mask R-CNN model results, blue points represent YOLOv8 model results.

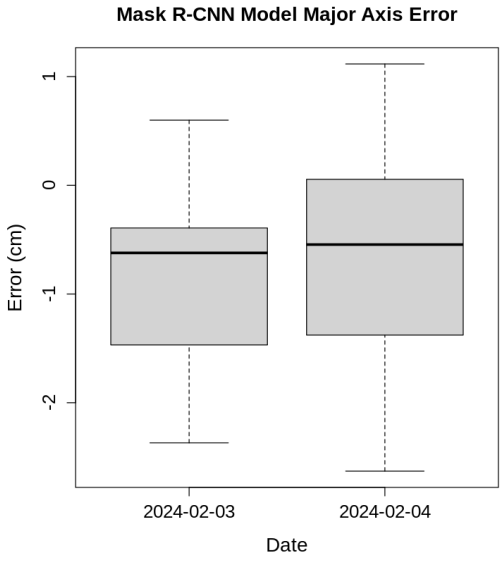
Table 2: Mean relative error for major and minor axes measurements during daytime and nighttime. Values in red correspond to the Mask R-CNN model, and values in blue correspond to the YOLOv8 model.

	Major Axis	Minor Axis
Day	0.011 0.011	0.012 0.01
Night	0.018 0.01	0.006 0.007

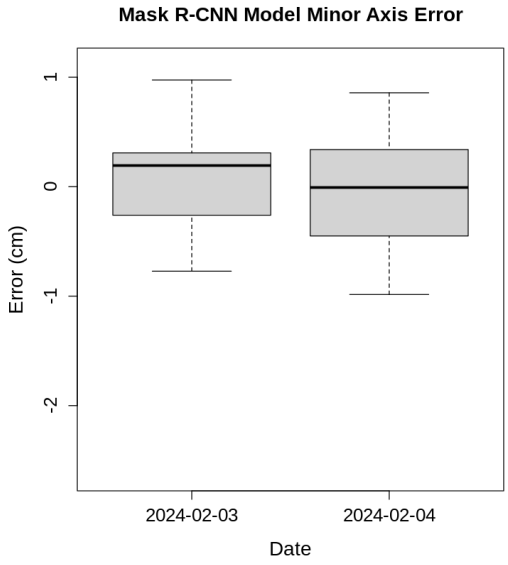
Figure 10 illustrates the error range for the Mask R-CNN and the YOLOv8 models. Based on the plots, one can see that the error range for the major axis measurement is wider than that for the minor axis, which was also demonstrated in Figure 9. However, for both axes, the

majority of the measurements have an absolute error close to or of less than one cm. The models were also evaluated and compared in terms of their computational time. Both models processed a 100-frame video, and the inference time for each frame was recorded. The resulting boxplot (Figure 11) illustrates the distribution of inference times. Figure 11 shows that the Mask R-CNN model computational time is significantly slower than the YOLOv8 model, by more than an order of magnitude (the mean inference time of the Mask R-CNN model is 1.24 seconds per frame, while this value is marginally less than 0.11 seconds per frame for the YOLOv8 model). While YOLOv8's efficient structure contributes to this speed difference, additional memory optimizations also likely play a significant role in its faster operation. Therefore, the observed difference shouldn't be solely attributed to YOLO's structural superiority.

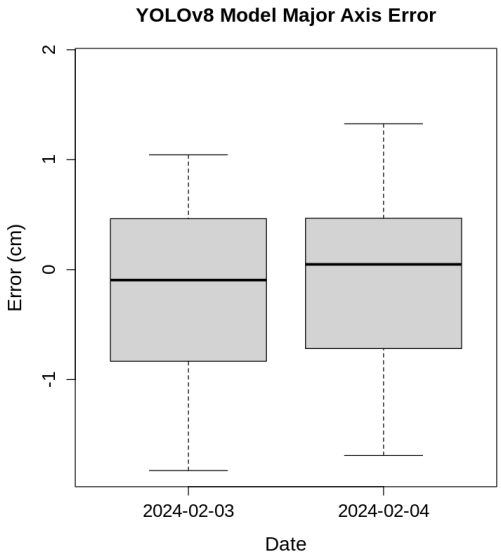
A



B



C



D

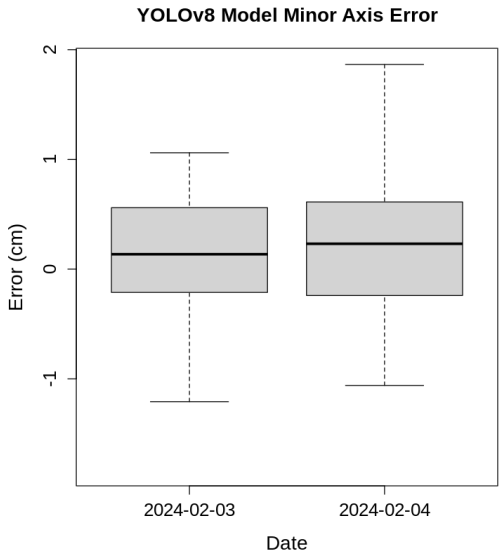


Figure 10: Models error range for the major and minor axes measurements. Mask R-CNN: respectively, A and B. YOLOv8: respectively, C and D

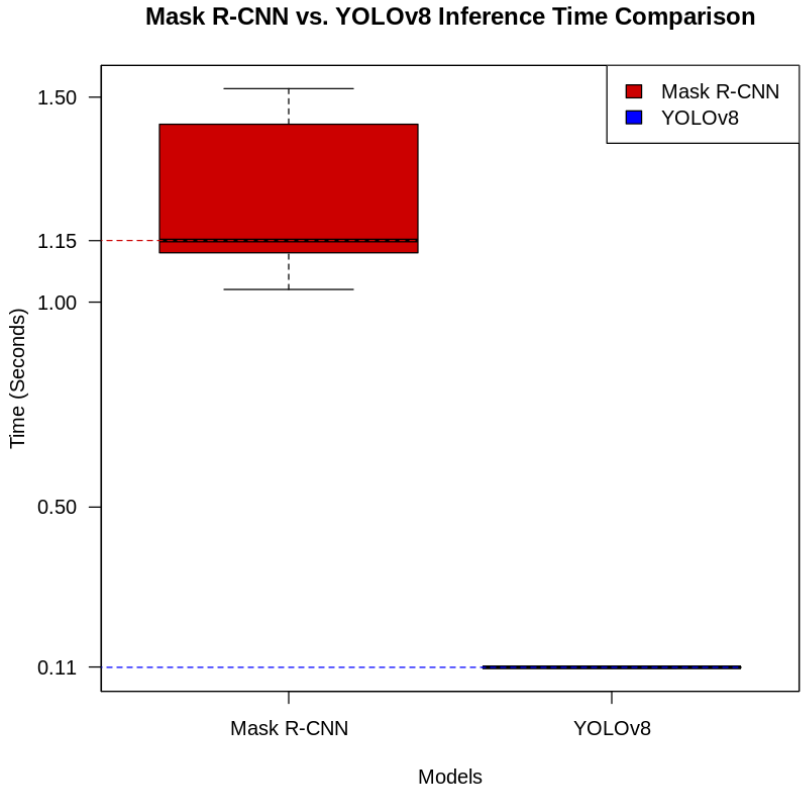


Figure 11: Inference time comparison between the YOLOv8 and the Mask R-CNN

These results were obtained using an NVIDIA T4 GPU with 16 gigabytes of dedicated memory. Nevertheless, many edge devices lack such capabilities, leading to significantly higher run times compared to the reported times above. This is where the YOLOv8 model shines, as its inference time remains within a feasible range even on edge devices with low computational resources, making it suitable for deployment on such devices.

3.2 System's performance in the field: results on water levels

As illustrated in Figure 12, while the models generally track each other, the YOLOv8 model tends to underestimate the water level, when the Mask R-CNN model tends to overestimate it, compared to the visual readings. These readings were performed according to the method described above, i.e., using the added perspective lines on each video frame picture. The estimated visual reading uncertainty was evaluated to be ±1 cm. This pattern is also observable in Figure 13, which compares the models' error range in capturing the water level with respect to the visual reading. Additionally, in comparison to Figure 10, the interquartile range for YOLOv8 is shifted downward, while the error interquartile range for the Mask R-CNN model is shifted upward (Figure 13), indicating that complex conditions affected both models' measurements. This effect on the Mask R-CNN model is also demonstrated in Figure 12. However, given the shorter models error range for the Mask R-CNN model, it can be inferred that the Mask R-CNN model is more reliable and consistent in its measurement than the YOLOv8 model.

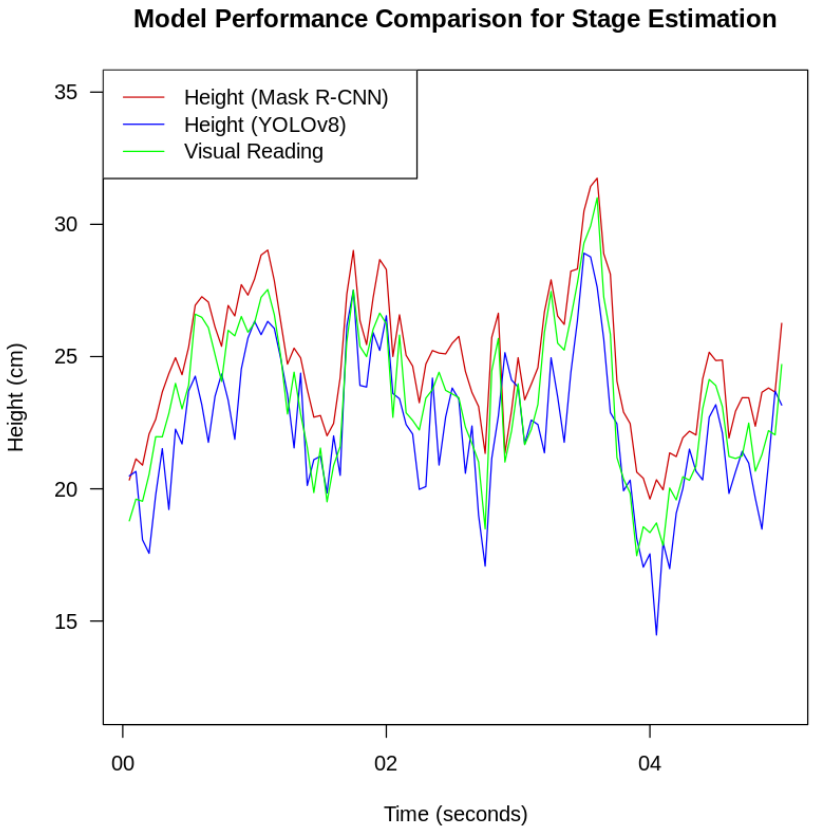


Figure 12: Comparison of the Mask R-CNN and YOLOv8 models with visual measurements for water stage estimation at the stormwater outlet.

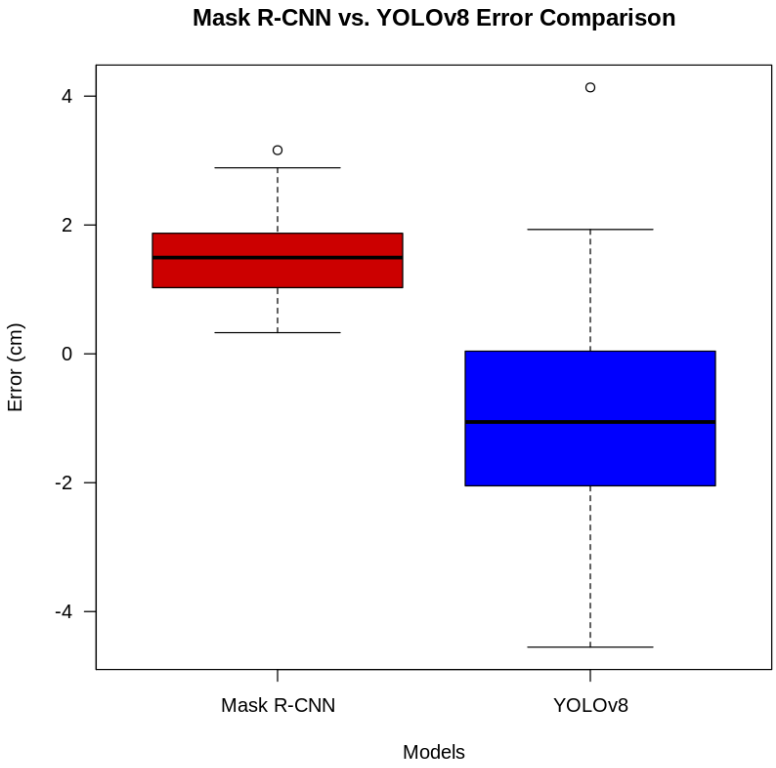


Figure 13: Comparison of the models' error ranges relative to visually measured water stages

As illustrated by Figure 14, a considerable number of stormflow events occur under turbulent conditions due to the slope and design of the conduits, as well as the volume of water typically channeled into the system during precipitation events.

Figure 14: The YOLOv8 model detected the unoccupied area of the culvert (left) compared to the Mask R-CNN model (right)

Consequently, developing a model capable of tracking water levels with fine detail and high precision is challenging. This difficulty is further compounded by the fact that most

stormflow events occur during dim weather conditions, affecting the resolution and visibility of the water level.

A difference and underestimation of a centimeter on average by the parsimonious YOLOv8 model is not ideal, as calculated flows would be systematically underestimated in the conditions illustrated. One of the advantages of using deep learning methods with images and videos in these conditions, is the ability of this method to capture variability that no other more traditional methods would capture, namely the extreme variability of stage over a few seconds at a given point in time. The deep learning methods can also handle the surface of the water when it is not horizontal by any mean, and can vary by 5 cm or more vertically in the culvert. This variability in time and in space (confirmed visually, data not shown) does not reflect actual flow variations, which are expected to be more stable. Instead, it reflects the extreme turbulence resulting from breaks in the connections between consecutive culvert sections (verified in this case study).

The effect of the YOLOv8's mask 'bleeding' (Figure 14), would likely be less severe with less turbulent flow and a more horizontal water level, as observed in other cases monitored for this study (data not shown). The penalty on precision resulting in a gross underestimation of water stage illustrated here, is thus somewhat contingent upon the case study.

3.3 Metrology of the system in the lab

To establish a baseline for comparison between field and lab results, the mean real-world size represented by each pixel within the outlet image is used as a reference metric and reported for different scenarios. The following bar plot illustrates the mean real-world size per pixel for various distances between the camera and the outlet, as determined from the lab experiments.

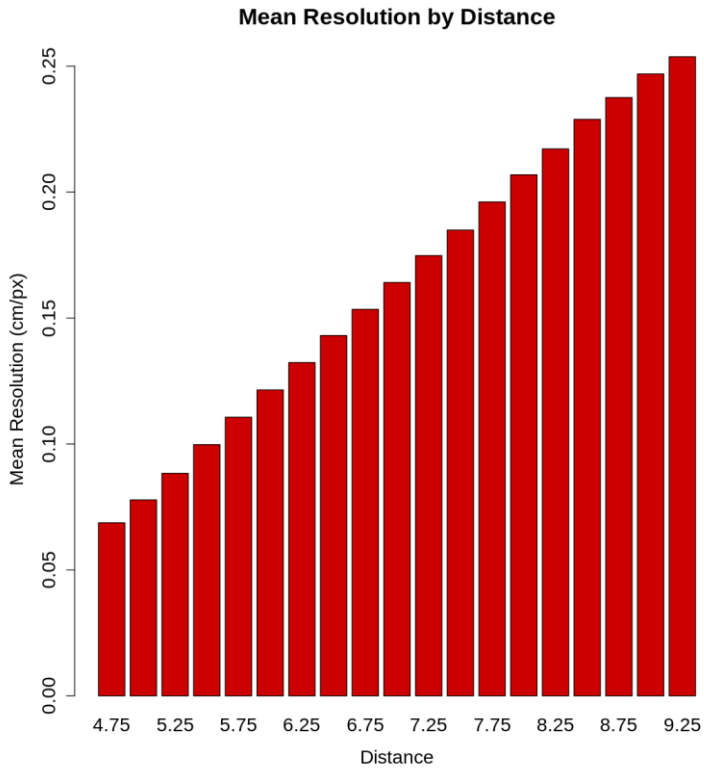


Figure 15: Mean resolution of the culvert's representation at various distances

Four factors were tested in these experiments: the object pixel resolution tested by distance of the camera to the culvert, the lighting conditions, i.e., light vs. dark conditions, the horizontal angle of the camera away from the culvert axis, and the model chosen, i.e., Mask R-CNN vs. Yolov8. The details of all these experiment results can be found in Nooshzadi (2024). A summary is provided here.

3.3.1 Detection of culvert and accurate measurements stop when camera is too far

When the camera was placed beyond a certain distance, both models started to make some erroneous detection of the culvert. Generally, the threshold distance was shorter for the Yolov8 model compared to the Mask R-CNN, and for night images compared to images taken during daylight (Figure 16). Detection of the culvert generally drastically lowered after 7.5 m away from the culvert in the lab, or for resolution lower than 0.23/px.

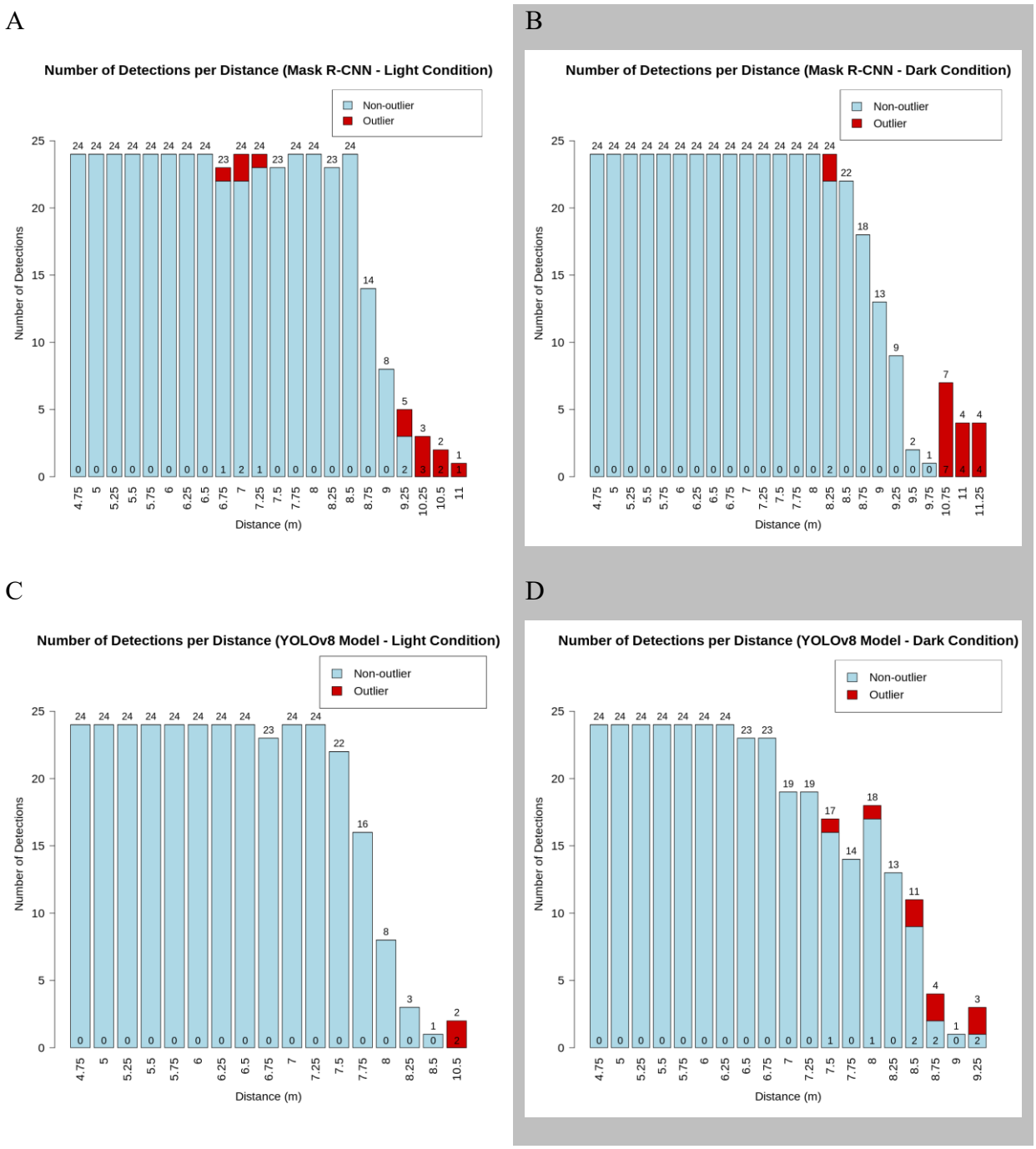


Figure 16: Detection of the culvert in the lab as a function of the distance for the Mask R-CNN (A and B) and the YOLOv8 model (C and D), for images during daylight (A and C) and night images (B and D)

3.3.2 Relative measurement error on diameter increased with image distortion, distance, YOLOv8 model and dark images

The main results are summarized in Figure 17 below. Additional results can be found in Nooshzadi (2024). The errors were calculated from the difference between the reference diameter that was measured with a tape measure (±1.5 mm) and the calculated diameters using our models.

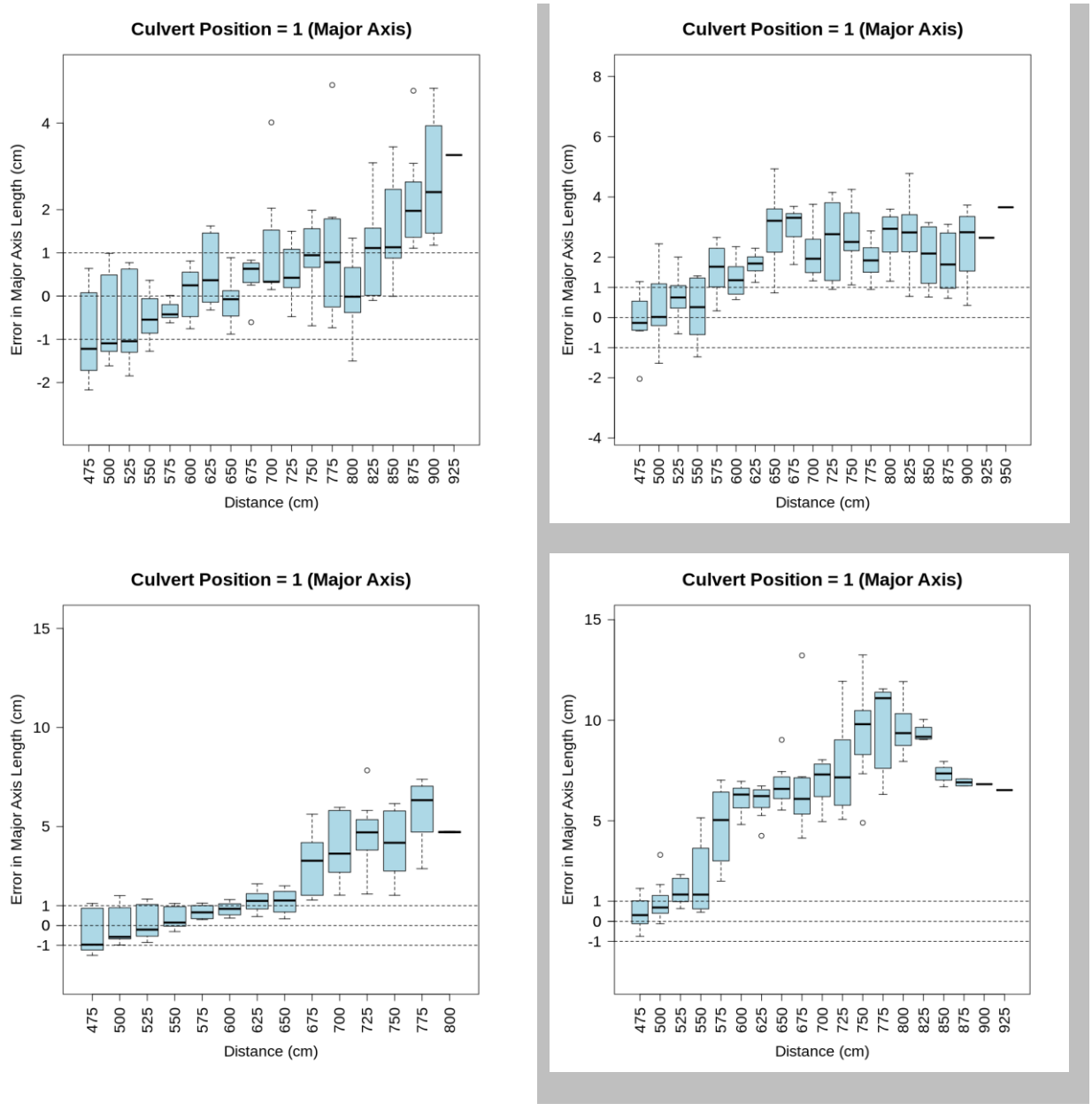
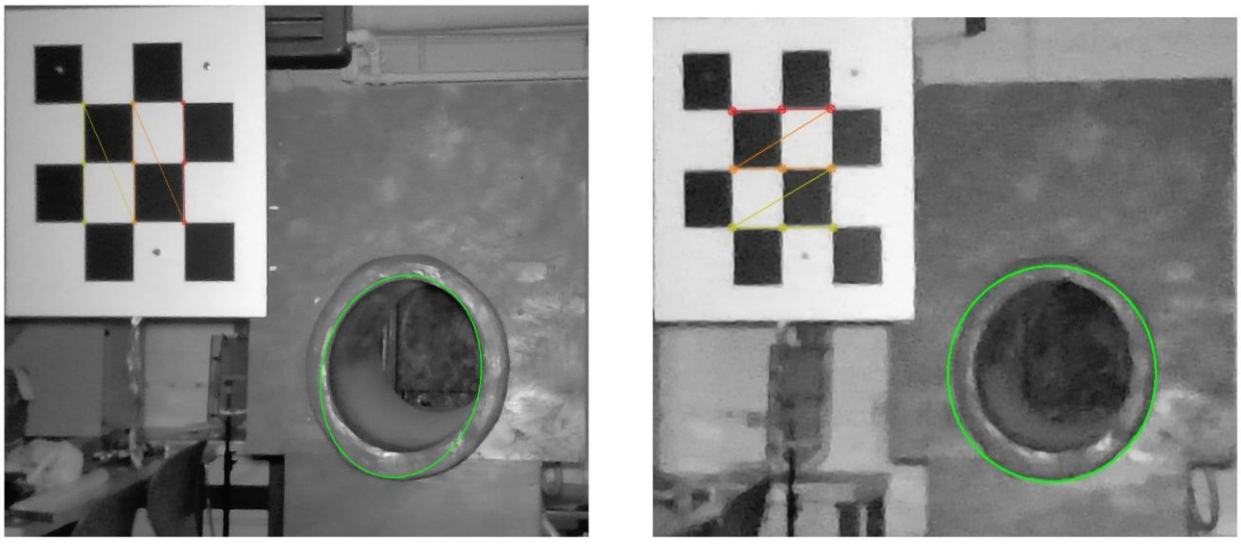


Figure 17: Error on estimations of the culvert diameter via the size of the major axis of the ellipse appearing on images taken in the lab as a function of the distance of the camera to the culvert. Top row: Mask R-CNN model; Bottom row: YOLOv8 model; Left column: daylight images; Right column: dark images

The camera used in the lab tended to distort the images more than that in the field. This tended to give poorer results than those expected and observed in the field, and gave poorer results when the angle between the culvert axis and the camera was large (results not shown in this report).

Lab results show that the Mask R-CNN model tended to work better and more consistently for daylight and night images, with errors within ±1 cm for distances below 7 m, of about 0.25 cm/px. Accuracy largely decreased at night. For the YOLOv8 model, when the culvert was properly detected, the errors were rather acceptable for daylight images (less than 6.5 m), but the errors largely increased suddenly as the ellipse fitting algorithm started largely overestimating ellipse sizes. This was particularly true for night images (Figure 18).



(a) (b)

Figure 18: Representative samples from the YOLOv8 model detections for dark images: (a) Edge detection bleeding (distance: 5.25 meters). (b) Model detected the outer edge instead of the inner edge (distance: 7 meters).

As a result of the analyses conducted in the lab, the Mask R-CNN model was generally found to be more capable of detecting the culvert under challenging conditions than the YOLOv8

model (Figure 17). For light images, while the Mask R-CNN model detections maintained an error range of ±1 cm up to 8 meters (equivalent to an outlet pixel resolution of 0.2 cm/px), the YOLOv8 model's detections remained within this practical error range only up to 6.5 meters, corresponding to an outlet pixel resolution of 0.15 cm/px. In terms of model capacity, the Mask R-CNN model consistently detected the outlet up to 8.5 meters (0.23 cm/px resolution). The YOLOv8 model's detections were consistent up to 7.5 meters (0.19 cm/px resolution), after which the number of detections decreased. For dark images (data not shown), although the number of detections increased for both models, the detections were less accurate than for light images. The practical distance for the Mask R-CNN model, within which the error range for measurements remains within ±1 cm, decreased to 5.5 meters (corresponding to a 0.1 cm/px resolution) under dark conditions.

For the YOLOv8 model, the practical distance decreased to 5 meters (corresponding to a 0.08 cm/px resolution). However, the YOLOv8 model exhibited significantly more errors than the Mask R-CNN model in dark images, indicating its greater sensitivity to low-light conditions. An overview of the models' practical limits for dark and light conditions is provided in Table 3.

Table 3: Practical distance limits and corresponding object pixel resolutions for both models under light and dark conditions, based on lab tests

	Light	Dark
Mask R-CNN	8m 0.2 cm/px	5.5m 0.1 cm/px
YOLOv8	6.5m 0.15 cm/px	5m 0.08 cm/px

Additionally, based on the error range plots for the major and minor axes, as well as the discussion of the camera's pincushion distortion, the impact of distortion on image measurements, especially magnified for near images, was found to contribute noticeably to model measurement errors. Therefore, based on the results, it is recommended that for field operations, the camera be positioned so that the outlet is located in the center of the image. Additionally, it is also advisable to try to keep the reference object (in this study, a chessboard) as close to the outlet as possible, as this will help to locate the reference object near the center of the image as well, thereby minimizing distortion effects on measurements. Furthermore, having the

reference object close to the culvert makes it easier to align it with the culvert face, avoiding errors caused by non-coplanarity.

To further enforce this idea, as discussed in Tomasi (2017), an inexpensive option is to mount onto the lens of a camera, an additional lens designed for a larger sensor. This will effectively crop the field of view to the central portion, where distortion is minimal. Another option to improve the model's performance is to register images with a camera with a longer focal length. As can be easily inferred from the simple pinhole camera model, this would lead to an enlargement of the object's representation in the image, thereby allowing the model to capture the outlet shape and details more accurately. Additionally, as discussed in Andrew Rolands (2020), the camera's angular field of view (AFOV) can be computed as:

$$\alpha = 2\tan^{-1}\left(\frac{d}{2bf}\right) \tag{10}$$

In equation (10), d represents the sensor diagonal, f represents the focal length, and α represents the angular field of view. The parameter b is the bellows factor, typically used to compensate for exposure in cameras with adjustable focus. However, given that the trail cameras used in this study have a fixed focus and do not include a bellows, it is safe to assume that this parameter equals 1. Based on the formula, increasing the focal length leads to a decrease in the field of view, which could necessitate placing the objects closer to the center of the image, thereby minimizing lens distortion effects.

Nevertheless, the results from the lab analyses cannot be fully applicable to field conditions for the following reasons. First, given the project's goal of measuring water levels at the mouth of stormwater culverts in the field, the training process for both models primarily focused on this objective. A majority of the images used for training were chosen from field images captured under various conditions. Only a limited number of images of the culvert prototype, taken outside the lab, were included to compensate for the limited variation in camera angles relative to the culvert in the field dataset. Therefore, it is not surprising that the models perform better in the field than in the lab.

Second, as can be observed in Figure 18 (captured in the lab), the lighting pattern over the culvert and the material used for building the prototype culvert differ from those at the field site. In the field, except for instances where external objects like vegetation or animals obstruct

the camera's view, the camera view is unobstructed, and the lighting covers the outlet relatively uniformly. In contrast, in the lab, the lighting from the camera is not as diffuse, and reflections from objects present in the lab are common. As a result, the uniform lighting conditions typical of the field are difficult to replicate in the lab. Additionally, the material used for building culverts in the field is typically concrete, which is too heavy for a prototype. The lighter weight material used for the culvert prototype was more light-absorbent, making it appear darker than a real culvert in the field.

Nonetheless, this does not imply that the lab tests were not useful. They provided valuable insights into the models' capacity and performance under challenging conditions. However, a deeper analysis like saliency map analysis using a method such as Grad-CAM (Selvaraju et al. 2020) could offer deeper insights into the models' specific behaviors and potential solutions for enhancing their performance.

4 Findings and Conclusion

To overcome the limitations of traditional image processing methods, we developed a hybrid method of using deep learning models and a geometrical method. The deep learning models layered complex architectures that are better suited to handle the complexities of field conditions to detect culvert with and without water. The geometrical method extracted characteristics of ellipses as culverts appear on images and and used homography transformation to express all results in real-world coordinates.

In this study, two instance segmentation models based on the Mask R-CNN and YOLOv8 architectures were developed to segment the visible portion of the outlet's face. This allows for the detection of the entire outlet (defined by its inner edges) under no-flow conditions and only the unoccupied area during flow conditions, enabling the calculation of the water level at the outlet by subtracting these two segmented areas.

Upon evaluating the models at a sample field site, as shown in Table 2 and Figure 13, one can confidently say that they provided, as a first attempt, an innovative, robust, relatively reliable ($\approx \pm$ 1 cm for appropriate image resolution) and promising method to measure water stage in stormwater outlets. In extremely turbulent flow where the water level can change by more than

10 cm in seconds, the models were able to capture the variations in the field at an uncertainty estimated at ≈ ± 2 cm. We are not aware of other techniques able to perform at that level in these conditions.

In the field, the models' performance was largely comparable, with the YOLOv8 model showing marginally better results. When the models were used to measure the culvert's dimensions using images taken every 15 minutes throughout a full day, the majority of measurements for the Mask R-CNN model fell within a ±1.5 cm error range, while the YOLOv8 model maintained a tighter ±1 cm range. During a 5-second video turbulent outflow event, the models recorded values very comparable with human visual measurements aided by perspective lines drawn over the culvert's unoccupied area. The Mask R-CNN model consistently overestimated the water level by up to 2 cm for most measurements, while the YOLOv8 model consistently underestimated the water level by up to 2 cm.

The metrology performed in the lab designed to assess the impact of image resolution, day vs. night images, and image distortion showed that under well-lit conditions, the practical distance at which the majority of measurements had an error range within ±1 cm was found to be 8 meters for the Mask R-CNN model (corresponding to an object pixel resolution of 0.2 cm/px) and 6.5 meters for the YOLOv8 model (corresponding to an object pixel resolution of 0.15 cm/px). However, under dark conditions, the practical distance for the Mask R-CNN and YOLOv8 models was reduced to 5.5 meters (0.1 cm/px) and 5 meters (0.08 cm/px), respectively. The YOLOv8 model also showed greater susceptibility to errors caused by the lighting pattern from the camera over the outlet edges.

5 Recommendations

The models developed in this study should be considered as proof-of-concept and require further development to be field-ready. As a follow-up to this project, analyzing saliency maps for the models using methods like Gradient-Weighted Class Activation Mapping (Grad-CAM) (Selvaraju et al. 2020) could provide deeper insights into their specific behaviors. Based on these insights, the models could be retrained or fine-tuned with additional data to enhance their performance and achieve better results. Another potential avenue for improvement could be exploring more recent, attention-based architectures like Mask2Former (Cheng et al. 2021)

which have demonstrated superior performance in segmentation benchmarks like COCO (Lin et al. 2014) compared to state-of-the-art models like the Mask R-CNN model used in this study. However, it is important to note that attention-based models are typically not optimized for inference speed and are often heavier and slower than convolutional neural networks like those used in this study.

Lastly, the models developed in this study are inherently image-based. Even when applied to videos, they treat them as discrete sets of frames rather than leveraging the spatial and temporal dependencies between frames. A simple solution to this limitation could be to use the Kalman filter method (Kalman 1960) to track distinguishable features across frames, thereby improving consistency between detections. Another, more comprehensive option would be to utilize video instance segmentation (VIS) models, which are specifically designed to track detections and maintain consistent features across frames. Several models have been introduced in this area. For example, the MaskTrack R-CNN model is built upon the Mask R-CNN model used in this study and incorporates a tracking branch to establish associations across frames (Yang et al. 2019). Among attention-based models, as of the time of this writing, the DVIS++ model is the top performer in terms of the average precision criterion (Zhang et al. 2023b).

6 Implementation and Technology Transfer

The models and their applications are the first stage of a two-stage project, which NCDOT is supporting as part of project RP2025-03. This report presents the first stage, i.e., the ability to measure water level in stormwater culverts; and the follow-up project aims at extending these capabilities to measure discharge, including velocity. The overall goal is to provide a user ready system that NCDOT personnel can use on a routine basis to monitor stormflow using cameras in the field.

7 Cited References

Adelson, E., Burt, P., Anderson, C., Ogden, J. M., and Bergen, J. (1984). Pyramid methods in image processing. RCA Engineer, 29(6):33–41.

Almsherqi, Z. A., McLachlan, C. S., Mossop, P., Knoops, K., and Deng, Y. (2005). Direct template matching reveals a host subcellular membrane gyroid cubic structure that is associated with SARS virus. Redox Rep., 10(3):167–171.

Rolands, A. D. (2020). Field Guide to Photographic Science. SPIE. CoreML documentation. https://developer.apple.com/documentation/coreml.

Bechle A. J.,Wu C.H., Liu W.-C., and Nobuaki K. (2012). Development and application of an automated River-Estuary discharge imaging system. J. Hydraul. Eng., 138(4):327–339.

Benoist, J. and Birgand, F. (2002). Les dispositifs de mesure des débits dans les bassins versants agricoles. Ingénieries eau-agriculture-territoires, (32):p. 51 – p. 63.

Bently, J. P. (1995). Principles of measurement systems. Singapore: Longman Singapore Publishers (Pte) Ltd.

Bertalmio, M., Sapiro, G., Caselles, V., and Ballester, C. (2000). Image inpainting. In Proceedings of the 27th annual conference on Computer graphics and interactive techniques, SIGGRAPH '00, pages 417–424, USA. ACM Press/Addison-Wesley Publishing Co.

Birgand, F., Chapman, K., Hazra, A., Gilmore, T., Etheridge, R., and Staicu, A.-M. (2022). Field performance of the GaugeCam image-based water level measurement system. PLOS Water, 1(7): e0000032.

Birgand, F., Lellouche, G., and Appelboom, T.W. (2013). Measuring flow in non-ideal conditions for short-term projects: Uncertainties associated with the use of stage-discharge rating curves. J. Hydrol., 503:186–195.

Bradley, A. A., Kruger, A., Meselhe, E. A., and Muste, M. V. I. (2002). Flow measurement in streams using video imagery. Water Resour. Res., 38(12):51–1–51–8.

Bradski, G. (2000). The OpenCV Library. Dr. Dobb's j. softw. tools prof. program.

Burt, A., Lehmkuhl, M.,Burt, C. M., and Styles, S.W. (1998).Water level sensor and datalogger testing and demonstration. Report No. 99-002.

Canny, J. (1986). A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell., 8(6):679–698.

Chahrour, N., Castaings, W., and Barthélemy, E. (2021). Image-based river discharge estimation by merging heterogeneous data with information entropy theory. Flow Meas. Instrum., 81:102039.

Chakravarthy, S., Sharma, R., and Kasturi, R. (2002). Noncontact level sensing technique using computer vision. IEEE Trans. Instrum. Meas., 51(2):353–361.

Chapman, K.W., Gilmore, T. E., Chapman, C. D., Birgand, F., Mittlestet, A. R., Harner, M. J., Mehrubeoglu, M., and Stranzl, Jr, J. E. (2022). Technical note: Open source software for water level measurement in images with a calibration target. Water Resour. Res., 58(8).

Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., and Girdhar, R. (2021). Masked-attention mask transformer for universal image segmentation. arXiv [cs.CV].

Chetpattananondh, K., Tapoanoi, T., Phukpattaranont, P., and Jindapetch, N. (2014). A self-calibration water level measurement using an interdigital capacitive sensor. Sensors and Actuators A: Physical, 209:175–182.

Cox, V. (2017). Translating Statistics to Make Decisions: A Guide for the Non-Statistician. A press.

Creutin, J. D., Muste, M., Bradley, A. A., Kim, S. C., and Kruger, A. (2003). River gauging using PIV techniques: a proof of concept experiment on the iowa river. J. Hydrol., 277(3):182–194.

Deng, J., Dong,W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE.

Doeblin, E. (1989). Measurement systems, application and design, ed. Mc GrawHill.

Eltner, A., Bressan, P. O., Akiyama, T., Gon.alves,W. N., and Marcato Junior, J. (2021). Using deep learning for automatic water stage measurements. Water Resour. Res., 57(3).

Engelen, L., Cre.lle, S., Schindfessel, L., and DeMulder, T. (2018). Spatio-temporal image based parametric water surface reconstruction: a novel methodology based on refraction. Meas. Sci. Technol., 29(3):035302.

Etheridge, J. R., Birgand, F., and Burchell, M. R. (2015). Quantifying nutrient and suspended solids fluxes in a constructed tidal marsh following rainfall: The value of capturing the rapid changes in flow and concentrations. Ecol. Eng., 78:41–52.

Fischler, M. A. and Bolles, R. C. (1987). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In Readings in Computer Vision, volume 24, pages 726–740. Elsevier.

Fitzgibbon, A., Pilu, M., and Fisher, R. B. (1999). Direct least square fitting of ellipses. IEEE Trans. Pattern Anal.Mach. Intell., 21(5):476–480.

Fitzgibbon, A.W. and Fisher, R. B. (1995). A buyer's guide to conic fitting. In Procedings of the British Machine Vision Conference 1995, number 51. British Machine Vision Association.

Fujita, I., Muste, M., and Kruger, A. (1998). Large-scale particle image velocimetry for flow analysis in hydraulic engineering applications. J. Hydraul. Res., 36(3):397–414.

Fujita, I., Notoya, Y., Tani, K., and Tateguchi, S. (2019). Efficient and accurate estimation of water surface velocity in STIV. Environ. Fluid Mech., 19(5):1363–1378.

Fujita, I.,Watanabe, H., and Tsubaki, R. (2007). Development of a non-intrusive and efficient flow monitoring technique: The space-time image velocimetry (STIV). International Journal of River Basin Management, 5(2):105–114.

Garg, S. and Garg, S. (2019). Automated cloud infrastructure, continuous integration and continuous delivery using docker with robust container security. In 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), pages 467–470. IEEE.

Garg, S., Pundir, P., Rathee, G., Gupta, P. K., Garg, S., and Ahlawat, S. (2021). On continuous integration / continuous delivery for automated deployment of machine learning models using MLOps. In 2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), pages 25–28. IEEE.

Gilmore, T. E., Birgand, F., and Chapman, K.W. (2013). Source and magnitude of error in an inexpensive image-based water level measurement system. J. Hydrol., 496:178–186.

Girshick, R. (2015). Fast r-cnn. In 2015 IEEE International Conference on Computer Vision (ICCV), pages 1440–1448.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In 2014 IEEE Conference on Computer Vision and Pattern Recognition, pages 580–587.

Godley, A. (2002). Flowmeasurement in partially filled closed conduits. FlowMeas. Instrum., 13(5-6):197–201.

Gupta, A., Chang, T.,Walker, J., and Letcher, B. (2022). Towards continuous streamflow monitoring with Time-Lapse cameras and deep learning. In ACM SIGCAS/SIGCHI Conference on Computing and Sustainable Societies (COMPASS), COMPASS '22, pages 353–363, New York, NY, USA. Association for Computing Machinery.

Hansen, I.,Warriar, R., Satzger, C., Sattler, M., Luethi,B., Pe.a-Haro, S., and Duester, R. (2017). An innovative image processing method for flow measurement in open channels and rivers. In Global Conference & Exhibition-2017 "Innovative Solutions in Flow Measurement and Control-Oil, Water and Gas, pages 28–30.

Härdle, W. K. and Simar, L. (2019). Applied Multivariate Statistical Analysis. Springer International Publishing.

Hartley, R. and Zisserman, A. (2004). Multiple View Geometry in Computer Vision. Cambridge University Press.

Hashemi, N. S., Aghdam, R. B., Ghiasi, A. S. B., and Fatemi, P. (2016). Template matching advances and applications in image analysis.

Hauet, A., Creutin, J.-D., and Belleudy, P. (2008). Sensitivity study of large-scale particle image velocimetry measurement of river discharge using numerical simulation. J. Hydrol., 349(1):178–190.

Hauet A., Kruger A., Krajewski- Witold F., Bradley A., Muste M., Creutin J.-D., and Wilson M. (2008). Experimental system for Real-Time discharge estimation using an Image-Based method. J. Hydrol. Eng., 13(2):105–110.

Haurum, J. B., Bahnsen, C. H., Pedersen, M., and Moeslund, T. B. (2020). Water level estimation in sewer pipes using deep convolutional neural networks. Water, 12(12):3412.

He, K., Gkioxari, G., Doll.r, P., and Girshick, R. (2017). Mask R-CNN.

Hies, T., Parasuraman, S. B.,Wang, Y., Duester, R., Eikaas, H., and Tan, K. (2012). Enhanced water-level detection by image processing.

Hofhauser, A., Steger, C., and Navab, N. (2008). Edge-based template matching and tracking for perspectively distorted planar objects. In Bebis, G., Boyle, R., Parvin, B., Koracin, D., Remagnino, P., Porikli, F., Peters, J., Klosowski, J., Arns, L., Chun, Y. K., Rhyne, T.-M., andMonroe, L., editors, Advances in Visual Computing, pages 35–44, Berlin, Heidelberg. Springer Berlin Heidelberg.

Hofhauser, A., Steger, C., and Navab, N. (2009). Edge-based template matching with a harmonic deformation model. In Ranchordas, A., Ara.jo, H. J., Pereira, J. M., and Braz, J., editors, Computer Vision and Computer Graphics. Theory and Applications, pages 176–187, Berlin, Heidelberg. Springer Berlin Heidelberg.

Holland, K. T., Puleo, J. A., and Kooney, T. N. (2001). Quantification of swash flows using video-based particle image velocimetry. Coast. Eng., 44(2):65–77.

Horn, B. K. P. and Schunck, B. G. (1981). Determining optical flow. Artif. Intell., 17:185–203.

Hough, P. V. (1962). Method and means for recognizing complex patterns. US Patent 3,069,654.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D.,Wang,W.,Weyand, T., Andreetto, M., and Adam, H. (2017). Mobile Nets: Efficient convolutional neural networks for mobile vision applications.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., and Murphy, K. (2016). Speed/accuracy trade-offs for modern convolutional object detectors.

Huang, Y.-W., Chen, C.-Y., Tsai, C.-H., Shen, C.-F., and Chen, L.-G. (2006). Survey on block matching motion estimation algorithms and architectures with new results. J. VLSI Signal Process. Syst. Signal Image Video Technol., 42(3):297–320.

Huiskonen, J. T., Parsy, M.-L., Li, S., Bitto, D., Renner, M., and Bowden, T. A. (2014). Averaging of viral envelope glycoprotein spikes from electron cryotomography reconstructions using jsubtomo. J. Vis. Exp., (92):e51714.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift.

ISO (2017). Hydrometry—measurement of discharge by the ultrasonic transit time (time of flight) method. Technical Report 6416, ISO.

ISO (2021). Hydrometry—acoustic doppler profiler—method and application for measurement of flow in open channels from a moving boat. Technical Report 24578, ISO.

Iwahashi, M. and Udomsiri, S. (2007). Water level detection from video with fir filtering. In 2007 16th International Conference on Computer Communications and Networks, pages 826–831.

Iwahashi, M., Udomsiri, S., Imai, Y., and Muramatsu, S. (2007). Water level detection for functionally layered video coding. In 2007 IEEE International Conference on Image Processing, volume 2, pages II – 321–II – 324.

Jähne, B. (1993). Spatio-temporal image processing: theory and scientific applications. Springer.

Jeanbourquin, D., Sage, D., Nguyen, L., Schaeli, B., Kayal, S., Barry, D. A., and Rossi, L. (2011). Flow measurements in sewers based on image analysis: automatic flow velocity algorithm. Water Sci. Technol., 64(5):1108–1114.

Ji, H.W., Yoo, S. S., Lee, B.-J., Koo, D. D., and Kang, J.-H. (2020). Measurement of wastewater discharge in sewer pipes using image analysis. Water, 12(6):1771.

Jocher, G., Chaurasia, A., and Qiu, J. (2023). Ultralytics YOLOv8.

Jodeau, M., Hauet, A., Paquier, A., Le Coz, J., and Dramais, G. (2008). Application and evaluation of LS-PIV technique for the monitoring of river surface velocities in high flow conditions. Flow Meas. Instrum., 19(2):117–127.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. J. Basic Eng., 82(1):35–45.

Kantoush, S. A., Schleiss, A. J., Sumi, T., and Murasaki, M. (2011). LSPIV implementation for environmental flow in various laboratory and field cases. Journal of Hydro-environment Research, 5(4):263–276.

Kaplan, N. H., Sohrt, E., Blume, T., and Weiler, M. (2019). Monitoring ephemeral, intermittent and perennial streamflow: a dataset from 182 sites in the attert catchment, luxembourg. Earth System Science Data, 11(3):1363–1374.

Keane, R. D. and Adrian, R. J. (1992). Theory of cross-correlation analysis of PIV images. Appl. Sci. Res., 49(3):191–215.

Kim, J., Han, Y., and Hahn, H. (2011). Embedded implementation of image-based water level measurement system. IET Comput. Vision, 5(2):125–133.

Kim, J. and Kim, J. (2020). Estimation of water surface flow velocity in coastal video imagery by visual tracking with deep learning. J. Coast. Res.

Kim, J. R. and Jeon, J.W. (2020). Sobel edge-based image template matching in fpga. In 2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia), pages 1–2.

Kim, Y., Muste, M., Hauet, A., Krajewski, W. F., Kruger, A., and Bradley, A. (2008). Stream discharge using mobile large-scale particle image velocimetry: A proof of concept. Water Resources Research, 44(9).

Kriech, A. J. and Osborn, L. V. (2022). Review of the impact of stormwater and leaching from pavements on the environment. J. Environ. Manage., 319:115687.

Le Coz, J., Camenen, B., Peyrard, X., and Dramais, G. (2012). Uncertainty in open-channel discharges measured with the velocity-area method. Flow Meas. Instrum., 26:18–29.

Le Coz, J., Hauet, A., Pierrefeu, G., Dramais, G., and Camenen, B. (2010). Performance of image-based velocimetry (LSPIV) applied to flash-flood discharge measurements in mediterranean rivers. J. Hydrol., 394(1):42–52.

Le Coz, J., Renard, B., Vansuyt, V., Jodeau, M., and Hauet, A. (2021). Estimating the uncertainty of video-based flow velocity and discharge measurements due to the conversion of field to image coordinates. Hydrol. Process., 35(5).

Li, J., Lu, Y., Shen, N., Pu, J., and Ma, Z. (2022). Adaptive image enhancement and dynamic template- matching-based edge extraction method for diamond roller on-machine profile measurement. Int. J. Adv. Manuf. Technol., 120(9):5997–6010.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Lawrence Zitnick, C., and Dollár, P. (2014). Microsoft COCO: Common objects in context.

Lin, Y.-T., Lin, Y.-C., and Han, J.-Y. (2018). Automatic water-level detection using single camera images with varied poses. Measurement, 127:167–174.

Liptak, B. G. and Lipták, B. G. (2003). Process measurement and analysis, volume 20. CRC press Florida.

Lloyd, P. M., Stansby, P. K., and Ball, D. J. (1995). Unsteady surface-velocity field measurement using particle tracking velocimetry. J. Hydraul. Res., 33(4):519–534.

Loshchilov, I. and Hutter, F. (2016). SGDR: Stochastic gradient descent with warmer starts.

Müller, A., Österlund, H., Marsalek, J., and Viklander, M. (2020). The pollution conveyed by urban runoff: A review of sources. Sci. Total Environ., 709:136125.

Muste, M., Fujita, I., and Hauet, A. (2008). Large-scale particle image velocimetry for measurements in riverine environments. Water Resour. Res., 44(4).

Nguyen, L. S., Schaeli, B., Sage, D., Kayal, S., Jeanbourquin, D., Barry, D. A., and Rossi, L. (2009). Vision-based system for the control and measurement of wastewater flow rate in sewer systems. Water Sci. Technol., 60(9):2281–2289.

Nikolov, G. and Nikolova, B. (2008). Virtual techniques for liquid level monitoring using differential pressure sensors. Recent, 9(2):49.

Noto, S., Tauro, F., Petroselli, A., Apollonio, C., Botter, G., and Grimaldi, S. (2022). Lowcost stage-camera system for continuous water-level monitoring in ephemeral streams. Hydrol. Sci. J., 67(9):1439–1448.

Otsu, N. (1979). A threshold selection method from Gray-Level histograms. IEEE Trans. Syst. Man Cybern., 9(1):62–66.

Pan, J., Yin, Y., Xiong, J., Luo, W., Gui, G., and Sari, H. (2018). Deep Learning-Based unmanned surveillance systems for observing water levels. IEEE Access, 6:73561–73571.

Peña-Haro, S., Carrel, M., Lüthi, B., Hansen, I., and Lukes, R. (2021). Robust Image-Based streamflow measurements for Real-Time continuous monitoring. Frontiers inWater, 3.

Pizer, S. M., Johnston, R. E., Ericksen, J. P., Yankaskas, B. C., andMuller, K. E. (2002). Contrastlimited adaptive histogram equalization: speed and effectiveness. In [1990] Proceedings of the First Conference on Visualization in Biomedical Computing. IEEE Comput. Soc. Press.

Prudencio, L. and Null, S. E. (2018). Stormwater management and ecosystem services: a review. Environ. Res. Lett., 13(3):033002.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 779–788.

Redmon, J. and Farhadi, A. (2017). Yolo9000: Better, faster, stronger. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 6517–6525, Los Alamitos, CA, USA. IEEE Computer Society.

Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster R-CNN: Towards Real-Time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell., 39(6):1137–1149.

Savitzky, A. and Golay, M. J. E. (1964). Smoothing and differentiation of data by simplified least squares procedures. Anal. Chem., 36(8):1627–1639.

Schoener G. (2018). Time-Lapse photography: Low-Cost, Low-Tech alternative for monitoring flow depth. J. Hydrol. Eng., 23(2):06017007.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2020). Grad-CAM: Visual explanations from deep networks via gradient-based localization. Int. J. Comput. Vis., 128(2):336–359.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for Large-Scale image recognition.

Sintorn, I.-M., Homman-Loudiyi, M., Söderberg-Naucl.r, C., and Borgefors, G. (2004). A refined circular template matching method for classification of human cytomegalovirus capsids in TEM images. Comput. Methods Programs Biomed., 76(2):95–102.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res., 15:1929– 1958.

Swain, M. J. and Ballard, D. H. (1990). Indexing via color histograms. In [1990] Proceedings Third International Conference on Computer Vision, pages 390–393.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. (2016). Inception-v4, Inception-ResNet and the impact of residual connections on learning.

Takagi, Y., Tsujikawa, A., Takato, M., Saito, T., and Kaida, M. (1998). Development of a noncontact liquid level measuring system using image processing. Water Sci. Technol., 37(12):381–387.

Tomasi, C. (2017). A Simple Camera Model.

Tsihrintzis, V. A. and Hamid, R. (1997). Modeling and management of urban stormwater runoff quality: A review. Water Resour. Manage., 11:137–164.

Tsubaki, R., Fujita, I., and Tsutsumi, S. (2011).Measurement of the flood discharge of a smallsized river using an existing digital video recording system. Journal of Hydro-environment Research, 5(4):313–321.

US EPA (2015). Urbanization - stormwater runoff.

USGS (2022). Hydrologic imagery visualization and information system (HIVIS). https://www.usgs.gov/tools/ hydrologic-imagery-visualization-and-information-system-hivis. Accessed: 2023-5-30.

Vass, G. (2000). The principles of level measurement-rf capacitance, conductance, hydrostatic tank gauging, radar, and ultrasonics are the leading sensor technologies in liquid level

tank measurement and control. Sensors-the Journal of Applied Sensing Technology, 17(10):55–64.

Willert, C. E. and Gharib, M. (1991). Digital particle image velocimetry. Exp. Fluids, 10(4):181–193.

Winston R. J. and Hunt W. F. (2017). Characterizing runoff from roads: Particle size distributions, nutrients, and gross solids. J. Environ. Eng., 143(1):04016074.

Wu, H., Zhao, R., Gan, X., and Ma, X. (2019). Measuring surface velocity of water flow by dense optical flow method. Water, 11(11):2320.

Yang, L., Fan, Y., and Xu, N. (2019). Video instance segmentation.

Young, S. N., Han, M., and Peschel, J. M. (2023). Computer vision approach for tile drain outflow rate estimation. Appl. Eng. Agric., 39(2):153–165.

Yu, H., Chen, C., Du, X., Li, Y., Rashwan, A., Hou, L., Jin, P., Yang, F., Liu, F., Kim, J., and Li, J. (2020). Tensorflow model garden. https://github.com/tensorflow/models.

Yu, J. and Hahn, H. (2010). Remote detection and monitoring of a water level using narrow band channel. J. Inf. Sci. Eng., 26(1):71–82.

Yuen, H. K., Illingworth, J., and Kittler, J. (1989). Detecting partially occluded ellipses using the hough transform. Image Vis. Comput., 7(1):31–37.

Yuliza, E., Salam, R. A., Amri, I., Atmajati, E. D., Hapidin, D. A., Meilano, I., Munir, M. M., Abdullah, M., and Khairurrijal (2016). Characterization of a water level measurement system developed using a commercial submersible pressure transducer. In 2016 International Conference on Instrumentation, Control and Automation (ICA), pages 99–102. IEEE.

Zhang, K., Lucas, B., and Grigorieff, N. (2023a). Exploring the limits of 2D template matching for detecting targets in cellular Cryo-EM images. Microsc.Microanal., 29(29 Suppl 1):931.

Zhang, T., Tian, X., Zhou, Y., Ji, S., Wang, X., Tao, X., Zhang, Y., Wan, P., Wang, Z., and Wu, Y. (2023b). DVIS++: Improved decoupled framework for universal video segmentation.

Zhang, Z. (2004). Camera calibration. In Sing Bing Kang, G. M., editor, Emerging topics in computer version.

Zhang, Z., Wang, Y., Zhang, J., and Mu, X. (2019a). Comparison of multiple feature extractors on faster RCNN for breast tumor detection. In 2019 8th International Symposium on Next Generation Electronics (ISNE), pages 1–4.

Zhang, Z., Zhou, Y., Liu, H., and Gao, H. (2019b). In-situ water level measurement using NIR-imaging video camera. FlowMeas. Instrum., 67:95–106.

Zhang, Z., Zhou, Y., Liu, H., Zhang, L., and Wang, H. (2019c). Visual measurement of water level under complex illumination conditions. Sensors, 19(19).

Zhao, H., Chen, H., Liu, B., Liu, W., Xu, C.-Y., Guo, S., and Wang, J. (2021). An improvement of the Space-Time image velocimetry combined with a new denoising method for estimating river discharge. Flow Meas. Instrum., 77:101864.