

# EYE OF HORUS: A VISION-BASED FRAMEWORK FOR REAL-TIME WATER LEVEL MEASUREMENT

Mohammad H. Erfani<sup>1</sup>, Corinne Smith<sup>2</sup>, Zhenyao Wu<sup>3</sup>, Elyas Asadi Shamsabadi<sup>4</sup>,  
Farboud Khatami<sup>1</sup>, Austin R.J. Downey<sup>1,2</sup>, Jasim Imran<sup>1</sup>, and Erfan Goharian<sup>\*1</sup>

<sup>1</sup>Department of Civil & Environmental Engineering, University of South Carolina Columbia, SC 29208, USA

<sup>2</sup>Department of Mechanical Engineering, University of South Carolina, Columbia, SC 29208, USA

<sup>3</sup>Department of Computer Science & Engineering, University of South Carolina, Columbia, SC 29201, USA

<sup>4</sup>School of Civil Engineering, Faculty of Engineering, The University of Sydney, Sydney, NSW 2006, Australia

## Abstract

Heavy rains and tropical storms often result in floods, which are expected to increase in frequency and intensity. Flood prediction models and inundation mapping tools provide decision-makers and emergency responders with crucial information to better prepare for these events. However, the performance of models relies on the accuracy and timeliness of data received from in-situ gaging stations and remote sensing; each of these data sources has its limitations, especially when it comes to real-time monitoring of floods. This study presents a vision-based framework for measuring water levels and detecting floods using Computer Vision and Deep Learning (DL) techniques. The DL models use time-lapse images captured by surveillance cameras during storm events for the semantic segmentation of water extent in images. Three different DL-based approaches, namely PSPNet, TransUNet, and SegFormer, were applied and evaluated for semantic segmentation. The predicted masks are transformed into water level values by intersecting the extracted water edges, with the 2D representation of a point cloud generated by an Apple iPhone 13 Pro LiDAR sensor. The estimated water levels were compared to reference data collected by an ultrasonic sensor. The results showed that SegFormer outperformed other DL-based approaches by achieving 99.55% and 99.81% for Intersection over Union (IoU) and accuracy, respectively. Moreover, the highest correlations between reference data and the vision-based approach reached above 0.98 for both the coefficient of determination ( $r^2$ ) and Nash-Sutcliffe Efficiency. This study demonstrates the potential of using surveillance cameras and Artificial Intelligence for hydrologic monitoring and their integration with existing surveillance infrastructure.

## 1 Introduction

Flood forecasts and Flood Inundation Mapping (FIM) can play an important role in saving human lives and reducing damages by providing timely information for evacuation planning, emergency management, and relief efforts [Gebrehiwot et al., 2019]. These models and tools are designed to identify and predict inundation areas and the severity of damage caused by storm events. Two primary sources of data for these models are in-situ gaging networks and remote sensing. For example, in-situ stream gages, such as those operated by the United States Geological Survey (USGS) provide useful streamflow information like water height and discharge at monitoring sites [Turnipseed and Sauer, 2010]. However, they cannot provide an adequate spatial resolution of streamflow characteristics [Lo et al., 2015]. The limitation of in-situ stream gages is further exacerbated by the lack of systematic installation along the waterways and accessibility issues [Li et al., 2018; King et al., 2018]. Satellite data and remote sensing can complement in-situ gage data by providing information at a larger spatial scale [Alsdorf et al., 2007]. However, continuous monitoring data for a region of interest remains to be a problem due to the limited revisit intervals of satellites, cloud cover, and systematic departures or biases [Panteras and Cervone, 2018]. Crowdsourcing methods have gained attention as a potential solution but their reliability is questionable [Schnebele et al., 2014; Goodchild, 2007; Howe, 2008]. To address these limitations and enhance real-time monitoring capabilities, surveillance cameras are in-

\*goharian@cec.sc.edu

46 tigated here as a new source of data for hydrologic monitoring and flood data collection. However, this  
47 requires a significant investment in Computer Vision (CV) and Artificial Intelligence (AI) techniques  
48 to develop reliable methods for detecting water in surveillance images and translating that information  
49 into numerical data.

50 Recent advances in CV offer new techniques for processing image data for the quantitative measure-  
51 ments of physical attributes from a site [Forsyth and Ponce, 2002]. However, there is limited knowledge  
52 of how visual information can be used to estimate physical water parameters using CV techniques.  
53 Inspired by the principle of the float method, Tsubaki et al. [2011] used different image processing tech-  
54 niques to analyze images captured by closed-circuit television (CCTV) systems installed for surveillance  
55 purposes to measure the flow rate during flood events. In another example, Kim et al. [2011] proposed  
56 a method for measuring water level by detecting the borderline between a staff gauge and the surface  
57 of water based on image processing of the captured image of the staff gage installed in the middle of  
58 the river. As the use of images for environmental monitoring becomes more popular, several studies  
59 have investigated the source and magnitude of errors common in image-based measurement systems,  
60 such as the effect of image resolution, lighting effects, perspective, lens distortion, water meniscus,  
61 and temperature changes [Elias et al., 2020; Gilmore et al., 2013]. Furthermore, proposed solutions  
62 to resolve difficulties originating from poor visibility have been developed to better identify readings  
63 on staff gages [Zhang et al., 2019]. Recently, Deep Learning (DL) has become prevalent across a wide  
64 range of disciplines, particularly in applied sciences such as CV and engineering.

65 DL-based models have been utilized by the water resources community to determine the extent of  
66 water and waterbodies visible in images captured by surveillance camera systems. These models can  
67 estimate the water level [Pally and Samadi, 2022]. In a similar vein, Moy de Vitry et al. [2019];  
68 Vandaele et al. [2021] employed a DL-based approach to identify floodwater in surveillance footage  
69 and introduced a novel qualitative flood index, SOFI, to determine water level fluctuations. SOFI  
70 was calculated by taking the aspect ratio of the area of the water surface detected within an image  
71 to the total area of the image. However, these types of methods, which make prior assumptions  
72 and estimate water level fluctuation roughly, cannot serve as a vision-based alternative for measuring  
73 streamflow characteristics. More systematic studies adopted photogrammetry to reconstruct a high-  
74 quality 3D model of the environment with a high spatial resolution to have a precise estimation of  
75 real-world coordination while measuring streamflow rate and stage. For example, Eltner et al. [2018,  
76 2021] introduced a method based on Structure from Motion (SfM), and photogrammetric techniques,  
77 to automatically measure the water stage using low-cost camera setups.

78 Advances in photogrammetry techniques enable 3D surface reconstruction with a high temporal and  
79 spatial resolution. These techniques are adopted to build 3D surface models from RGB imagery [West-  
80 oby et al., 2012; Eltner and Schneider, 2015; Eltner et al., 2016]. However, most of the photogrammetric  
81 methods are still expensive as they rely on differential global navigation satellite systems (DGNSS),  
82 ground control points (GCPs), commercial software, and data processing on an external computing  
83 device [Froideval et al., 2019]. A LiDAR scanner, on the other hand, is now easily available since the  
84 introduction of the iPad Pro and iPhone 12 Pro in 2020 by Apple. This device is the first smartphone  
85 equipped with a native LiDAR scanner and offers a potential paradigm shift in digital field data acqui-  
86 sition which puts these devices at the forefront of smartphone-assisted fieldwork [Tavani et al., 2022].  
87 So far, the iPhone LiDAR sensor has been used in different studies such as forest inventories [Gollob  
88 et al., 2021] and coastal cliff site [Luetzenburg et al., 2021]. The availability of LiDAR sensors to build  
89 3D environments, and advancements in DL-based models offer a great potential to produce numerical  
90 information from ground-based imageries.

91 This paper presents a vision-based framework for measuring water levels from time-lapse images. The  
92 proposed framework introduces a novel approach by utilizing the iPhone LiDAR sensor as a laser scan-  
93 ner, which is commonly available on consumer-grade devices, for scanning and constructing a 3D point  
94 cloud of the region of interest. During the data collection phase, time-lapse images and ground truth  
95 water level values were collected using an embedded camera and ultrasonic sensor. The water extent  
96 in the captured images was determined automatically using semantic segmentation DL-based models.  
97 For the first time, the performance of three different state-of-the-art DL-based approaches, including  
98 Convolutional Neural Networks (CNN), hybrid CNN-Transformer, and Transformers-Multilayer Per-  
99 ceptron (MLP), was evaluated and compared. CV techniques were applied for camera calibration, pose

100 estimation of the camera setup in each deployment, and 3D-2D reprojection of the point cloud onto  
101 the image plane. Finally, K-Nearest Neighbors (KNN) was used to find the nearest projected (2D)  
102 point cloud coordinates to the water line on the river banks, for estimating the water level in each  
103 time-lapse image.

## 104 2 Deep Learning Architectures

105 Since this study tends to cover a wide range of DL approaches, this section solely focuses on reviewing  
106 different DL-based architectures. So far, different DL networks were applied and evaluated for semantic  
107 segmentation of the waterbodies within the RGB images captured by cameras [Erfani et al., 2022]. All  
108 existing semantic segmentation approaches—CNN and Transformer-based—share the same objective of  
109 classifying each pixel of a given image but differ in the network design.

110 CNN-based models were designed to imitate the recognition system of primates [Shamsabadi et al.,  
111 2022], while possessing different network designs such as low-resolution representations learning [Long  
112 et al., 2015; Chen et al., 2017], high-resolution representations recovering [Badrinarayanan et al., 2015;  
113 Noh et al., 2015; Lin et al., 2017], contextual aggregation schemes [Yuan and Wang, 2018; Zhao et al.,  
114 2017; Yuan et al., 2020], feature fusion and refinement strategy [Lin et al., 2017; Huang et al., 2019;  
115 Li et al., 2019; Zhu et al., 2019; Fu et al., 2019]. CNN-based models follow local to global features in  
116 different layers of the forward pass, which used to be thought of as a general intuition of the human  
117 recognition system. In this system, objects are recognized through the analysis of texture and shape-  
118 based clues—local and global representations and their relationship in the entire field of view. Recent  
119 research, however, shows significant differences exist between the visual behavioral system of humans  
120 and CNN-based models [Geirhos et al., 2018b; Dodge and Karam, 2017; De Cesarei et al., 2021; Geirhos  
121 et al., 2020, 2018a], and reveal higher sensitivity of the visual systems in humans to global features  
122 rather than local ones [Zheng et al., 2018]. This fact drew attention to models that focus on the global  
123 context in their architectures.

124 Developed by Dosovitskiy et al. [2020], Vision Transformer (ViT) was the first model that showed  
125 promising results on a computer vision task (image classification) without using convolution operation  
126 in its architecture. In fact, ViT adopts “Transformers,” as a self-attention mechanism, to improve  
127 accuracy. “Transformer” was initially introduced for sequence-to-sequence tasks such as text trans-  
128 lation [Vaswani et al., 2017]. However, as applying the self-attention mechanism on all image pixels  
129 is computationally expensive, the Transformer-based models could not compete with the CNN-based  
130 models until the introduction of ViT architecture which applies self-attention calculations on the low-  
131 dimension embedding of small patches originating from splitting the input image, to extract global  
132 contextual information. Successful performance of ViT on image classification inspired several subse-  
133 quent works on Transformer-based models for different computer vision tasks [Liu et al., 2021].

134 In this study, three different DL-based approaches including CNN, hybrid CNN-Transformer, and  
135 Transformers-Multilayer Perceptron (MLP) were trained and tested for semantic segmentation of wa-  
136 ter. For these approaches, the selected models were PSPNet [Zhao et al., 2017], TransUNet [Chen  
137 et al., 2021] and SegFormer [Xie et al., 2021], respectively. The performance of these models is eval-  
138 uated and compared using conventional metrics, including class-wise Intersection over Union (IoU) and  
139 per-pixel accuracy (ACC).

## 140 3 Study Area

141 In order to evaluate the performance of the proposed framework for measuring the water levels in rivers  
142 and channels, a time-lapse camera system has been deployed at Rocky Branch, South Carolina. This  
143 creek is approximately 6.5 km long and collects stormwater from the University of South Carolina  
144 campus and the City of Columbia. Rocky Branch is subjected to rapid changes in water flow and  
145 discharges into the Congaree River [Morsy et al., 2016]. The observation site is located within the  
146 University of South Carolina campus behind 300 Main Street (see Figure 1a).

147 An Apple iPhone 13 Pro LiDAR sensor was used to scan the region of interest. Although there is  
148 no official information about the technology and hardware specifications, Gollob et al. [2021] reports

149 the LiDAR module operates at the 8XX nm wavelength and consists of an emitter (Vertical Cavity  
150 Surface-Emitting Laser with Diffraction Optics Element, VCSEL DOE) and a receptor (Single Photon  
151 Avalanche Diode array-based Near Infrared Complementary Metal Oxide Semiconductor image sensor,  
152 SPAD NIR CMOS) based on direct-time-of-flight technology. Comparisons between the Apple LiDAR  
153 sensor and other types of laser scanners including hand-held, industrial, and terrestrial have been  
154 conducted by several recent studies [Mokroš et al., 2021; Vogt et al., 2021]. Gollob et al. [2021] tested  
155 and reported the performance of a set of eight different scanning apps, and found three applications  
156 including 3D Scanner App, Polycam and SiteScape suitable for actual practice tests. The objective of  
157 this study is not the evaluation of the iPhone LiDAR sensor and app performance. Therefore, the 3D  
158 Scanner App [LABS, 2022] was used with the following settings: confidence = high, range = 5.0 m,  
159 masking = None, and resolution = 5 mm, for scanning and 3D reconstruction processing. The scanned  
160 3D point cloud and its corresponding scalar field are shown in Figure 1b and Figure 1c, respectively.

161 As the LiDAR scanner settings were set at the highest level of accuracy and computational demand,  
162 scanning the whole region of interest at the same time was not possible. So, the experimental region  
163 was divided into several sub-regions and scanned in multi-step. In order to assemble the sub-region  
164 LiDAR scans, several GCPs were considered in the study area. These GCPs were measured by a total  
165 station (Topcon GM Series) and used as landmarks to align distinct 3D point clouds with each other  
166 and create an integrated point cloud encompassing the entirety of the study area.

167 Moreover, several ArUco markers were installed for estimating camera (extrinsic) parameters. In  
168 each setup deployment, these parameters should be recalculated (additional information can be found  
169 in section 4.3). Since it was not possible to accurately measure the real-world coordination of ArUco  
170 markers by the LiDAR scanner, the coordinates of the top-left corner of markers were also measured by  
171 the surveying total station. To establish a consistent coordinate system, the 3D point cloud scanned for  
172 each sub-region was transformed into the total station's coordinate system. The real-world coordinates  
173 of ArUco markers were then added to the 3D point cloud (see Figure 1b).

## 174 4 Methodology

175 This study introduces the Eye of Horus, a vision-based framework for hydrologic monitoring and  
176 real-time water level measurements in bodies of water. The proposed framework includes three main  
177 components. The first step is designing two deployable setups for data collection. These setups consist  
178 of a programmable time-lapse camera run by Raspberry Pi and an ultrasonic sensor run by Arduino.  
179 After collecting data, the first phase (Module 1) involves configuring and training DL-based models  
180 for semantic segmentation of water in the captured images. In the second phase (Module 2), CV  
181 techniques for camera calibration, spatial resection, and calculating projection matrix are discussed.  
182 Finally, in the third phase (Module 3), an ML-based model uses the information achieved by CV  
183 models to find the relationships between real-world coordinates of water level in the captured images  
184 (see Figure 2).

### 185 4.1 Data Acquisition

186 Two different single-board computers (SBC) were used in this study, Raspberry Pi (Zero W) for  
187 capturing time-lapse images of a river scene, and Arduino (Nano 3.x) for measuring water level as the  
188 ground truth data. These devices were designed to communicate with each other, i.e., to trigger the  
189 other to start or stop recording. During capturing time-lapse images, the Pi camera device triggers the  
190 ultrasonic sensor for measuring the corresponding water level. The camera device is equipped with the  
191 Raspberry Pi Camera Module 2 which has a Sony IMX219 8-megapixel sensor. This sensor is able to  
192 capture an image size of  $4,256 \times 2,832$  pixels. However, in this study, the image resolution was set to  
193  $1,920 \times 1,440$  pixels to balance image quality and computational cost in subsequent image processing  
194 steps. This setup is also equipped with a 1200 mAh UPS lithium battery power module to provide  
195 uninterrupted power to the Pi SBC (see Figure 3a).

196 The Arduino-based device records the water level. The design is based on an unmanned aerial ve-  
197 hicle (UAV) deployable sensor created by Smith et al. [2022]. The nRF24L01+ single-chip 2.4 GHz  
198 transceiver allows the Arduino and Raspberry Pi to communicate via radio frequency (RF). The chip

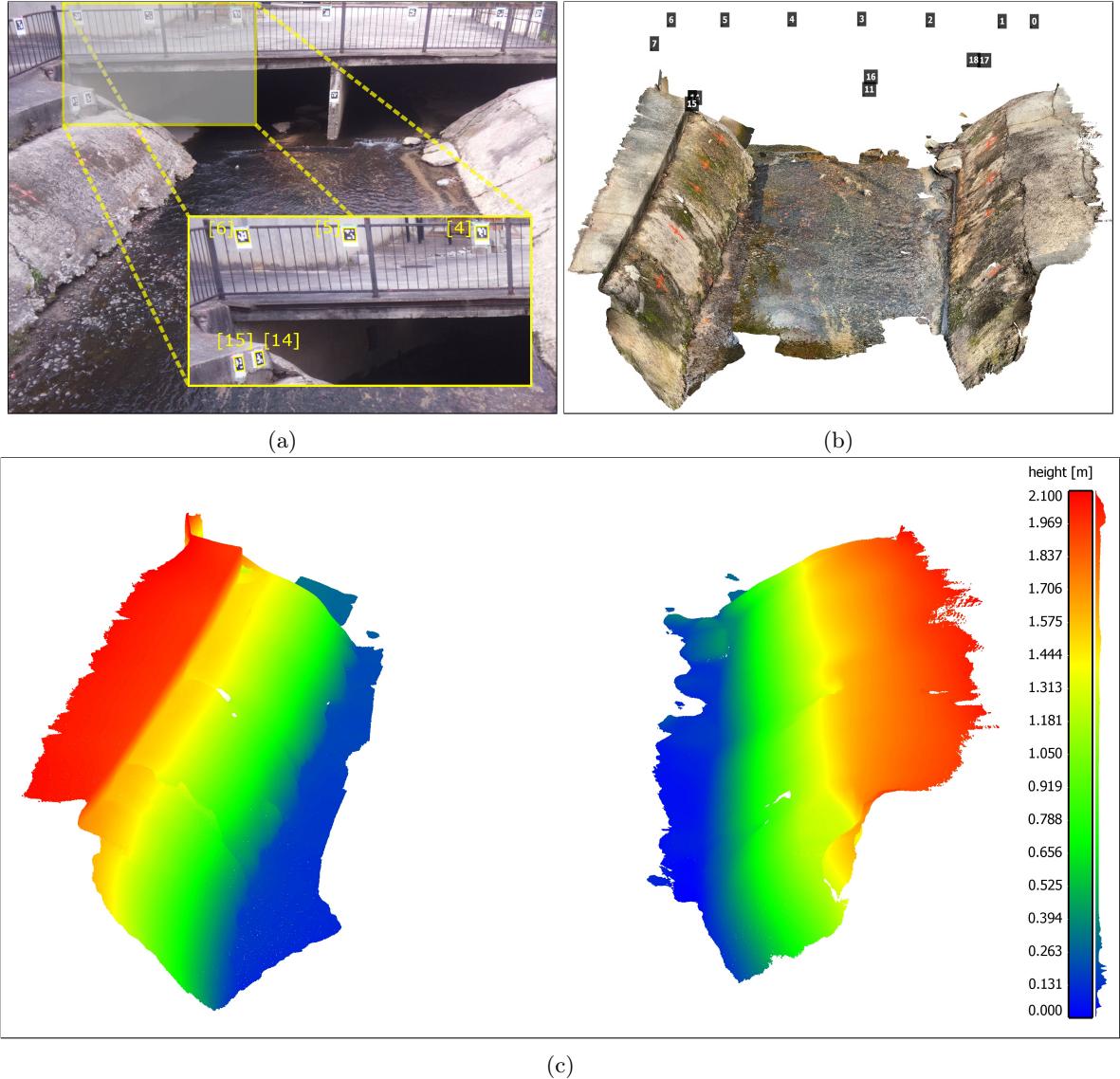


Figure 1: Study area of the Rocky Branch Creek. (a) View of the region of interest, (b) The scanned 3D point cloud of the region of interest including an indication of the ArUco markers' locations, and (c) The scalar field of left and right banks of Rocky Branch in the region of interest (the colorbar and the frequency distribution of  $z$  values for the captured points are shown on the right side).

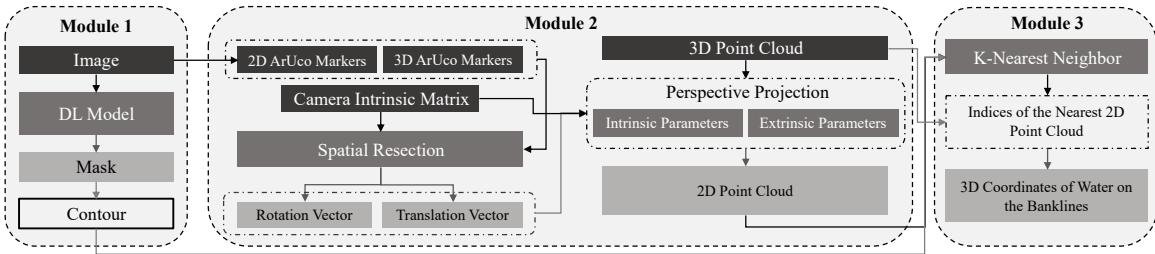


Figure 2: The Eye of Horus workflow includes three main modules starting from processing images captured by the time-lapse camera to estimating water level by projecting the waterline on river banks using CV techniques.

199 is housed in both packages and the channel, pipe addresses, data rate, and transceiver/receiver configuration are all set in the software. The HC-SR04 ultrasonic sensor is mounted to the base of the 200 Arduino device and provides a contactless water level measurement. Two permanent magnets at the 201 top of the housing attach to a ferrous structure and allow the ultrasonic sensor to be suspended up to 202 14 feet over the surface of the water. The device also includes a microSD card module and DS3231 203 real-time clock, which enable data logging and storage on-device as well as transmission. The device 204 is powered by a rechargeable 7.4V 1500 mAh lithium polymer battery (see Figure 3b). 205

206 The Arduino device waits to receive a ping from the Raspberry Pi device to initiate data collection. 207 The ultrasonic sensor measures the distance from the sensor transducer to the surface of the water. 208 The nRF24L01+ transmits this distance to the Raspberry Pi device and saves the measurement and a 209 time stamp from the real-time clock to an onboard microSD card. This acts as backup data storage, in 210 case transmission to the Raspberry Pi fails. The nRF24L01+ RF transceivers have an experimentally 211 determined range of up to 30 ft which allows flexibility in the relative placement of the camera to the 212 measuring site.

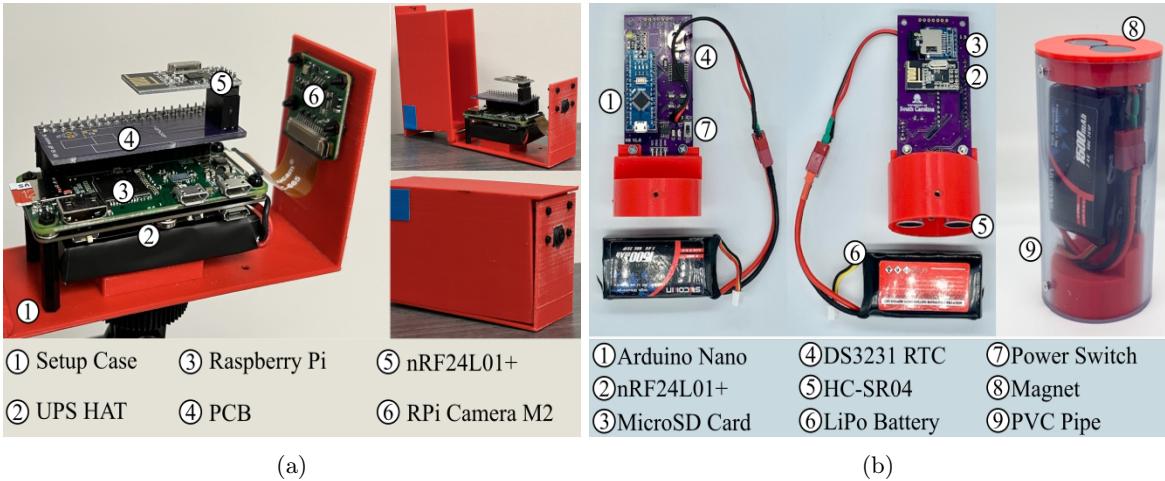


Figure 3: Data acquisition devices. (a) Beena, run by Raspberry Pi (Zero W) for capturing time-lapse images of the river scene; and (b) Aava, run by Arduino Nano for measuring water level correspondence.

213 A dataset for semantic segmentation was created by collecting images from a specific region of interest 214 at different times of the day and under various flow regimes. This dataset includes 1,172 images, with 215 manual annotations of the streamflow in the creek for all of them. The dataset is further divided into 216 812 training images, 124 validation images, and 236 testing images.

## 217 4.2 Deep Learning Model for Water Segmentation

218 The water extent can be automatically determined on the 2D image plane with the help of DL-based 219 models. The task of semantic segmentation was applied within the framework of this study to delineate 220 the water line on the left and right banks of the channel. Three different DL-based models were trained 221 and tested in this study. PSPNet, the first model, is a CNN-based semantic segmentation multi-scale 222 network which can better learn the global context representation of a scene [Zhao et al., 2017]. ResNet- 223 101 [He et al., 2016] was used as the backbone of this model to encode input images into the features. 224 ResNet architecture takes the advantage of “Residual blocks” that assist the flow of gradients during 225 the training stage allowing effective training of deep models even up to hundreds of layers. These 226 extracted features are then fed into a pyramid pooling module in which feature maps produced by 227 small to large kernels are concatenated to distinguish patterns of different scales [Minaee et al., 2021].

228 TransUNet, the second model, is a U-shaped architecture that employs a hybrid of CNN and Trans- 229 formers as the encoder to leverage both the local and global contexts for precise localization and 230 pixel-wise classification [Chen et al., 2021]. In the encoder part of the network, CNN is first used as a 231 feature extractor to generate a feature map for the input image, which is then fed into Transformers 232 to extract long-range dependencies. The resulting features are upsampled in the decoding path and

233 combined with detailed high-resolution spatial information skipped from the CNN to make estimations  
 234 on each pixel of the input image.

235 SegFormer, the third model, unifies a novel hierarchical Transformer, which does not require the position-  
 236 al encodings used in standard Transformers, and MultiLayer Perceptron (MLP) performs efficient  
 237 segmentation [Xie et al., 2021]. The hierarchical Transformer introduced in the encoder of this architec-  
 238 ture gives the model the attention ability to multiscale features (high-resolution fine and low-resolution  
 239 coarse information) in the spatial input without the need for positional encodings that may adversely  
 240 affect a models performance when testing on a different resolution from training. Moreover, unlike  
 241 other segmentation models that typically use deconvolutions in the decoder path, a lightweight MLP  
 242 is employed as the decoder of this network that inputs the features extracted at different stages of  
 243 the encoder to generate a prediction map faster and more efficiently. Two different variants, including  
 244 SegFormer-B0 and SegFormer-B5, were applied in this study. The configuration of the models imple-  
 245 mented in this study is elaborated in Table 1. The total number of parameters (Params), occupied  
 246 memory size on GPU (Total Size), and input image size (Batch Size) are reported in Million (M),  
 247 Megabyte (MB), and Batch size  $\times$  Height  $\times$  Width  $\times$  Channel ( $B, H, W, C$ ) respectively.

Table 1: The configuration of models trained and tested in this study.

Model Names	Params (M)	Total Size (MB)	Batch Size ( $B, H, W, C$ )	Loss Function	Optimizer	LR
PSPNet	66.2	7,178	$2 \times 500 \times 500 \times 3$	Binary Cross Entropy	SGD	2.50E-04
TransUNet	20.1	6,017	$2 \times 448 \times 448 \times 3$	Cross Entropy + Dice	SGD	2.50E-04
SegFormer-B0	3.7	2,217	$2 \times 512 \times 512 \times 3$	Cross Entropy	AdamW	6.00E-05
SegFormer-B5	82.0	27,666	$2 \times 1024 \times 1024 \times 3$	Cross Entropy	AdamW	6.00E-05

248 The models were implemented using PyTorch. During the training procedure, the loss function, opti-  
 249 mizer, and learning rate were set individually for each model based on the results of preliminary runs  
 250 used to find the optimal hyperparameters. In the case of PSPNet and TransUNet, the base learn-  
 251 ing rate was set to  $2.5 \times 10^{-4}$  and decayed using the poly policy [Zhao et al., 2017]. These networks  
 252 were optimized using stochastic gradient descent (SGD) with a momentum of 0.9 and weight decay of  
 253 0.0001. For SegFormer (B0 and B5), a constant learning rate of  $6.0 \times 10^{-5}$  was used, and the networks  
 254 were trained with the AdamW optimizer [Loshchilov and Hutter, 2017]. All networks were trained for  
 255 30 epochs with a batch size of two. The training data for PSPNet and TransUNet were augmented  
 256 with horizontal flipping, random scaling, and random cropping.

### 257 4.3 Projective Geometry

258 In this study, CV techniques are used for different purposes. First, CV models were used for camera  
 259 calibration. They include focal length, optical center, radial distortion, camera rotation, and transla-  
 260 tion. These parameters provide the information (parameters or coefficients) about the camera that is  
 261 required to determine the relationship between 3D object points in the real-world coordinate system  
 262 and its corresponding 2D projection (pixel) in the image captured by that calibrated camera. Gener-  
 263 ally, camera calibration models estimate two kinds of parameters. First, the intrinsic parameters of  
 264 the camera (e.g., focal length, optical center, and radial distortion coefficients of the lens). Second,  
 265 extrinsic parameters (refer to the orientation– rotation, and translation– of the camera) with respect  
 266 to the real-world coordinate system.

267 To estimate the camera intrinsic parameters, OpenCV built-in was applied for camera calibration using  
 268 a 2D checkerboard [Bradski, 2000]. The focal length ( $f_x, f_y$ ), optical centers ( $c_x, c_y$ ), and the skew  
 269 coefficient ( $s$ ) can be used to create a camera intrinsic matrix  $\mathbf{K}$ :

$$\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

270 The camera extrinsic parameters were determined using the pose computation problem, Perspective-n-  
 271 Point (PnP), which consists of solving for the rotation, and translation that minimizes the reprojection

272 error from 2D-3D point correspondences [Marchand et al., 2015]. The PnP estimates the extrinsic  
 273 parameters given a set of ‘object points,’ their corresponding ‘image projections,’ as well as the camera  
 274 intrinsic matrix and the distortion coefficients. The camera extrinsic parameters can be represented  
 275 as a combination of a  $3 \times 3$  rotation matrix  $\mathbf{R}$  and a  $3 \times 1$  translation vector  $\mathbf{t}$ :

$$[\mathbf{R} | \mathbf{t}] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \quad (2)$$

276 Equation 3 represents the ‘Projection Matrix,’ in a homogeneous coordinate system. The projection  
 277 matrix consists of two parts: the intrinsic matrix ( $\mathbf{K}$ ), containing intrinsic parameters, and the extrinsic  
 278 matrix ( $[\mathbf{R} | \mathbf{t}]$ ) which can be represented as follows:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & s & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{[\mathbf{R} | \mathbf{t}]} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (3)$$

279 Direct Linear Transformation (DLT) is a mathematical technique commonly used to estimate the  
 280 parameters of the Projection Matrix. The DLT method requires a minimum of six pairs of known  
 281 3D-2D correspondences to establish twelve equations and estimate all parameters of the Projection  
 282 Matrix. Generally, the intrinsic parameters remain constant for a specific camera model, such as the  
 283 Raspberry Pi Camera Module 2, and can be reused for all images captured by that camera. However,  
 284 the extrinsic parameters change whenever the camera’s location is altered. Consequently, for each  
 285 setup deployment, recalculation of the extrinsic parameters is necessary to reconstruct the Projection  
 286 Matrix. To simplify this process, the PnP method was replaced with DLT. It can reduce the required  
 287 number of 3D-2D correspondence pairs to three, by reusing the intrinsic parameters.

288 Additionally, ArUco markers were incorporated to represent pairs of known 3D-2D correspondences.  
 289 For this purpose, the pixel coordinates of ArUco markers were determined using the OpenCV ArUco  
 290 marker detection module on the 2D image plane, and the corresponding 3D real-world coordinates  
 291 were measured by the total station. With these 3D-2D point correspondences, the spatial position  
 292 and orientation of the camera can be estimated for each setup deployment. After retrieving all the  
 293 necessary parameters, a full-perspective camera model can be generated. Using this model, the 3D  
 294 point cloud is projected onto the 2D image plane. The projected (2D) point cloud represents the 3D  
 295 real-world coordinates of the nearest 2D pixel correspondence on the image plane

#### 296 4.4 Machine Learning for Image Measurements

297 Using the projection matrix, the 3D point cloud is projected on the 2D image plane (see Figure 4). The  
 298 projected (2D) point cloud is intersected with the water line pixels, the output of the DL-based model  
 299 (Module 1), to find the nearest point cloud coordinate. To achieve this objective, we utilize the K-  
 300 Nearest Neighbors (KNN) algorithm. Notably, the indices of the selected points remain consistent for  
 301 both the 3D point cloud and the projected (2D) correspondences. As a result, by utilizing the indices  
 302 of the chosen projected (2D) points, the corresponding real-world 3D coordinates can be retrieved.

#### 303 4.5 Performance Metrics

304 The performance of the proposed framework is evaluated based on four different metrics including  
 305 coefficient of determination ( $r^2$ ), Nash-Sutcliffe Efficiency (NSE), Root Mean Square Error (RMSE),  
 306 and Percent bias (PBIAS).  $R^2$  is a widely used metric that quantifies how much of the observed  
 307 dispersion can be explained in a linear relationship by the prediction.

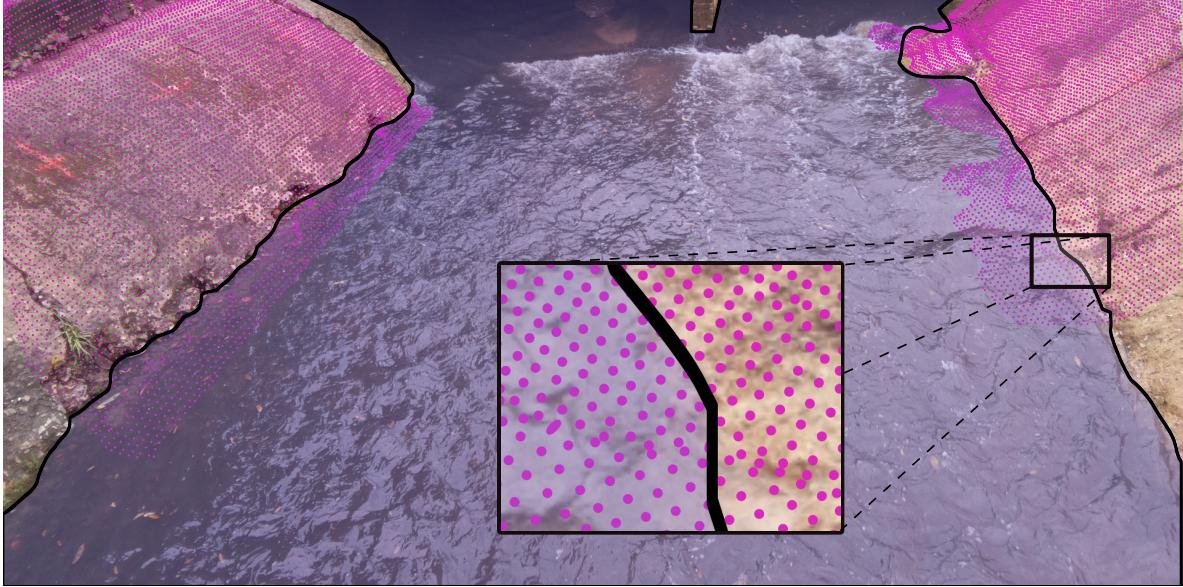


Figure 4: KNN is used to find the nearest projected (2D) point cloud (magenta dots) to the water line (black line) on the image plane.

$$r^2 = \left( \frac{\sum_{i=1}^n (O_i - \bar{O})(P_i - \bar{P})}{\sqrt{\sum_{i=1}^n (O_i - \bar{O})^2 \cdot \sum_{i=1}^n (P_i - \bar{P})^2}} \right)^2 \quad (4)$$

308 However, if the model systematically over- or under-estimates the results,  $r^2$  will still be close to 1.0  
 309 as it only takes dispersion into account [Krause et al., 2005]. NSE, another commonly used metric  
 310 in hydrology, presents the model performance with an interpretable scale and is used to differentiate  
 311 between ‘good’ and ‘bad’ models [Knoben et al., 2019].

$$NSE = 1 - \frac{\sum_{i=1}^n (O_i - P_i)^2}{\sum_{i=1}^n (O_i - \bar{O})^2} \quad (5)$$

312 RMSE represents the square root of the average of squares of the errors, the differences between  
 313 predicted values and observed values.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (O_i - P_i)^2} \quad (6)$$

314 The PBIAS of estimated water level, compared against the ultrasonic sensor data was also used to  
 315 show where the two estimates are close to each other and where they significantly diverge [Lin et al.,  
 316 2020].

$$PBIAS = \frac{100}{n} \sum_{i=1}^n \frac{(O_i - P_i)}{\sum_{i=1}^n O_i} \quad (7)$$

317 Where  $n$  is the number of data points,  $O$  and  $P$  are observed and predicted values, respectively.

## 318 5 Results and Discussion

319 The results of this study are presented in two sections. First, the performance of DL-based models is  
320 discussed. Then, in the second section, the performance of the proposed framework is evaluated for  
321 five different deployments.

### 322 5.1 DL-based Models Results

323 The performance of DL-based models for the task of semantic segmentation is evaluated and compared  
324 in this section. Since the proposed dataset includes just two classes, “river” and “non-river”, “non-river”  
325 was omitted from the evaluation process, and the performance of models is only reported for the  
326 “river” class of the test set. The class-wise intersection over union (IoU) and the per-pixel accuracy  
327 (ACC) were considered the main evaluation metrics in this study. According to Table 2, both variants  
328 of SegFormer– SegFormer-B0, and SegFormer-B5– outperform other semantic segmentation networks  
329 on the test set. Considering the models’ configurations detailed in Table 1, SegFormer-B0 can be  
330 considered the most efficient DL-based network, as it is comprised of only 3.7 M trainable parameters  
331 and occupies just 2,217 Megabytes of GPU ram during training. In Figure 5, four different visual  
332 representations of the models’ performance on the validation set of the proposed dataset are presented.  
333 Since the water level is estimated by intersecting the water line on river banks with the projected (2D)  
334 point cloud, precise delineation of the water line is of utmost importance to achieve better results in  
335 the following steps. This means that estimating the correct location of the water line on creek banks in  
336 each time-lapse image plays a more significant role than performance metrics in this study. Taking the  
337 quality of water line detection into account and based on the visual representations shown in Figure 5,  
338 SegFormers’ variants still outperform DL-based approaches. In this regard, a comparison of PSPNet  
339 and TransUNet showed that PSPNet can delineate the water line more clearly, while the segmented  
340 area is more integrated for TransUNet outputs.

Table 2: The performance metrics of different DL-based approaches.

Model Names	IoU (River)	ACC (River)
PSPNet	94.88%	95.84%
TransUNet	93.54%	96.89%
SegFormer-B0	99.38%	99.77%
SegFormer-B5	99.55%	99.81%

341 CNNs are typically limited by the nature of their convolution operations, leading to architecture-  
342 specific issues such as locality [Geirhos et al., 2018a]. Consequently, CNN-based models may achieve  
343 high accuracy on training data, but their performance can decrease considerably on unseen data.  
344 Additionally, compared to Transformer-based networks, they perform poorly at detecting semantics  
345 that requires combining long- and short-range dependencies. Transformers can relax the biases of  
346 DL-based models induced by Convolutional operations, achieving higher accuracy in localization of  
347 target semantics and pixel-level classification with lower fluctuations in varied situations through the  
348 leverage of both local and global cues [Naseer et al., 2021]. Yet, various transformer-based networks  
349 may perform differently depending on the targeted task and the network’s architecture. TransUNet  
350 adopts Transformers as part of its backbone; however, Transformers generate single-scale low-resolution  
351 features as output [Xie et al., 2021], which may limit the accuracy when multi-scale objects or single  
352 objects with multi-scale features are segmented. The problem of producing single-scale features in  
353 standard Transformers is addressed in SegFormer variants through the use of a novel hierarchical  
354 Transformer encoder [Xie et al., 2021]. This approach has resulted in human-level accuracy being  
355 achieved by Segformer-B0 and -B5 in the delineation of the water line, as shown in Figure 5. The  
356 predicted masks are in satisfactory agreement with the manually annotated images.

### 357 5.2 Water Level Estimation

358 This section reports the framework performance based on several deployments in the field. The perfor-  
359 mance results are separately shown for the left and right banks and compared with ultrasonic sensor  
360 data as the ground truth. The ultrasonic sensor was evaluated previously that documented an average

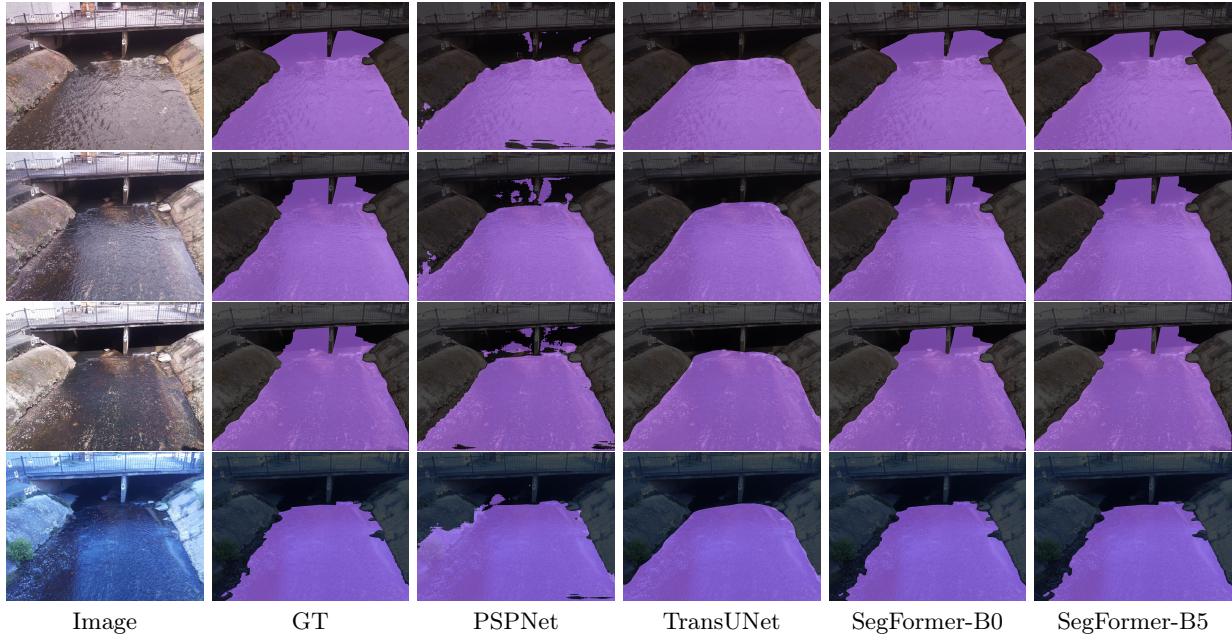


Figure 5: Visual representations of different DL-based image segmentation approaches on the validation dataset.

361 distance error of 6.9 mm [Smith et al., 2022]. The setup was deployed on several rainy days. The  
 362 results of each deployment are reported in Table 3.

Table 3: The performance metrics of the framework for five different days of setup deployment.

Deployment Date	Position	Metrics			
		$r^2$	NSE	RMSE	PBIAS
Aug/17/2022	Left Bankline	0.8019	0.5258	0.0409	10.6401
	Right Bankline	0.7932	0.7541	0.0294	-0.4848
Aug/19/2022	Left Bankline	0.7701	0.5713	0.0647	16.1015
	Right Bankline	0.9678	0.9588	0.0201	-3.4752
Aug/25/2022	Left Bankline	0.7690	0.5700	0.0435	-7.7091
	Right Bankline	0.8922	0.8711	0.0238	-1.7738
Nov/10/2022	Left Bankline	0.9461	0.8129	0.0511	-13.1183
	Right Bankline	0.9857	0.9790	0.0171	-1.5210
Nov/11/2022	Left Bankline	0.9588	0.8881	0.0397	-10.3656
	Right Bankline	0.9855	0.9829	0.0155	-1.7987

363 In addition to Table 3, the results of each deployment are visually demonstrated in Figure 6. The scatter  
 364 plots show the relationships between the ground truth data (measured by the ultrasonic sensor), and  
 365 the banks of the river. The scatter plots visually present whether the camera readings overestimate or  
 366 underestimate the ground truth data. Moreover, the time-series plot of water level is shown for each  
 367 deployment separately. A hydrograph, showing changes in the water level of a stream over time can  
 368 be a useful tool for demonstrating whether camera readings can satisfactorily capture the response  
 369 of a catchment area to rainfall. The proposed framework can be evaluated in terms of its ability to  
 370 accurately track and identify important characteristics of a flood wave, such as the rising limb, peak,  
 371 and recession limb.

372 The first deployment was done on Aug 17, 2022 (see Figure 6a). The initial water level of the base  
 373 flow and parts of the rising limb were not captured in this deployment. Table 3 shows that the  
 374 performance results of the right bank camera readings are better than those of the left bank.  $R^2$  for  
 375 both banks was about 0.80 showing a strongly related correlation between the water level estimated by

376 the framework and ground truth data. Figure 6a shows how the left and right bank camera readings  
377 perform during the rising limb; the right bank camera readings still underestimated the water level  
378 during this time frame, and during the recession limb, the left bank camera readings overestimated  
379 the water level. However, the hydrograph plot shows that both left and right bank camera readings  
380 were able to capture the peak water level.

381 The second deployment was done on Aug 19, 2022. In this deployment, all segments of the hydrograph  
382 were captured. According to Table 3, the performance of the right bank camera readings was better  
383 than the left bank one; more than 0.95 was reported for  $R^2$  and NSE of the right bankline. Figure 6b  
384 shows during the rising limb and crest segment both banks estimated the water level similar to ground  
385 truth. During the recession limb, the right bank water level estimation kept coincident with ground  
386 truth, while the left bank overestimated the water level. The third deployment was on Aug 25, 2022.  
387 This time water level of the recession limb and the following base flow were captured (see Figure 6c).  
388 The right bank camera readings with  $R^2$  of 0.89 performed better than the left bank. This time, left  
389 bank camera readings underestimated the water level over the recession limb, but during the following  
390 base flow, the water level was estimated correctly by cameras on both banks.

391 The results indicate that the right bank camera readings performed better than the left bank. Further  
392 investigation of the field conditions revealed that stream erosion had a more significant impact on the  
393 concrete surface of the left bank, resulting in patches and holes that were not scanned by the iPhone  
394 LiDAR. As a result, the KNN algorithm used to find the nearest (2D) point cloud coordinates to the  
395 water line could not accurately represent the corresponding real-world coordinates of these locations.  
396 Figure 7 shows a box plot and scatter plot of the estimated water level for a time-lapse image captured  
397 at 13:29 on Aug 19, 2022. The patches and holes on the left bank surface caused instability in water  
398 level estimation for the region of interest. The box plot of the left bank (Cam-L-BL) was taller than  
399 that of the right bank (Cam-R-BL), indicating that the estimated water level was spread over larger  
400 values in the left bank due to the presence of these irregularities.

401 After analyzing the initial results, the deployable setups were modified to enhance the quality of data  
402 collection. The programming code of the Arduino device, Aava, was modified to measure five different  
403 records for water level, each time it is triggered by the camera device, Beena, and transmit the average  
404 distance to the Raspberry Pi device. This modification decreased the number of noise spikes in the  
405 measured data and allowed a better comparison between camera readings and ground truth data.  
406 The case of the camera device, Beena, was redesigned to protect the single board against rain without  
407 requiring an umbrella which makes the camera setup unstable in stormy weather and causes a decrease  
408 in the precision of measurements. Moreover, an opening is incorporated into the redesigned case to  
409 connect an external power bank to enhance the run time. Finally, the viewpoint of the camera was  
410 subtly shifted to the right to adjust the share of the river banks on the camera's field of view.

411 The results of the deployments on Nov 10, 2022, and Nov 11, 2022, demonstrate that modifications  
412 to the setup have significantly improved the results of the left bank (as shown in Table 3). NSE  
413 improved from approximately 0.55 for the first three setup deployments to over 0.80 for the modified  
414 deployments. Figure 8 shows the setup performances during all segments of the flood wave. The peaks  
415 were captured by the right bankline on both deployment dates, and there was no effect of noisy spikes  
416 on either camera readings or ground truth data. However, the right bank images still underestimated  
417 the water level during the rainstorms.

## 418 6 Conclusion

419 This study introduced Eye of Horus, a vision-based framework for hydrologic monitoring and measuring  
420 real-time water-related parameters, e.g., water level, from surveillance images captured during flood  
421 events. Time-lapse images and real water level correspondences were collected by Raspberry Pi camera  
422 and Arduino HC-SR05 ultrasonic sensor, respectively. Moreover, Computer Vision and Deep Learning  
423 techniques were used for semantic segmentation of water surface within the captured images and for  
424 reprojecting the 3D point cloud constructed with an iPhone LiDAR scanner, on the (2D) image plane.  
425 Eventually, the K-Nearest Neighbor algorithm was used to intersect the projected (2D) point cloud  
426 with the water line pixels extracted from the output of the Deep Learning model, to find the real-world  
427 3D coordinates.

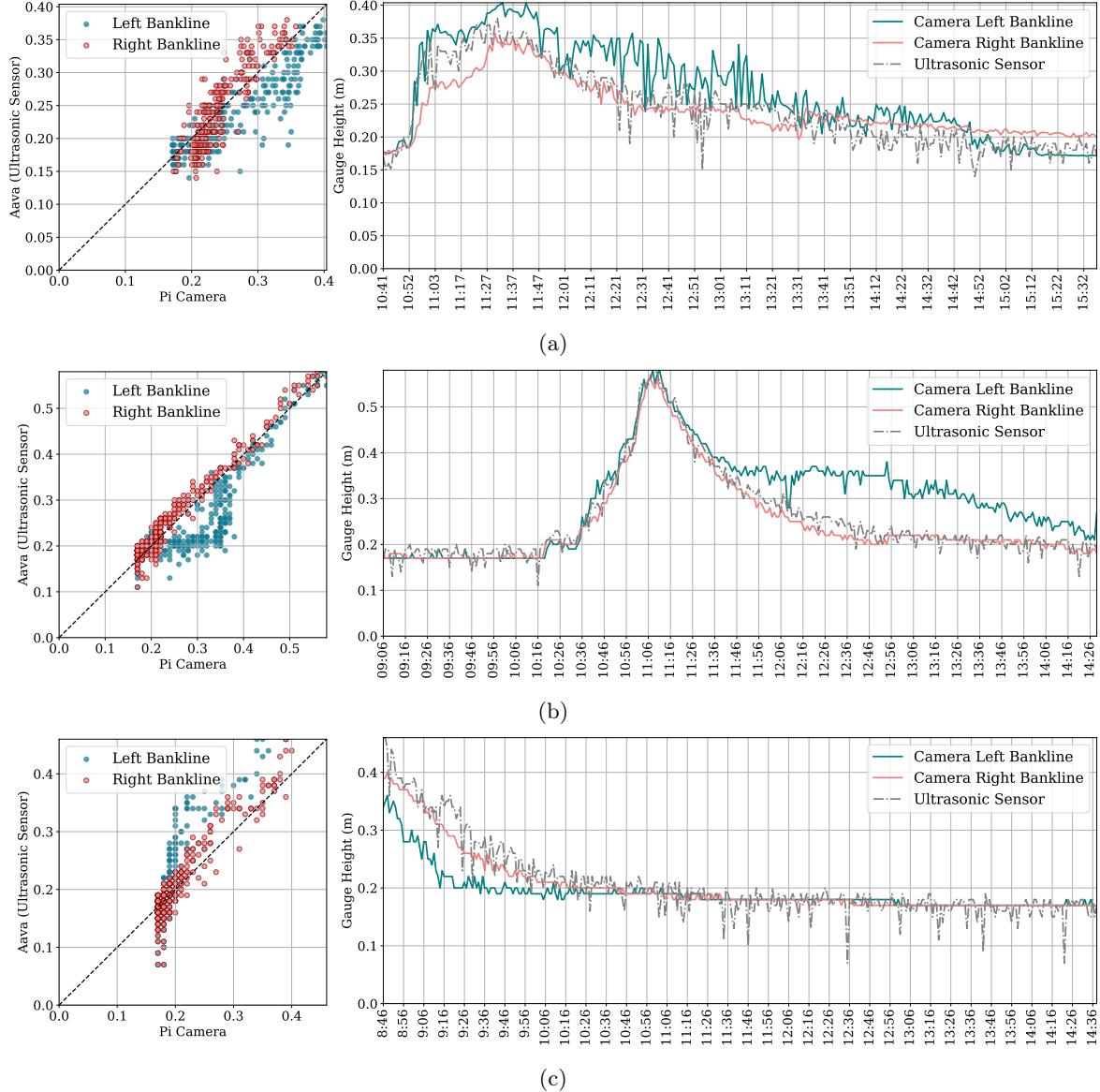


Figure 6: Scatter plot and time series plot for estimated water level by the proposed framework and measured by the ultrasonic sensor for setup deployment on (a) Aug 17, 2022 (b) Aug 19, 2022, and (c) Aug 25, 2022.

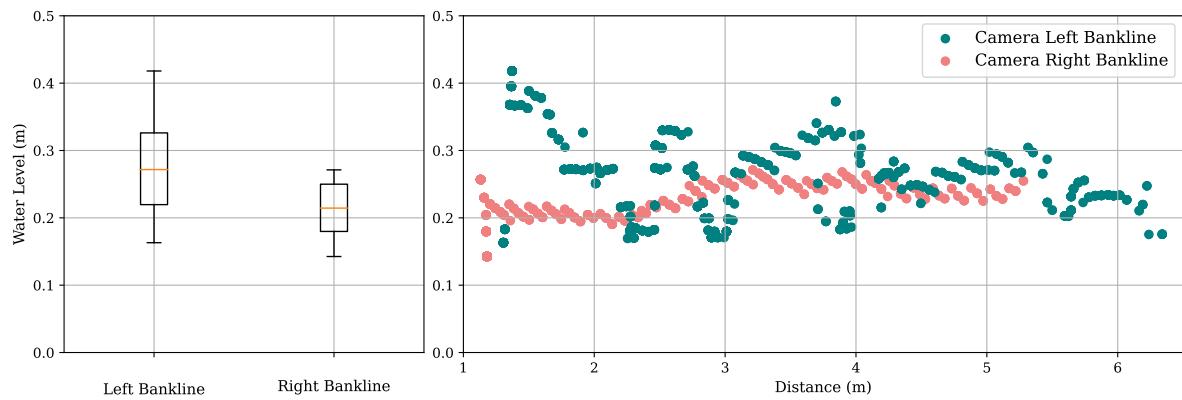


Figure 7: Water level fluctuation along both left and right banks for the flow regime for an image captured at 13:29 on Aug 19, 2022.

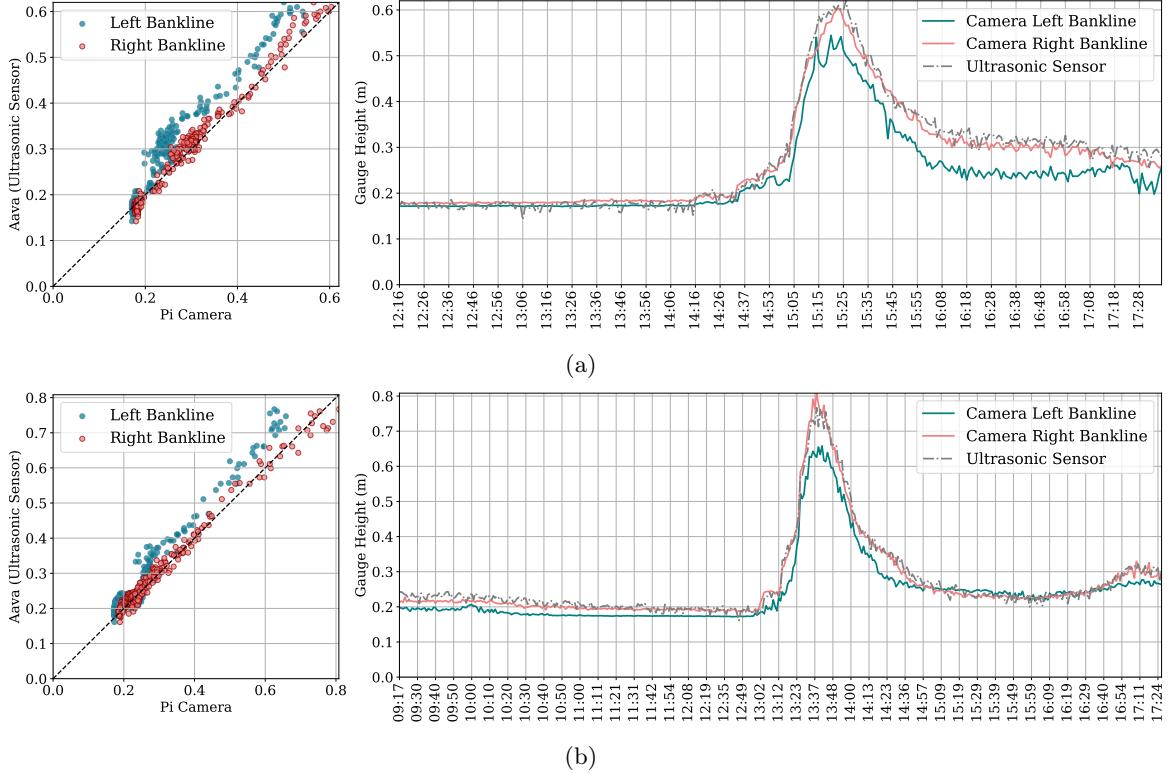


Figure 8: Scatter plot and time series plot for estimated water level by the proposed framework and measured by the ultrasonic sensor for setup deployment on (a) Nov 10, 2022, and (b) Nov 11, 2022.

428 A vision-based framework offers a new alternative to current hydrologic data collection and real-  
 429 time monitoring systems. Hydrological models require geometric information for estimating discharge  
 430 routing parameters, stage, and flood inundation maps. However, determining bankfull characteristics  
 431 is a challenge due to natural or anthropogenic down-cutting of streams. Using visual sensing, stream  
 432 depth, water velocity, and instantaneous streamflow at bankfull stage can be reliably measured.

## 433 7 Data Availability Statement

434 The framework and codes developed and used in this study are publicly available online in the GitHub  
 435 repository (<https://github.com/smhassanerfani/horus>).

## 436 References

437 Douglas E Alsdorf, Ernesto Rodríguez, and Dennis P Lettenmaier. Measuring surface water from  
 438 space. *Reviews of Geophysics*, 45(2), 2007.

439 Vijay Badrinarayanan, Ankur Handa, and Roberto Cipolla. Segnet: A deep convolutional encoder-  
 440 decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293*, 2015.

441 G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

442 Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille,  
 443 and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation.  
 444 *arXiv preprint arXiv:2102.04306*, 2021.

445 Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab:  
 446 Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected  
 447 crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2017.

448 Andrea De Cesarei, Shari Cavicchi, Giampaolo Cristadoro, and Marco Lippi. Do humans and deep con-  
449 volutional neural networks use visual information similarly for the categorization of natural scenes?  
450 *Cognitive Science*, 45(6):e13009, 2021.

451 Samuel Dodge and Lina Karam. A study and comparison of human and deep learning recognition  
452 performance under visual distortions. In *Int. Conf. Comput. Communication and Networks*, pages  
453 1–7. IEEE, 2017.

454 Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas  
455 Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image  
456 is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*,  
457 2020.

458 Melanie Elias, Anette Eltner, Frank Liebold, and Hans-Gerd Maas. Assessing the influence of temper-  
459 ature changes on the geometric stability of smartphone-and raspberry pi cameras. *Sensors*, 20(3):  
460 643, 2020.

461 Anette Eltner and Danilo Schneider. Analysis of different methods for 3d reconstruction of natural  
462 surfaces from parallel-axes uav images. *The Photogrammetric Record*, 30(151):279–299, 2015.

463 Anette Eltner, Andreas Kaiser, Carlos Castillo, Gilles Rock, Fabian Neugirg, and Antonio Abellán.  
464 Image-based surface reconstruction in geomorphometry—merits, limits and developments. *Earth  
465 Surface Dynamics*, 4(2):359–389, 2016.

466 Anette Eltner, Melanie Elias, Hannes Sardemann, and Diana Spieler. Automatic image-based water  
467 stage measurement for long-term observations in ungauged catchments. *Water Resources Research*,  
468 54(12):10–362, 2018.

469 Anette Eltner, Patrik Olá Bressan, Thales Akiyama, Wesley Nunes Gonçalves, and Jose Marcato Ju-  
470 nior. Using deep learning for automatic water stage measurements. *Water Resources Research*, 57  
471 (3):e2020WR027608, 2021.

472 Seyed Mohammad Hassan Erfani, Zhenyao Wu, Xinyi Wu, Song Wang, and Erfan Goharian. Atlantis:  
473 A benchmark for semantic segmentation of waterbody images. *Environmental Modelling & Software*,  
474 149:105333, 2022.

475 David A Forsyth and Jean Ponce. *Computer vision: a modern approach*. prentice hall professional  
476 technical reference, 2002.

477 Laurent Froideval, Kevin Pedoja, Franck Garestier, Pierre Moulon, Christophe Conessa, Xavier Pellerin  
478 Le Bas, Kalil Traoré, and Laurent Benoit. A low-cost open-source workflow to generate georeferenced  
479 3d sfm photogrammetric models of rocky outcrops. *The Photogrammetric Record*, 34(168):365–384,  
480 2019.

481 Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention  
482 network for scene segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3146–3154,  
483 2019.

484 Asmamaw Gebrehiwot, Leila Hashemi-Beni, Gary Thompson, Parisa Kordjamshidi, and Thomas E  
485 Langan. Deep convolutional neural network for flood extent mapping using unmanned aerial vehicles  
486 data. *Sensors*, 19(7):1486, 2019.

487 Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland  
488 Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves  
489 accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018a.

490 Robert Geirhos, Carlos RM Temme, Jonas Rauber, Heiko H Schütt, Matthias Bethge, and Felix A  
491 Wichmann. Generalisation in humans and deep neural networks. *Adv. Neural Inform. Process. Syst.*,  
492 31, 2018b.

493 Robert Geirhos, Kristof Meding, and Felix A Wichmann. Beyond accuracy: quantifying trial-by-trial  
494 behaviour of cnns and humans by measuring error consistency. *Adv. Neural Inform. Process. Syst.*,  
495 33:13890–13902, 2020.

496 Troy E Gilmore, François Birgand, and Kenneth W Chapman. Source and magnitude of error in an  
 497 inexpensive image-based water level measurement system. *Journal of hydrology*, 496:178–186, 2013.

498 Christoph Gollob, Tim Ritter, Ralf Kraßnitzer, Andreas Tockner, and Arne Nothdurft. Measurement  
 499 of forest inventory parameters with Apple iPad pro and integrated LiDAR technology. *Remote  
 500 Sensing*, 13(16):3129, 2021.

501 Michael F Goodchild. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):  
 502 211–221, 2007.

503 Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recogni-  
 504 tion. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016.

505 Jeff Howe. *Crowdsourcing: How the power of the crowd is driving the future of business*. Random  
 506 House, 2008.

507 Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet:  
 508 Criss-cross attention for semantic segmentation. In *Int. Conf. Comput. Vis.*, pages 603–612, 2019.

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

511 Tyler V King, Bethany T Neilson, and Mitchell T Rasmussen. Estimating discharge in low-order rivers  
 512 with high-resolution aerial imagery. *Water Resources Research*, 54(2):863–878, 2018.

513 Wouter JM Knoben, Jim E Freer, and Ross A Woods. Inherent benchmark or not? comparing nash–  
 514 sutcliffe and kling–gupta efficiency scores. *Hydrology and Earth System Sciences*, 23(10):4323–4331,  
 515 2019.

516 Peter Krause, DP Boyle, and Frank Bäse. Comparison of different efficiency criteria for hydrological  
 517 model assessment. *Advances in Geosciences*, 5:89–97, 2005.

518 LAAN LABS. 3D Scanner App – LiDAR Scanner for iPad Pro & iPhone Pro. Available online:  
 519 <https://3dscannerapp.com/>, 2022. Accessed on Sep 16, 2022.

520 Xia Li, Zhisheng Zhong, Jianlong Wu, Yibo Yang, Zhouchen Lin, and Hong Liu. Expectation-  
 521 maximization attention networks for semantic segmentation. In *Int. Conf. Comput. Vis.*, pages  
 522 9167–9176, 2019.

523 Zhenlong Li, Cuizhen Wang, Christopher T Emrich, and Diansheng Guo. A novel approach to leverag-  
 524 ing social media for rapid flood mapping: a case study of the 2015 south carolina floods. *Cartography  
 525 and Geographic Information Science*, 45(2):97–110, 2018.

526 Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks  
 527 for high-resolution semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1925–  
 528 1934, 2017.

529 Peirong Lin, Ming Pan, George H Allen, Renato Prata de Frasson, Zhenzhong Zeng, Dai Yamazaki,  
 530 and Eric F Wood. Global estimates of reach-level bankfull river width leveraging big data geospatial  
 531 analysis. *Geophysical Research Letters*, 47(7):e2019GL086405, 2020.

532 Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo.  
 533 Swin transformer: Hierarchical vision transformer using shifted windows. In *Int. Conf. Comput.  
 534 Vis.*, pages 10012–10022, 2021.

535 Shi-Wei Lo, Jyh-Horng Wu, Fang-Pang Lin, and Ching-Han Hsu. Visual sensing for urban flood  
 536 monitoring. *Sensors*, 15(8):20006–20029, 2015.

537 Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic seg-  
 538 mentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3431–3440, 2015.

539 Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint  
 540 arXiv:1711.05101*, 2017.

541 Gregor Luetzenburg, Aart Kroon, and Anders A Bjørk. Evaluation of the apple iphone 12 pro lidar  
 542 for an application in geosciences. *Scientific reports*, 11(1):1–9, 2021.

543 Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality: a  
544 hands-on survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):2633–2651, 2015.

545 Shervin Minaee, Yuri Y Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri  
546 Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Trans. Pattern Anal. Mach.  
547 Intell.*, 2021.

548 Martin Mokroš, Tomáš Mikita, Arunima Singh, Julián Tomaštík, Juliána Chudá, Piotr Węzyk, Karel  
549 Kuželka, Peter Surový, Martin Klimánek, Karolina Zięba-Kulawik, et al. Novel low-cost mobile  
550 mapping systems for forest inventories as terrestrial laser scanning alternatives. *International Journal  
551 of Applied Earth Observation and Geoinformation*, 104:102512, 2021.

552 Mohamed M Morsy, Jonathan L Goodall, Fadi M Shatnawi, and Michael E Meadows. Distributed  
553 stormwater controls for flood mitigation within urbanized watersheds: case study of rocky branch  
554 watershed in columbia, south carolina. *Journal of Hydrologic Engineering*, 21(11):05016025, 2016.

555 Matthew Moy de Vitry, Simon Kramer, Jan Dirk Wegner, and João P Leitão. Scalable flood level  
556 trend monitoring with surveillance cameras using a deep convolutional neural network. *Hydrology  
557 and Earth System Sciences*, 23(11):4621–4634, 2019.

558 Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad  
559 Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Adv. Neural  
560 Inform. Process. Syst.*, 34:23296–23308, 2021.

561 Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic  
562 segmentation. In *Int. Conf. Comput. Vis.*, pages 1520–1528, 2015.

563 R.J Pally and S Samadi. Application of image processing and convolutional neural networks for flood  
564 image classification and semantic segmentation. *Environmental Modelling & Software*, 148:105285,  
565 2022.

566 George Panteras and Guido Cervone. Enhancing the temporal resolution of satellite-based flood ex-  
567 tent generation using crowdsourced data for disaster monitoring. *International Journal of Remote  
568 Sensing*, 39(5):1459–1474, 2018.

569 E Schnebele, G Cervone, and N Waters. Road assessment after flood events using non-authoritative  
570 data. *Natural Hazards and Earth System Sciences*, 14(4):1007, 2014.

571 Elyas Asadi Shamsabadi, Chang Xu, and Daniel Dias-da Costa. Robust crack detection in masonry  
572 structures with transformers. *Measurement*, 200:111590, 2022.

573 Corinne Smith, Joud Satme, Jacob Martin, Austin R.J. Downey, Nikolaos Vitzilaios, and Jasim Imran.  
574 UAV rapidly-deployable stage sensor with electro-permanent magnet docking mechanism for flood  
575 monitoring in undersampled watersheds. *HardwareX*, 12:e00325, oct 2022. doi: 10.1016/j.ohx.2022.  
576 e00325.

577 Stefano Tavani, Andrea Billi, Amerigo Corradetti, Marco Mercuri, Alessandro Bosman, Marco Cuf-  
578 faro, Thomas Seers, and Eugenio Carminati. Smartphone assisted fieldwork: Towards the digital  
579 transition of geoscience fieldwork using lidar-equipped iphones. *Earth-Science Reviews*, 227:103969,  
580 2022.

581 Ryota Tsubaki, Ichiro Fujita, and Shiho Tsutsumi. Measurement of the flood discharge of a small-sized  
582 river using an existing digital video recording system. *Journal of Hydro-environment Research*, 5  
583 (4):313–321, 2011.

584 D Phil Turnipseed and Vernon B Sauer. Discharge measurements at gaging stations. Technical report,  
585 US Geological Survey, 2010.

586 Remy Vandaele, Sarah L Dance, and Varun Ojha. Deep learning for automated river-level monitoring  
587 through river-camera images: an approach based on water segmentation and transfer learning.  
588 *Hydrology and Earth System Sciences*, 25(8):4435–4453, 2021.

589 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz  
590 Kaiser, and Illia Polosukhin. Attention is all you need. *Adv. Neural Inform. Process. Syst.*, 30, 2017.

591 Maximilian Vogt, Adrian Rips, and Claus Emmelmann. Comparison of ipad pro®’s lidar and  
592 truedepth capabilities with an industrial 3d scanning solution. *Technologies*, 9(2):25, 2021.

593 Matthew J Westoby, James Brasington, Niel F Glasser, Michael J Hambrey, and Jennifer M Reynolds.  
594 ‘structure-from-motion’photogrammetry: A low-cost, effective tool for geoscience applications. *Ge-  
595 omorphology*, 179:300–314, 2012.

596 Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer:  
597 Simple and efficient design for semantic segmentation with transformers. *Adv. Neural Inform. Pro-  
598 cess. Syst.*, 34:12077–12090, 2021.

599 Yuhui Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint  
600 arXiv:1809.00916*, 2018.

601 Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmen-  
602 tation. In *Eur. Conf. Comput. Vis.*, pages 173–190. Springer, 2020.

603 Zhen Zhang, Yang Zhou, Haiyun Liu, and Hongmin Gao. In-situ water level measurement using  
604 nir-imaging video camera. *Flow Measurement and Instrumentation*, 67:95–106, 2019.

605 Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing  
606 network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages  
607 2881–2890, 2017.

608 Yufeng Zheng, Jun Huang, Tianwen Chen, Yang Ou, and Wu Zhou. Processing global and local features  
609 in convolutional neural network (cnn) and primate visual systems. In *Mobile Multimedia/Image  
610 Processing, Security, and Applications 2018*, volume 10668, pages 44–51. SPIE, 2018.

611 Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural  
612 networks for semantic segmentation. In *Int. Conf. Comput. Vis.*, pages 593–602, 2019.