Summer 2023

Developing a Vision-Based Framework for Measuring and Monitoring Water Resource Systems Using Computer Vision and Deep Learning Techniques

Seyed Mohammad Hassan Erfani

Developing a Vision-based Framework for Measuring and Monitoring Water Resource Systems Using Computer Vision and Deep Learning Techniques

by

Seyed Mohammad Hassan Erfani

Bachelor of Science
Islamic Azad University of Mashhad 2011

Master of Science
Ferdowsi University of Mashhad 2015

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy in

Civil and Environmental Engineering

College of Engineering and Computing

University of South Carolina

2023

Accepted by:

Erfan Goharian, Major Professor

Jasim Imran, Committee Member

Yu Qian, Committee Member

Yan Tong, Outside Committee Member

Ann Vail, Dean of the Graduate School

Dedication

To my incredible mother, who embodies unwavering strength. In the face of numerous losses and disappointments that life presents, she has consistently demonstrated her remarkable resilience and determination.

Acknowledgments

I am deeply grateful to the numerous individuals who have played a significant role in the completion of this dissertation. Their unwavering support, guidance, and contributions have been instrumental in shaping and enriching this research.

First and foremost, I extend my sincere appreciation to my esteemed academic advisors and educators, Dr. Erfan Goharian, Dr. Jasim Imran, Dr. Song Wang, Dr. Yan Tong, and Dr. Austin Downey. Their expertise, mentorship, and valuable insights have been invaluable throughout this journey. Their dedication to my academic growth and their unwavering belief in my abilities have been a constant source of motivation and inspiration.

I would also like to express my gratitude to my cohorts, colleagues, and co-authors, Mahdi Erfani, Ahad Hassan Tanim, Dr. Zhenyao Wu, Dr. Xinyi Wu, and Corinne Smith. Their collaborative spirit, intellectual discussions, and willingness to share their expertise have significantly influenced the development and refinement of this work. Their constructive feedback and valuable suggestions have been instrumental in improving the quality and depth of my research.

Finally, I am grateful to all my friends, family, and loved ones for their unwavering support, patience, and encouragement throughout this challenging academic endeavor. Their belief in my abilities and their constant presence have been a tremendous source of strength and motivation.

Abstract

Increased vulnerability of water systems to extreme events and climate change is among the profound challenges facing the management of water resource systems around the world. Extreme events, including droughts, floods, and natural hazards have become more frequent and intensive, particularly in coastal regions. Floods, for instance, caused tens of billions of US dollars losses and put the lives of thousands in danger, globally. To cope with the adverse consequences of floods, a wide range of structural, non-structural, and emergency measures are studied and deployed by flood management sectors. Various flood simulation, mapping, and forecast systems have been developed to predict flood events, warn the public and inform decision-makers to react accordingly for making better decisions to protect lives and property. Development and level of accuracy of these systems, however, rely heavily on the availability and quality of temporal and spatial data received from ground-based gauge sensing, remote sensing, and more recently crowdsourcing. While all these data sources provide useful information, they have their limitations, such as the small spatial scale of in-situ gauging or satellites' long revisit period. Recent advancements in Artificial Intelligence provide a unique opportunity to gather and analyze complementary flood information from new and unconventional sources of data and accelerate flood modeling. This research aims to introduce a vision-based framework using surveillance imageries to enhance the monitoring, modeling, and management of water resources, using Computer Vision and Deep Learning techniques. This vision-based framework interprets visual features of the captured images into water-related numerical parameters, such as water level and inundation area. Such a framework will support flood

modeling by providing real-time input information for data assimilation and inform decision-makers and first responders to undertake appropriate actions and adaptation strategies facing flood risk.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Extreme events, such as droughts, floods, and natural hazards have become more frequent and intensive, due to global warming and climate change, and their associated damages and socio-economic inverse impacts are increasing (Zscheischler et al. 2020). Flood, for example, is among the most devastating natural hazards, and its intensity and frequency are anticipated to occur even more in the future, especially in the coastal regions (Webster 2013; Piecuch et al. 2018). According to (Hirabayashi et al. 2013), reported annual losses caused by floods reached tens of billions of US dollars and thousands of people were killed each year. In order to cope with the adverse consequences of floods, a wide range of structural and emergency actions may be considered, among which developing flood early warning systems and mapping are of high importance.

Early warnings and monitoring systems of flood can protect human lives and prevent further damages by providing timely information needed to plan for evacuation, emergency management, and relief work (Gebrehiwot et al. 2019). Flood inundation models are often developed to simulate and predict affected areas and the degree of damage caused by storm events. These models rely heavily on temporal and spatial hydroclimate data received from ground-based gauges, remote sensing (Lo et al. 2015), and more recently crowdsourcing (Howe 2008). However, all these sources have their own shortages and limitations. For example, gauge sensing is only capable of measuring stream flow data at specific locations, and therefore, provides inadequate information about the spatial distribution of flood (Lo et al. 2015). Remote sensing data is constrained by the limited revisit intervals of satellites, cloud covering, and systematic departures (or biases) (Panteras and Cervone 2018). Finally, the reliability and confidence level of crowdsourcing methods have been criticized due to the lack of verification (Schnebele, Cervone, and Waters 2014; Goodchild 2007).

In addition to the limitations associated with the flood data collection methods, modeling techniques and underlying mathematical equations used to simulate flood

processes and watershed responses are highly complex and uncertain due to the dynamic nature of flood generation and propagation (Mosavi, Ozturk, and Chau 2018). The challenges are imposed by the non-linearity of hydrological processes, temporal and spatial variability of effective parameters, and equifinality (Fatichi et al. 2016). These limitations (i.e., lack of adequate data and limited knowledge about physical processes in hydrology) have compelled researchers of the water resources community to adopt data-driven and Machine Learning (ML) approaches to simulate flood and other hydrological processes. These models are capable of discovering the underlying physical relationships existing in different hydrological parameters while extracting and interpreting these relationships are still a challenge. In addition to solving problems, ML models have shown great capabilities for pre-processing and analyzing big data using decomposition, dimensionality reduction, anomaly detection, and denoising techniques (Quilty and Adamowski 2018). Moreover, recent advances in ML, specifically in Deep Learning (DL), configure model architectures to better deal with (RGB) images (Krizhevsky, Sutskever, and Hinton 2012), which allows analyzing and processing digital images for the purpose of extracting visual contexts and numerical information from those. *Thus, this research aims to introduce images and videos captured by ground-based cameras as a new source of data for measuring hydrological data and monitoring water systems, particularly in the context of floods.*

The proposed framework is able to extract actual field information such as water stage, flood extent, and flow rate (Chow, Maidment, and Larry 1988) with high spatio-temporal resolution from the processed time-lapse images. High-resolution data will be used to support flood models to better estimate flood inundation and water level in a real-time manner at different monitored locations. Developing such a vision-based framework requires access to comprehensive and large image datasets for training and testing DL-based models and analyzing water visual features and textures as visual objects in digital images. Unfortunately, such a dataset and exclusive DL-based model

customized for semantic segmentation of water does not exist. Thus, the first phase of this research focused on developing a new water-related semantic segmentation image dataset and an exclusive DL-based model for semantic segmentation of water and water-related objects from ground-based (RGB) images.

In the following chapter– Literature Review– existing flood data sources and measurement techniques are reviewed, and their advantages and disadvantages are discussed. The flood modeling frameworks and their required parameters are presented, and a technical overview of ML and its application in the water engineering field are described to draw potential pathways toward using DL-based techniques for solving existing challenges in flood detection, monitoring, and modeling research. This chapter is followed by the proposed research plan and presents the research questions, hypotheses, and objectives. The third chapter is devoted to describing the first phase of research which includes the development of a water-related image dataset and DL-based model for semantic segmentation of different waterbodies in digital (RBG) images. The fourth chapter introduces ATeX, a new image dataset for the classification and texture analysis of water in different surrounding environments. Finally, in the fifth chapter, the proposed framework for using surveillance imagery as a data source to extract numerical water data is presented.

Chapter 2

Literature Review

This chapter reviews the different hydrological data sources and flood modeling approaches. First, an overview of the measurement method is explained. This is followed by a discussion of important benefits and drawbacks. Different flood inundation modelings as well as Machine Learning techniques are reviewed afterward. The chapter is concluded with my research plan to introduce surveillance imagery as a new source of flood data using recent advances in Computer Vision and Deep Learning.

2.1 Sources of flood data

2.1.1 In-situ Gauge Sensing

The measurements from in-situ gauges are often used for calibration and simulation purposes of models. hydrological processes vary in space and time, and are random or probabilistic, in character. These uncertainties create a requirement for hydrological measurement to provide observed data at or near the location of interest so that conclusions can be drawn directly from on-site observations (Chow, Maidment, and W. 1964). Stream gauges provide near real-time flooding information (e.g. water height, stream flow) for monitored locations (Turnipseed and Sauer 2010). Although the in-situ gauging stations are considered the most accurate approach for estimating river discharge, they have some inherent problems as follows:

- These stations are expensive to maintain and in many cases, they are not installed systematically along any given waterway (King, Neilson, and Rasmussen 2018)
- Some locations are technically, logistically, and politically difficult to access which leads to dispersed measurements and insufficient information to map the flooded area completely (Li et al. 2018).

- Stream gauges are not useful when the water level rises beyond the limit of ground-based gauges, and stream flow can wash the gauges away during intense floods (Li et al. 2018).

- In-situ gauges are vulnerable to external interferences and they have relatively high operating costs for regular calibration (Gao et al. 2019).

2.1.2 Remote Sensing

The use of remote sensing data to produce inundation mapping (Townsend and Walsh 1998) and to assess flood hazards in near real-time has been popular over the past few decades (Gebrehiwot et al. 2019). Multispectral images have been utilized for extracting the characteristics of hydrological surfaces, including topography, soil saturation status, and delineation of flooding zones (Qiao et al. 2012). However, thus far, studies on the remote measurement of water levels have mainly focused on large-area waterbodies, such as oceans, interior lakes, lagoons, and large rivers (width $> 100m$) (Lo et al. 2015). These studies showed, by integration of remote sensing and topographic data, water stage (Schumann et al. 2009; Alsdorf, Rodriguez, and Lettenmaier 2007), discharge (Bjerklie et al. 2003; Smith and Pavelsky 2008) are retrievable.

Remote altimetry technology can be used to continuously measure the water level variation within a large area and can thus be used to extensively monitor an entire flood event (Alsdorf, Rodriguez, and Lettenmaier 2007). Data from both optical and microwave sensors can be used for this mission. Synthetic Aperture Radar (SAR) systems offer the possibility to operate day and night and they can penetrate clouds and heavy rainfall. This feature is of special importance considering the weather conditions under which flood events usually occur (Schlaffer et al. 2015). Moreover, the number of automatic image processing algorithms to derive flooding from SAR data has increased since the launch of the high-resolution Synthetic Aperture Radar (SAR) satellite systems TerraSAR-X, Radarsat-2, and Cosmo-SkyMed constellation

(CSK) for the timely provision of crisis information during inundation events (Martinis, Kersten, and Twele 2015). However, due to the restriction of orbital cycles and inter-track spacing of satellite movements, the limitation of remote measurement data in continuous monitoring and the observation of fixed points is a key problem. So, it is difficult to use remote sensing technology for long-term and real-time monitoring at a fixed region of interest.

2.1.3 Crowdsourcing

Crowdsourcing is the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call (Howe 2008). With respect to recent investments of municipalities and related authorities in the concept of sensor networks for addressing security and surveillance issues, humans themselves can be considered as a specific type of sensor network. Each person is equipped with five senses and the intelligence to compile and interpret what they sense. This network of human sensors has over a billion components, each an intelligent synthesizer, and interpreter of local information.

Volunteered geographic information (VGI), a variant of crowdsourcing, certainly focuses on geographic information systems (GIS) and more generally on the discipline of geography and its relationship to the public (Goodchild 2007). VGI can be considered an effective use of "user-generated content" for flood studies, enabled by the pervasive use of social media. The proliferation of social media platforms such as Twitter, Facebook, and Instagram has led to the generation of massive amounts of user-generated geospatial data from real-time data streams. The temporal resolution of social media is much higher than that of remote sensing data. In fact, the crowdsourced data via social media can be generated and published almost instantaneously after the occurrence of an incident, which makes it very suitable for situational aware-

ness applications (Yin et al. 2015). Social media and particularly Twitter (Huang, Wang, and Li 2018) tend to be utilized by ordinary people during the occurrence of a disaster and natural hazard, providing real-time information which is very valuable for the disaster management agencies (Palen et al. 2010).

However, the reliability and confidence level associated with VGI have been criticized due to the lack of verification (Schnebele, Cervone, and Waters 2014). The participants are largely untrained, their actions are almost always voluntary, and the results may or may not be accurate (Howe 2008). These systems completely rely on self-motivated citizens who have no obvious incentive to spend time creating the content of VGI. Moreover, there are no elaborate standards and specifications to govern the production of geographic information and the content is asserted by its creator without citation, reference, or other authority. Moreover, such contribution is still largely unavailable to most of the world's population who live in developing countries due to inaccessibility to the required technology, such as the Internet, smartphones, and social media.

The collected data from different sources is fed into flood inundation models for model calibration or prediction purposes. These models require different input data and subsequently provide different output information. The following section outlines different types of flood modelings and their features.

2.2 Flood inundation modeling

Generally, efficient modeling requires paying attention to data availability, desired output variables, and their time and space scales, as well as the level of accuracy, computational resources, and algorithms. The end users should concern about these differences to wisely select a model balancing their demands against model complexity and data requirements (Teng et al. 2017). For flood forecasting, as an example, models must have fast run time for real-time data assimilation (Sharma and Machi-

wal 2021). These models require time series data of river stage and the amount of rainfall occurring on a real-time basis. For flood hazard mapping, on the other hand, maximum flood extent and water depth may be sufficient (Huang, Wang, and Li 2018). Several studies have been conducted for flood inundation mapping, which could roughly break into three categories:

2.2.1 Hydrodynamic models

These models are widely used to simulate detailed flood dynamics. They can be directly linked to hydrological models and river models to provide flood risk mapping, flood forecasting, and scenario analysis (Teng et al. 2017). Depending on their spatial representation of the floodplain flow, the models can be dimensionally grouped into 1D, 2D, and 3D models. 2D hydrodynamic models are perhaps the most widely used models in flood extent mapping and flood risk estimation studies. These models can solve full shallow water equations and they are able to simulate the timing and duration of inundation with high accuracy. Nevertheless, they are computationally intensive. Generally, hydrodynamic models have no analytical solution, but can be solved using numerical techniques. These methods need to be satisfied by flood information (e.g., water height and stream flow).

2.2.2 Simplified (non-physics-based) methods

These models can be labeled as simplified conceptual models and are based on more modest representations of physical processes and have run times orders of magnitudes shorter than hydrodynamic models, making them useful tools for large-scale applications where only final (maximum) flood extent and water levels are required (Teng et al. 2017). One of these models is the Rapid Flood Spreading Method (RFSM) (Lhomme et al. 2008) which divides the floodplain into elementary areas that represent topographic depressions in the preprocessing. It then spreads the

flood volumes by filling these areas using a filling/spilling process. Another simplified conceptual model is based on the so-called theory of "planar method" or "bathtub method". It derives the flood extent by intersecting a series of planes at fine intervals with a high-resolution DEM and instantly links the water stage/volume with the flood extent (Teng et al. 2015).

As conceptual models can be run just by one input, i.e., water height, in addition to in-situ gauges, various types of tools can be used to provide water height to obtain flood mapping. For example, spatio-temporal data retrieved by VGI can provide the required information for these models to create hazard maps (Schnebele and Cervone 2013; Cervone et al. 2016; Cervone et al. 2017; Li et al. 2018; Huang, Wang, and Li 2018).

2.2.3 Remote Sensing

Remote sensing techniques are the alternatives for efficient post-event flood mapping using multispectral (optical) imagery, radar data, and digital elevation models (DEMs). These data allow for delineating flood extent over large areas in near real-time depending on the satellite's spatio-temporal resolution (Munoz et al. 2021). Many image-processing techniques exist to successfully derive flood areas and extent. Recently, some studies integrated multi-source satellite-based data, such as backscattering radar data with either multispectral imagery or DEMs, for land cover classification tasks using different techniques (Xu et al. 2019; Feng et al. 2019; Muñoz et al. 2021). These methods have been successfully applied for flood mapping in riverine and coastal floodplains, too (Munoz et al. 2021).

2.2.4 Machine Learning

ML techniques have been commonly used among water resources research communities over the past decades for different purposes, such as time series analysis, re-

gression, and classification (Maier and Dandy 2000). There are several studies that comprehensively reviewed the application of ML models in water resources, hydrology, and flood prediction (Iqbal et al. 2021; Sit et al. 2020; Mosavi, Ozturk, and Chau 2018; Nourani et al. 2014; Shen 2018; Razavi 2021). This implies that the application and development of ML techniques to water-related topics can be considered as new mainstream of earth and environmental sciences.

Stream flow forecasting was traditionally done by using physical-based approaches to model the behavior of the water systems (Paniconi and Putti 2015; Fatichi et al. 2016). These models often need a variety of calibration data that are not accessible or can be computationally expensive (Razavi 2021). Data-driven methods have long been used as an alternative for rainfall-runoff modelling (Hsu et al. 2002; Kişi 2007; Rasouli, Hsieh, and Cannon 2012; Erdal and Karakurt 2013; Sun, Wang, and Xu 2014; Shortridge, Guikema, and Zaitchik 2016; Hosseini and Mahjouri 2016; Lima, Hsieh, and Cannon 2017; Adnan et al. 2019; Cheng et al. 2020; Xiang, Yan, and Demir 2020). Among the most used ML methods in hydrological modeling are neural networks (Chiang, Chang, and Chang 2004; Cigizoglu 2005; Mutlu et al. 2008; Anusree and Varghese 2016) Support Vector Machines (SVM) (Vapnik 1999; Collobert and Bengio 2001; Dibike et al. 2001; Wu, Chau, and Li 2009; Tikhamarine, Souag-Gamane, and Kisi 2019) and regression trees (Senthil Kumar et al. 2013; Charoenporn 2017). More recently, DL-based models including Long Short-Term Memory (LSTM) network (Kratzert et al. 2018; Kratzert et al. 2019) and Temporal Convolutional Networks (TCN) (Lea et al. 2017; Bai, Kolter, and Koltun 2018), have been applied for rainfall-runoff modeling (Yan et al. 2020; Duan, Ullrich, and Shu 2020).

Accurate urban land-use mapping is a challenging task in remote sensing data analysis. ML models can play a critical role in solving spatial problems using prediction, classification, and clustering models. These models can be applied to both types of satellite data, optical and radar images. For example, ML classifiers such as

SVM, Naive Bayes (NB), classification and regression tree (CART), K-nearest neighbor (KNN), and random forest classifier have been used for per-pixel and object-based classification on multispectral satellite images (Huang, Davis, and Townshend 2002; Pal 2005; Adam et al. 2014; Qian et al. 2014). Recently, with more experience and improved methodology, there are some successes in using ML models for land-use classification and flood detection using Synthetic Aperture Radar (SAR) satellite imagery (Tanim et al. 2022). Moreover, fusing multifrequency SAR and optical multispectral data for land cover applications such as crop classification and waterbody mapping have been done using both ML models (Garg et al. 2022) and DL models (Muñoz et al. 2021). Some of these studies designed model architecture customized for the classification of multisource RS images (Feng et al. 2019; Kussul et al. 2017).

2.3 Research Plan

As it is mentioned, flood models commonly need some initial inputs, and boundary conditions to perform. This information is normally measured by in-situ gauging, remote sensing, and more recently crowdsourcing. In this section, we aim to introduce a vision-based framework as a new source for measuring hydrological data. This framework uses time-lapse images captured by surveillance cameras for estimating the characteristics of stream flow.

Vision-based analysis of waterbodies can provide important insights for monitoring, analyzing, and managing water resource systems (e.g., visual flood detection). Vision-based analysis can complement conventional ground-based gauging and remote sensing systems to address their existing shortages for measuring flood data, such as water stage, and flow rate, needed for modeling.

Developing such a vision-based framework requires detecting water in digital images. Water, however, is a challenging object for image processing. Inherently, water

is shapeless and transparent, but it appears in different forms, textures, and colors depending on the surrounding environment. So, detection, classification, and tracking of water in images and videos are difficult tasks. There are still some visual differences resulting from the texture and color of water in different waterbodies which provide potential information needed for DL-based models to better analyze water images. Thus, developing a vision-based framework for measuring physical water parameters from videos and images during flood events, requires answering three underlying and interrelated research questions:

1. Considering all visual challenges associated with water and water-related objects, can DL-based models detect and classify water types and waterbodies in ground-based digital images?

2. What are the visual features of water in digital RGB images that can better represent different water types and waterbodies?

3. By having DL-based models capable of detecting, classifying, and segmenting different water types in digital images, does surveillance imagery provide enough information for estimating characteristics of stream flows such as water level and discharge?

As DL-based approaches always require large-scale training data with necessary ground-truth annotations, and the lack of such a public dataset for waterbody segmentation significantly impedes the research on this problem, I hypothesize that *i) Developing a new image dataset for semantic segmentation of waterbodies and water-related objects can enable applied research studies on water and water-related issues and determine whether the DL-based models can analyze and process water and waterbodies.*

The recognition strategy of CNN-based approaches is to follow "local" to "global" features in various stages of the visual pathways. It means CNN models recognize

objects through the analysis of texture and shape-based clues– local and global representations and their relationship in the entire field of view. So, I hypothesize that *ii) Developing a classification image dataset emphasizing color and texture analysis of water in digital (RGB) images facilitates discovering representative visual features and properties of different water types and waterbodies.* Such information is necessary for developing DL-based models customized for water and water resources applications.

I also hypothesize *iii) estimating the geometric properties of a surveillance camera enables the potential which can be exploited by Computer Vision techniques for estimating flow characteristics.*

2.4 Research objectives and methods

The main objectives of this proposal are, i) developing a dataset for semantic segmentation of natural and artificial waterbodies and related objects. ii) developing a dataset for the task of classification, as well as texture, and color analysis of water in digital images and iii) developing a vision-based framework for measuring characteristics of stream flow using computer vision and deep learning techniques.

2.4.1 Developing a dataset for semantic segmentation

Large-scale annotated datasets, such as COCO (Lin et al. 2014), PASCAL Context (Mottaghi et al. 2014), ADE20K (Zhou et al. 2019), Mapillary Dataset (Neuhold et al. 2017) and BDD100K (Yu et al. 2020), make it possible for researchers to develop DL-based models for real-world applications. Considering the most related dataset to water resources problems, (Gebrehiwot et al. 2019) collected a small number of top-view waterbody datasets (100 images) using Unmanned Aerial Vehicles (UAVs) which contains only four categories (i.e., water, building, vegetation, and road). In addition, (Sazara, Cetin, and Iftekharuddin 2019) introduced a larger dataset (253 images) which just focuses on flood region segmentation. Sarp et al. 2020 provided a

dataset that consists of 441 annotated roadway flood images. However, these datasets have limitations in either the number of annotated images, or the categories they covered, and none of those considers more complex classes of waterbodies such as sea, lake, river, reservoir, and wetland. Therefore, we will develop a new dataset as the first large-scale annotated dataset to provide a wide range of waterbodies and water-related objects.

This dataset is designed and developed with the goal of including different water types, either those that exist in the natural environment or in artificial water systems. So, labels are based on the most frequent objects used in water-related studies or can be found in real-world scenes. General labels (e.g., human, car, vegetation) are also considered for providing contextual information related to different waterboides. In order to gather a corpus of images, we use Flickr API to query and collect "medium-sized" unique images for each label based on "Creative-Commons" licenses. Downloaded images are sieved in accordance with the opinion of experts in water resources. Finally, images are annotated by annotators who have a decent background in water resources engineering as well as experience working with the CVAT, which is a free, open-source, and web-based image annotation tool.

2.4.2 Developing a classification dataset for water

Considering different water features and their combinations, water can appear in completely different forms and colors. The classification dataset is designed and developed with the goal of representing various textures and colors in which water usually appears in different waterbodies. Images are derived from the semantic segmentation dataset. In computer science, texture analysis has been widely applied for different purposes such as facial recognition and expression analysis (Liu et al. 2014). Facial expression analysis refers to developing models for automatically analyzing and recognizing facial motions and feature changes from visual information (Tian,

Kanade, and Cohn 2005). These features can be extracted either by hand-designed filters (Zhang et al. 1998; Zhao and Pietikainen 2007) or trained ML models (Meng et al. 2017). So far, many researchers have attempted to improve the feature extraction techniques to provide better facial recognition (Tong, Liao, and Ji 2007). There are still challenging factors in the facial recognition task associated with expression, illumination changes, and aging. These dynamic features changing from one face to another, arguably make this type of problem similar to waterbody classification. In both cases, developed models must differentiate the same objects in essence from each other using different subtleties. In the case of water, however, the lack of water datasets covering wide ranges of different waterbodies prevents researchers from performing the texture analysis on water.

2.4.3 Vision-based surveillance setup

The core of this task is to develop a vision-based framework using surveillance imageries for measuring water levels in the community's local stream or river, using computer vision and deep learning techniques. This task can be divided into the following sub-tasks:

Data Acquisition

In order to create a 3D point cloud of the study area iPhone LiDAR scanner is used. The ground control points (GCPs) are also needed to define a local coordinate system. The GCPs are measured with a total station, and both 3D point cloud and camera coordinates will be transformed into the GCPs coordinate system. At least, eight permanent GCPs will be set up and measured with a total station. The permanent GCPs are ArUco markers laminated with waterproof pouches. These GCPs will be used to determine the exterior calibration parameters– camera's position and orientation– for a series of images that are captured during each period of observation.

Two different single-board computers (SBC) are used in this study, Raspberry Pi for capturing time-lapse images of a river scene and Arduino for measuring water level as the ground truth data. These devices are designed to communicate with each other. During capturing time-lapse images, the Pi camera device triggers the ultrasonic sensor for measuring the corresponding water level.

Semantic segmentation of water

The water extent can be automatically determined on the 2D image plane with the help of DL-based models. The task of semantic segmentation is applied in this step to delineate the water line on the left and right banks of the river. Different DL-based approaches will be trained and tested.

Projective Geometry

Computer vision techniques are used for different purposes, in this step. First, CV models are used for camera calibration. The process of estimating the parameters of a camera is called camera calibration. They include focal length, optical center, radial distortion, camera rotation, and translation. The interior camera parameters are estimated by Checkerboard (Zhang 2000). The exterior calibration parameters will be estimated via "spatial resection" using the non-linear Levenberg-Marquardt minimization scheme (Madsen, Nielsen, and Tingleff 2004).

Machine Learning for Image Measurements

Using the projection matrix, the 3D point cloud is projected on the 2D image plane. The projected (2D) point cloud is intersected with the water line pixels, the output of the DL-based model (Module 1), to find the nearest point cloud coordinate. For this purpose, K-Nearest Neighbors (KNN) algorithm is used. The indices of the selected points are the same for both the 3D point cloud and the projected (2D)

correspondences. So, using the indices of the selected projected (2D) points, the corresponding real-world 3D coordinates are retrievable.

2.5 Overview of Next Chapter

According to the research questions and hypotheses mentioned in this section, the following chapter introduces ATLANTIS, a benchmark for semantic segmentation of waterbody images.

Chapter 3

ATLANTIS: A Benchmark for Semantic Segmentation of Waterbody Images [1]

[1] Erfani, S.M.H., Wu, Z., Wu, X., Wang, S. and Goharian, E., 2022. Environmental Modelling & Software, 149, p.105333. Reprinted here with permission of the publisher.

Every year, floods claim tens of billions of US dollars losses and thousands of lives globally (Hirabayashi et al. 2013). Accurate detection, measurement, and tracking of the waterbodies can help both the public and decision-makers to take appropriate actions to minimize the risk and losses (Huang, Wang, and Li 2018; Gebrehiwot et al. 2019). With the popularity of smartphones and airborne imagery, various data at flooding sites can be collected rapidly and continuously to provide more useful and heterogeneous information source (Eltner et al. 2021; Hosseiny 2021; Eltner et al. 2018; Cervone et al. 2016; Schnebele and Cervone 2013), compared to the conventional gauge sensing and remote sensing (Lo et al. 2015). As a fundamental step to leverage such collected images for modeling and decision-making, we need first to conduct a refined semantic segmentation of included waterbodies and related objects in such scenes, which we focus on in this paper.

With the advancement of deep neural networks, semantic segmentation has achieved great success in recent years on various kinds of images, such as natural images (Lin et al. 2014), street images (Cordts et al. 2016; Yu et al. 2020), and medical images (Bernal et al. 2017; Jha et al. 2020). These successes have motivated the application of deep learning across a wide range of disciplines (Razavi 2021). However, waterbody images pose many new unique challenges for semantic segmentation. In some forms, water preserves intrinsic properties such as reflection, transparency, shapeless and colorless visual features; which in turn, brings difficulties to the semantic segmentation of water and related objects. Moreover, in some other forms, these properties can be affected by illumination sources from surroundings, turbidity, and turbulence. Different-labeled waterbodies, such as river and canal, or lake and reservoir, often have similar visual characteristics that make the task of semantic segmentation even harder. As shown in our later experiments, these unique challenges may significantly affect the performance of the existing semantic segmentation networks.

Meanwhile, deep-learning-based approaches always require large-scale training data with necessary ground-truth annotations. The lack of such a public dataset for waterbody segmentation significantly impedes the research on this problem. The collection and annotation of such a dataset can be very laborious and time-consuming to cover a wide range of waterbodies and related objects. There is no specific repository providing relevant images. In addition, team members and annotators are required to have prior knowledge of water resources engineering to be capable of selecting and precisely annotating the images.

In this paper, we present a new benchmark, ATLANTIS (ArTificiaL And Natural waTer-bodIes dataSet). For the first time, this dataset has covered a wide range of natural and man-made (artificial) waterbodies such as sea, lake, river, canal, reservoir, and dam. ATLANTIS includes 5,195 pixel-wise annotated images split into 3,364 training, 535 validation, and 1,296 testing images. As shown in Table 3.1, in addition to 35 waterbody and water-related objects, ATLANTIS also covers 21 general labels. Moreover, we construct ATLANTIS Texture (ATeX) dataset, which consists of 12,503 patches for the water-bodies texture classification, sampled from 15 kinds of waterbodies in ATLANTIS.

Table 3.1 List of the ATLANTIS labels.

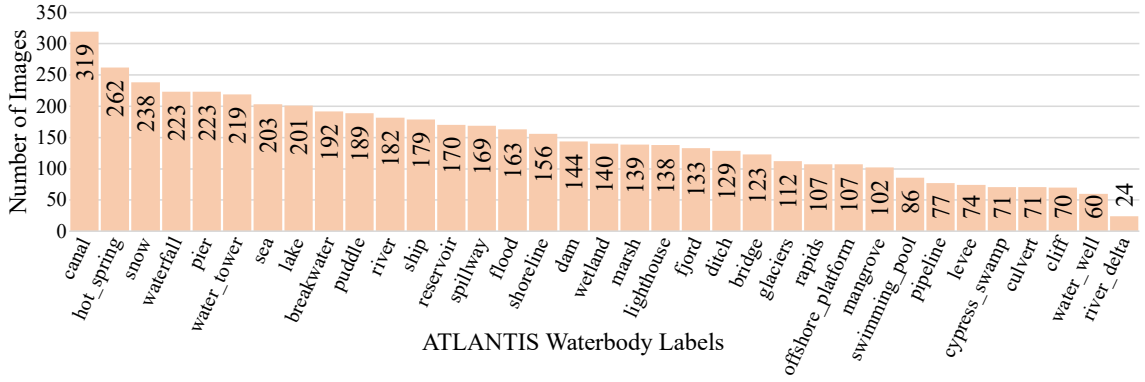
Artificial	breakwater; bridge; canal; culvert; dam; ditch; levee; lighthouse; pipeline; pier; offshore platform; reservoir; ship; spillway; swimming pool; water tower; water well.
Natural	cliff; cypress tree; fjord; flood; glaciers; hot spring; lake; mangrove; marsh; puddle; rapids; river; river delta; sea; shoreline; snow; waterfall; wetland.
General	road; sidewalk; building; wall; fence; pole; traffic sign; vegetation; terrain; sky; train; person; car; bus; truck; bicycle; parking meter; motorcycle; fire hydrant; boat; umbrella.

In order to tackle the inherent challenges in the segmentation of waterbodies, AQUANet is developed which takes an advantage of two different paths to process

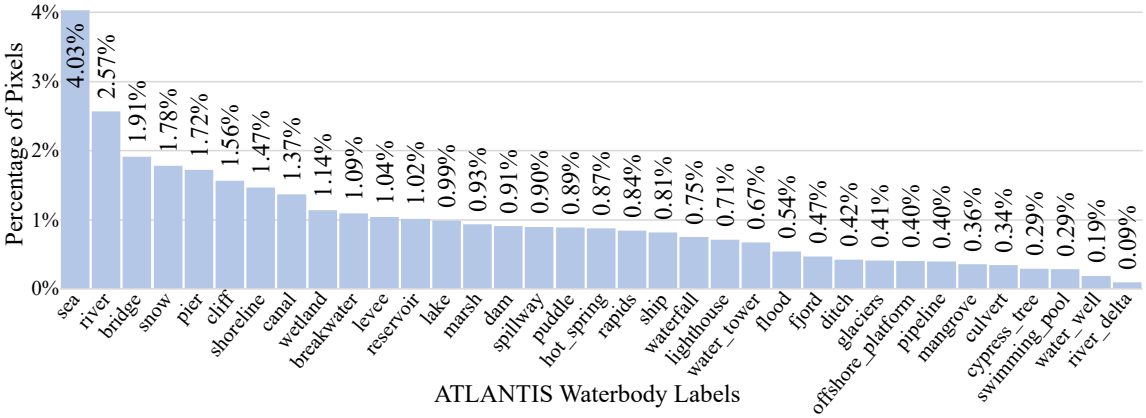
the aquatic and non-aquatic regions, separately. Each path includes low-level features and cross-path modulation, to adjust features for better representation. The results show that the proposed AQUANet outperforms the other ten state-of-the-art semantic-segmentation networks on ATLANTIS, and the ablation studies justify the effectiveness of the components of the proposed AQUANet.

3.1 Related Work

All existing semantic segmentation approaches share the same goal to classify each pixel of a given image but differ in the network design, including low-resolution representations learning (Long, Shelhamer, and Darrell 2015; Chen et al. 2017a), high-resolution representations recovering (Badrinarayanan, Handa, and Cipolla 2015; Noh, Hong, and Han 2015; Lin et al. 2017), contextual aggregation schemes (Yuan and Wang 2018; Zhao et al. 2017; Yuan, Chen, and Wang 2020), feature fusion and refinement strategy (Lin et al. 2017; Huang et al. 2019; Li et al. 2019; Zhu et al. 2019; Fu et al. 2019). Typically, method designs are dependent on their respective datasets and all the mentioned networks are developed by training on benchmark datasets such as Cityscapes (Cordts et al. 2016), COCO (Lin et al. 2014) and VOC (Everingham et al. 2010) where the inter-class boundary is clear even for the within-group categories (e.g., car and truck). As mentioned above, waterbody images pose new challenges to semantic segmentation. Previous works on waterbody segmentation mainly use satellite imagery (Munoz et al. 2021). In this work, we focus on natural waterbody images terrestrially captured by various cameras and design AQUANet, a new two-path semantic segmentation network, by including an aquatic branch explicitly for waterbody classes.



(a)



(b)

Figure 3.1 (a) The frequency distribution of images for different waterbodies in ATLANTIS (b) The Percentage of pixels for waterbody labels.

3.2 ATLANTIS Dataset

The ATLANTIS dataset is designed and developed with the goal of capturing a wide range of water-related objects, either those that exist in the natural environment or the infrastructure and man-made (artificial) water systems. In this dataset, labels were first selected based on the most frequent objects, used in water-related studies or can be found in real-world scenes. Aside from the background objects, a total of 56 labels, including 17 artificial, 18 natural waterbodies, and 21 general labels, are selected (Table 3.1). These general labels are considered for providing contextual information that most likely can be found in water-related scenes. After finalizing the selection of waterbody labels, a comprehensive investigation on each individual

label was performed by annotators to make sure all the labels are vivid examples of those objects in real-world. Moreover, sometimes some of the water-related labels, e.g., levee, embankment, and floodbank, have been used interchangeably in the water resources field; thus, those labels are either merged into a unique group or are removed from the dataset to prevent an individual object receives different labels.

In order to gather a corpus of images, we have used Flickr API to query and to collect "medium-sized" unique images for each label based on eight commonly used "Creative-Commons", "No Known Copyright Restrictions" and "United States Government Work" licenses. Downloaded images were then filtered by a two-stage hierarchical procedure. In the first stage, each annotator was assigned to review a specific list of labels and remove irrelevant images based on that specific list of labels. In the second stage, several meetings were held between the entire annotation team and the project coordinator to finalize the images which appropriately represent each of 56 labels.

This sieving procedure has been applied four times in order to meet the limit and reach the current number of images. The percentage of image acceptance rate for the third and fourth phases are 14.41% and 5.06%, respectively. It means if we want to add 1000 more images to the dataset, we should process at least 20,000 images. Finally, images were annotated by annotators who have solid water resources engineering background as well as experience working with the CVAT (Sekachev et al. 2020), which is a free, open-source, and web-based image/video annotation tool.

3.2.1 Dataset statistics

Figure 3.1 shows the frequency distribution of the number of images and the percentage of pixels for waterbody labels. Such a long-tailed distribution is common for semantic segmentation datasets (Lin et al. 2014; Zhou et al. 2019) even if the number of images that contain specific labels is pre-controlled. Such frequency distribution

for pixels would be inevitable for objects existing in the real-world. Taking "water tower" as an example, despite having 219 images, the percentage of pixels is less than many other labels in the dataset. Figure 3.2 shows the positive but weak correlation between the number of images for each label and the corresponding pixels. In total, only 4.89% of pixels are unlabeled, and 34.17% and 60.94% of pixels belong to waterbodies (natural and man-made) and general labels, respectively. As it is evident, the main proportion of pixels belongs to general labels. This clearly shows the importance of general labels for better scene understanding (Caesar, Uijlings, and Ferrari 2018) and accurate object classification in a semantic segmentation network.

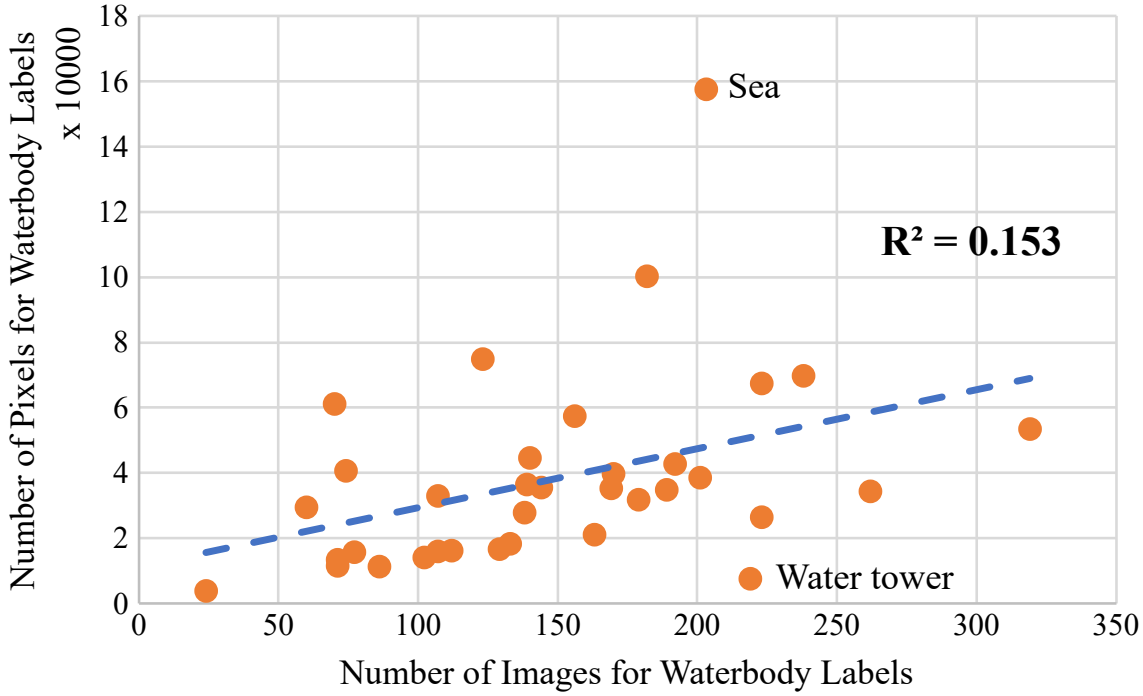


Figure 3.2 Correlation between a number of images for each label and the corresponding pixels.

Spatial analysis

Following ADE20K dataset (Zhou et al. 2019), a spatial analysis, known as "mode of the object segmentation" has been done on the ground truth segmentation map for each label. Specifically, considering a waterbody label L with n images in ATLANTIS,

we resize their corresponding n ground-truth segmentation maps to 512×512 pixels. We then count the most frequent label at each pixel of the map and construct a "mode of segmentation" map for label L, as shown in Figure 3.3. This map demonstrates the spatial distributions of the most frequent co-occurred labels with respect to a given waterbody label. Based on this, we equipped our proposed network with cross-path feature modulation to cope with the difficulties associated with the recognition of waterbody labels having visual similarities.

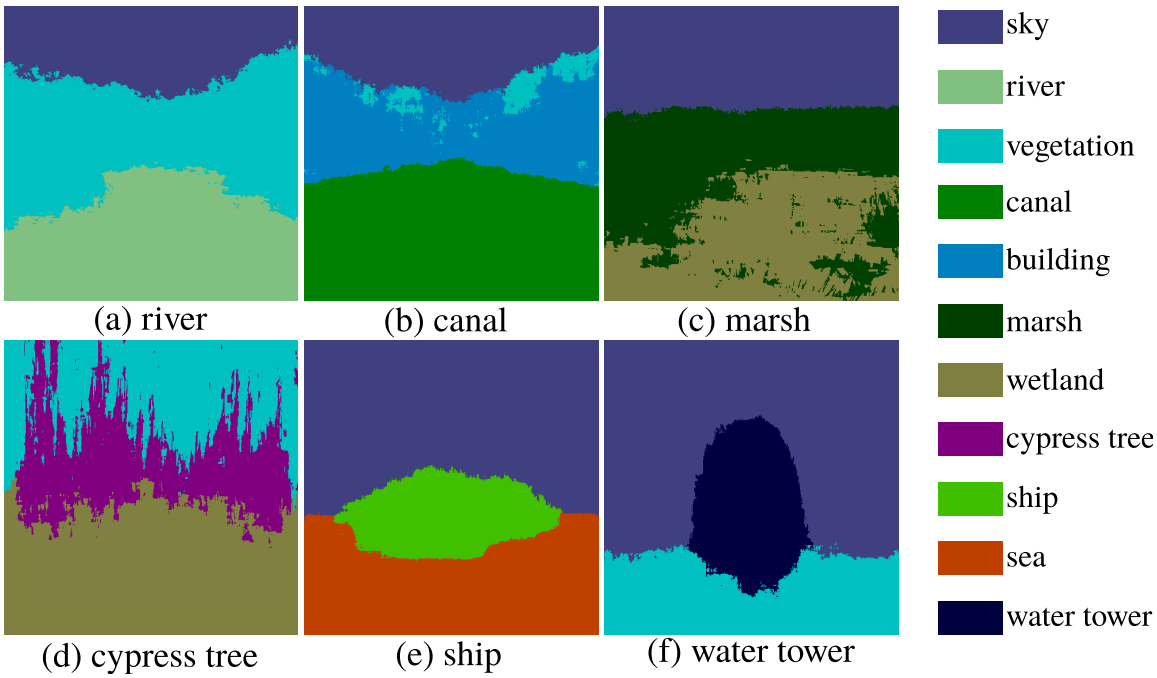


Figure 3.3 Spatial analysis of four different waterbody labels.

3.2.2 Image annotation

Annotation pipeline

ATLANTIS was annotated by six annotators having prior knowledge in the area of water resources. The goal of the annotation task is to balance speed and quality. Generally, time spent on a single image can range from 4 minutes to 25 minutes depending on the image's complexity. In this project, each kind of waterbody was assigned to a specific annotator. Before the annotation of a label, all the images

of that label are scrutinized and discussed by a group of experts in water resources engineering. We can see that the annotation of complex flood scenes takes more time since such images are usually captured in urban areas and have more elaborated components as shown in Figure 3.4.

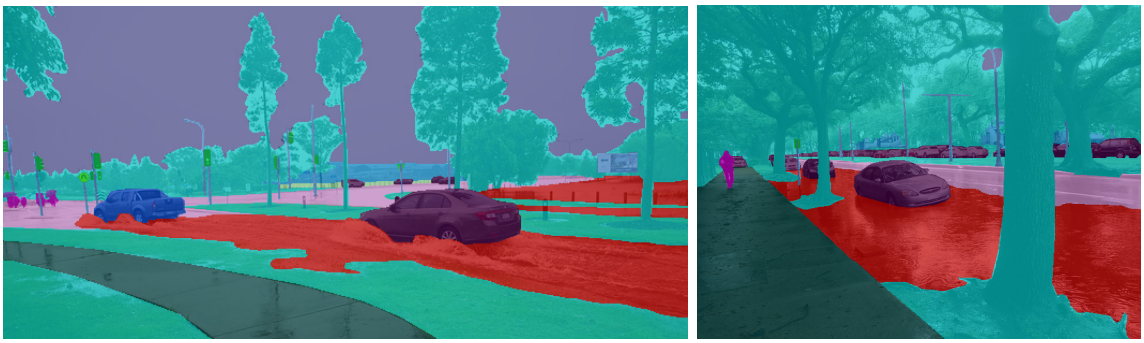


Figure 3.4 Two samples of complex flood scenes

Consistency analysis

While one image is annotated by one annotator for ATLANTIS, we perform additional consistency analysis across annotators and over time for an annotator. We choose 52 images from ATLANTIS, by including both images that are highly susceptible to wrong labeling and those contain objects prone to be either left unannotated or wrongly annotated. We ask three annotators to annotate them again and compare the results against the already approved ground truth in ATLANTIS. The accuracy and mIoU in terms of all 52 images (total) and the subsets of images that had been annotated by themselves before (individual) are shown in Table 3.2. We can see that an annotator can process the images that he/she annotated before with much better consistency.

3.2.3 ATLANTIS Texture (ATex)

Waterbodies usually bear texture appearance and it is an interesting problem to study whether different kinds of waterbodies may show subtle differences associated with

Table 3.2 Consistency analysis for three annotators.

		Annotator 1	Annotator 2	Annotator 3
Total	acc	84.00	79.93	79.00
	mIoU	62.29	59.33	57.34
Individual	acc	91.11	90.44	94.69
	mIoU	72.83	76.14	75.69

the texture features. For this purpose, we construct a new waterbody texture dataset, ATeX, by cropping patches from ATLANTIS and taking the corresponding annotated waterbody label as the label of the patch. We set patch size to 32×32 pixels and all pixels in a cropped patch must have the same waterbody label in the original image. We also ensure there is no partial overlap between any two patches. In total we collected 12,503 patches with 15 waterbody labels: Two waterbody labels "estuary" and "swamp" are added based on the nearby tree species– mangrove "estuary" and cypress for "swamp", while four waterbody labels "canal", "ditch", "reservoir" and "fjord" are omitted because of high visual similarities with other labels. Sample images of ATeX dataset are shown in Figure 3.5. We split ATeX into 8,753 for training, 1,252 for validation and 2,498 for testing. In the later experiment, we train different models to evaluate their classification performance on ATeX images.

3.3 AQUANet

Typically, existing semantic segmentation networks are designed based on a strong backbone (e.g. ResNet (He et al. 2016)) to extract features from images with additional feature aggregation schemes such as ASP-OC (Yuan and Wang 2018) and PPM (Zhao et al. 2017). Because of difficulties associated with the semantic segmentation of waterbodies, we design AQUANet to segment aquatic and non-aquatic categories, separately, as shown in Figure 3.6.

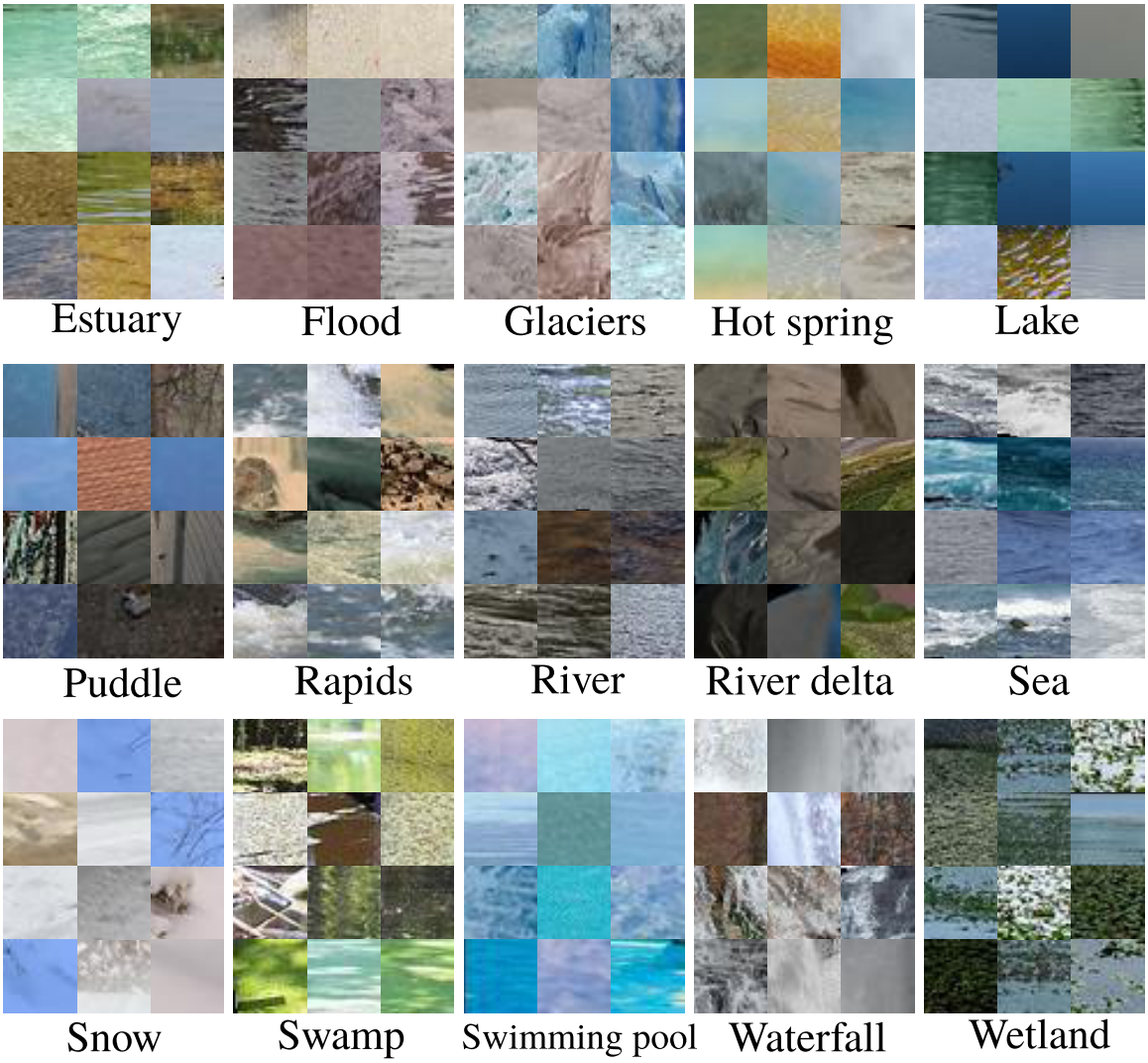


Figure 3.5 Samples of ATeX texture images.

3.3.1 Network architecture

According to Figure 3.6, the input image is first fed into a ResNet-101 (pre-trained on ImageNet (Deng et al. 2009)) to extract the feature F with a size of $C \times H \times W$. Then, the feature is sent into two separate paths for further processing. The aquatic path is to segment different types of waterbodies including sea, river, waterfall, wetland and etc., while the non-aquatic path is to segment other categories such as ship and bridge. In each path, the feature F is first modulated by the low-level feature F_l extracted from the third convolutional layer of ResNet-101, and then passed into

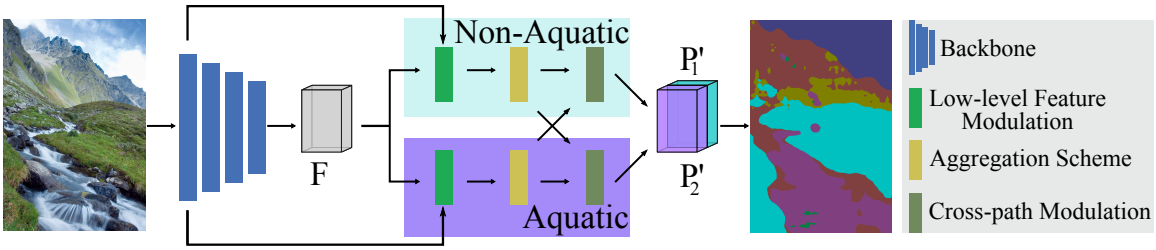


Figure 3.6 The network architecture of proposed AQUANet.

ASP-OC (Yuan and Wang 2018) to produce the probability map. In the last step, two cross-path modulation blocks are applied to adjust the probability maps P_1 and P_2 in parallel. Finally, the resulting probability maps are concatenated and upsampled to the size of the original image.

3.3.2 Feature modulation

The goal of the feature modulation $\mathcal{M}$ is to adjust a feature map F_1 given feature map F_2 to represent the adjusted feature F_1'. It can be formulated as:

$$F_1' = \mathcal{M}(F_1|F_2). \tag{3.1}$$

To generate the modulated feature F_1', the parameters α and β are learned from F_2 via the feature modulation that consists of three downsampling layers, six 1×1 convolutional layers and two leakyReLU layers as shown in Figure 3.7. The learned parameters α and β have the shape as the F_1. Then, the resulting feature F_1' is constructed as follows according to (Wang et al. 2018; Park et al. 2019):

$$F_1' = \alpha \cdot F_1 + \beta + F_1. \tag{3.2}$$

Low-level feature modulation

To enhance the low-level texture representation of the waterbodies, we propose to use the low-level feature F_l to modulate the feature F and the resulting feature F' is defined as:

$$F' = \mathcal{M}(F|F_l). \tag{3.3}$$

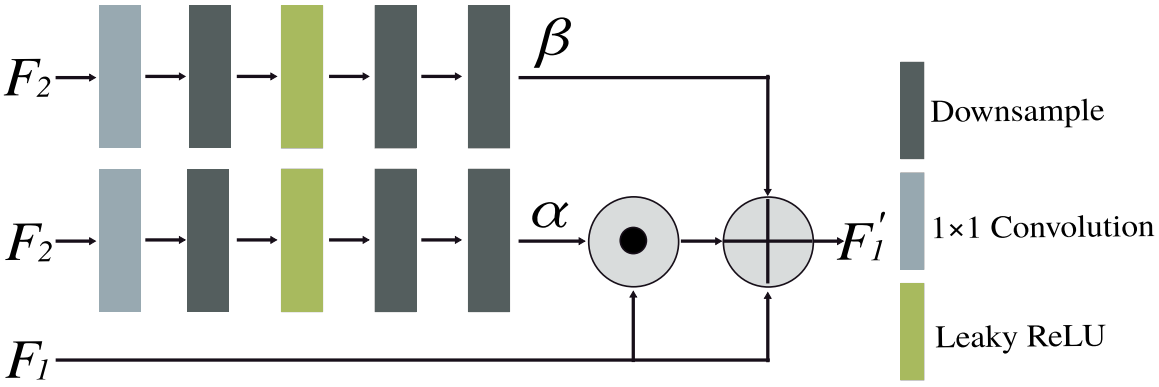


Figure 3.7 An illustration of the feature modulation.

Note that different channels of the receptive field in the third convolutional layer of ResNet-101 are used to construct the low-level feature F_l for the two paths.

Cross-path modulation

We also propose a cross-path modulation to aggregate the outputs of the probability maps. The adjusted probability maps resulting from this module are defined as:

$$P_1' = \mathcal{M}(P_1|P_2), \tag{3.4}$$

$$P_2' = \mathcal{M}(P_2|P_1), \tag{3.5}$$

where the P_1' and P_2' represent the final probability maps for the aquatic and non-aquatic objects, respectively. Note that the modulation layer is not sharing weights across the two paths.

3.3.3 Loss function

The proposed network is trained in a fully supervised fashion constrained by the widely-used cross-entropy loss function on both final prediction P (main loss) and the intermediate feature produced by the fourth block of the ResNet-101 (auxiliary loss). Following (Zhao et al. 2017), the weights of the main and the auxiliary loss are set to 1 and 0.4, respectively.

3.4 Experiments

To demonstrate the effectiveness of the proposed method for water-bodies semantic segmentation, we train AQUANet on the proposed ATLANTIS dataset. For performance evaluation, we take the mean of class-wise intersection over union (mIoU) and the per-pixel accuracy (acc) as the main evaluation metrics. To further evaluate the performance of the waterbodies, we calculate the mean IoU for aquatic categories (A-mIoU) and the accuracy in the aquatic region (A-acc). Aquatic categories include 17 labels showing just water content in different forms and bodies, e.g., sea, river, lake, etc.

3.4.1 Experimental settings

The AQUANet is implemented using PyTorch. During training, the base learning rate is set to 2.5×10^{-4} and it is decayed following the poly policy (Zhao et al. 2017). The network is optimized using SGD with a momentum of 0.9 and weight decay of 0.0001. In total, we train the network for 30 epochs, around 80K iterations with a batch size of 2. The training data are augmented with random horizontal flipping, random scaling ranging from 0.5 to 2.0, and random cropping with the size of 640×640.

Table 3.3 The per-category results on the ATLANTIS test set by current state-of-the-art methods and our AQUANet. The best and the second best results are highlighted with **bold** font and underline, respectively.

Method	canal	ditch	fjord	flood	glaciers	hot spring	lake	puddle	rapids	reservoir	river	river delta	sea	snow	swimming pool	waterfall	wetland	**A-acc** (%)	**A-mIoU**	**acc** (%)	**mIoU**
PSPNet	53.8	29.0	42.9	46.5	57.2	53.9	29.7	**54.7**	38.2	29.8	28.8	65.5	63.5	49.9	47.7	48.4	47.5	66.19	46.29	72.72	40.85
DeepLabv3	52.5	27.2	52.3	43.8	58.7	42.5	31.1	54.2	46.0	32.4	27.1	51.1	61.5	46.3	**53.6**	52.8	52.9	65.83	46.25	69.21	36.23
DANet	50.5	**34.1**	37.1	37.0	51.0	61.6	23.8	51.5	42.8	30.2	31.5	63.5	60.4	50.8	43.1	55.2	54.6	62.00	45.80	74.12	39.60
CCNet	41.1	17.4	35.2	26.9	43.7	47.9	18.6	43.8	29.9	16.6	23.7	48.3	53.3	47.6	38.4	51.1	34.1	51.98	36.33	70.84	36.11
EMANet	46.1	16.6	27.1	23.0	53.8	**63.7**	17.2	43.6	42.2	17.2	21.0	**68.6**	53.5	47.3	36.1	52.1	36.2	55.88	39.13	71.93	36.43
ANNet	50.9	22.8	31.6	32.0	53.1	58.1	25.6	52.9	**48.4**	20.8	28.6	56.8	60.4	**51.1**	43.9	**57.9**	51.4	61.51	43.90	74.06	39.79
GCNet	**56.6**	19.0	44.7	34.8	46.9	36.1	**35.8**	39.4	39.9	**41.6**	32.4	67.0	62.2	46.4	42.9	50.7	59.7	**69.89**	44.48	68.64	37.73
DNLNet	54.4	26.3	48.8	36.3	**63.2**	55.3	35.5	52.3	40.4	32.1	31.3	37.1	61.7	48.3	52.4	48.7	54.6	67.72	45.80	71.95	39.97
OCNet	56.4	33.6	48.0	37.3	57.7	55.2	29.2	50.6	43.8	35.1	**35.6**	65.9	62.7	47.2	47.9	53.1	54.9	67.97	47.89	73.54	41.19
OCRNet	52.4	19.4	46.9	34.9	48.3	58.8	30.4	39.7	42.5	29.8	31.9	55.5	55.4	47.3	43.6	56.8	51.5	65.90	43.83	71.66	36.17
Ours	55.0	27.7	**53.4**	**47.0**	63.1	60.5	33.2	54.4	46.3	39.0	34.7	63.2	**64.2**	50.3	44.9	53.0	**66.1**	68.63	**50.34**	**75.18**	**42.22**

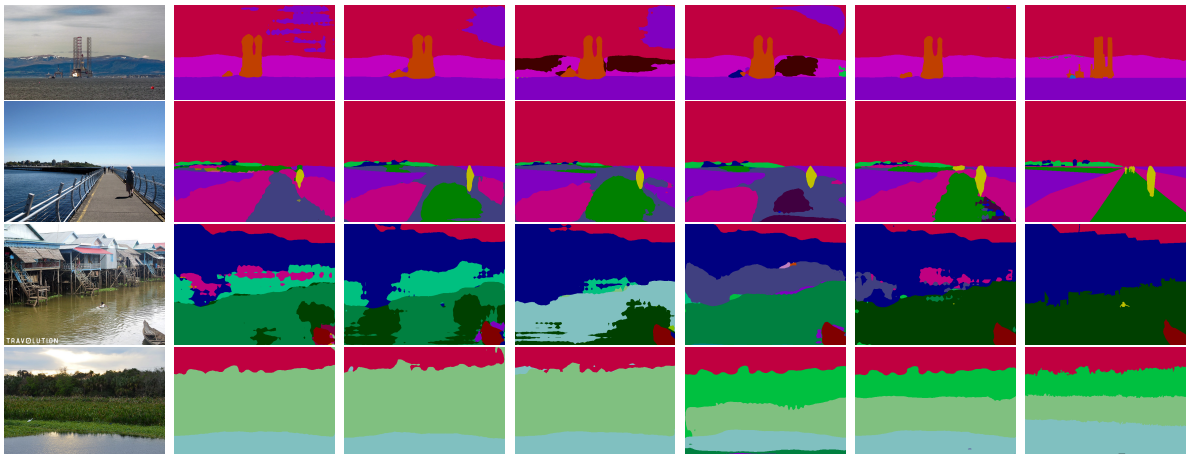


Figure 3.8 Visualisation comparison of AQUANet and four well-known methods on the ATLANTIS validation set.

3.4.2 Comparisons

We use several state-of-the-art networks to perform training and testing on ATLANTIS, including PSPNet (Zhao et al. 2017), DeepLabv3 (Chen et al. 2017b), CCNet (Huang et al. 2019), EMANet (Li et al. 2019), ANNet (Zhu et al. 2019), DANet (Fu et al. 2019), DNLNet (Yin et al. 2020), GCNet (Cao et al. 2019), OCNet (Yuan and Wang 2018), OCRNet (Yuan, Chen, and Wang 2020). For a fair comparison, we train all the networks with the same backbone (ResNet-101) for 30 epochs. As shown in Table 3.3, the proposed AQUANet outperforms all these networks on waterbody image semantic segmentation. Figure 3.8 shows the visualization results of some samples from ATLANTIS validation set. Considering the ground truth and in comparison with other networks' outputs, the boundaries between different classes are better preserved in AQUANet output. Compared with (Chen et al. 2017b), (Zhao et al. 2017), (Fu et al. 2019), and (Yuan and Wang 2018), our method achieves better results in both the aquatic and non-aquatic regions.

3.4.3 Failure cases

Due to the challenges associated with the segmentation of waterbody images, there are still many failure cases we found in the testing stage. Three failure examples are shown in Figure 3.9, from which we observe that many aquatic classes are vulnerable to misclassification– here sea is misclassified to lake and river (row 1-2) and river is misclassified to canal (row 3).

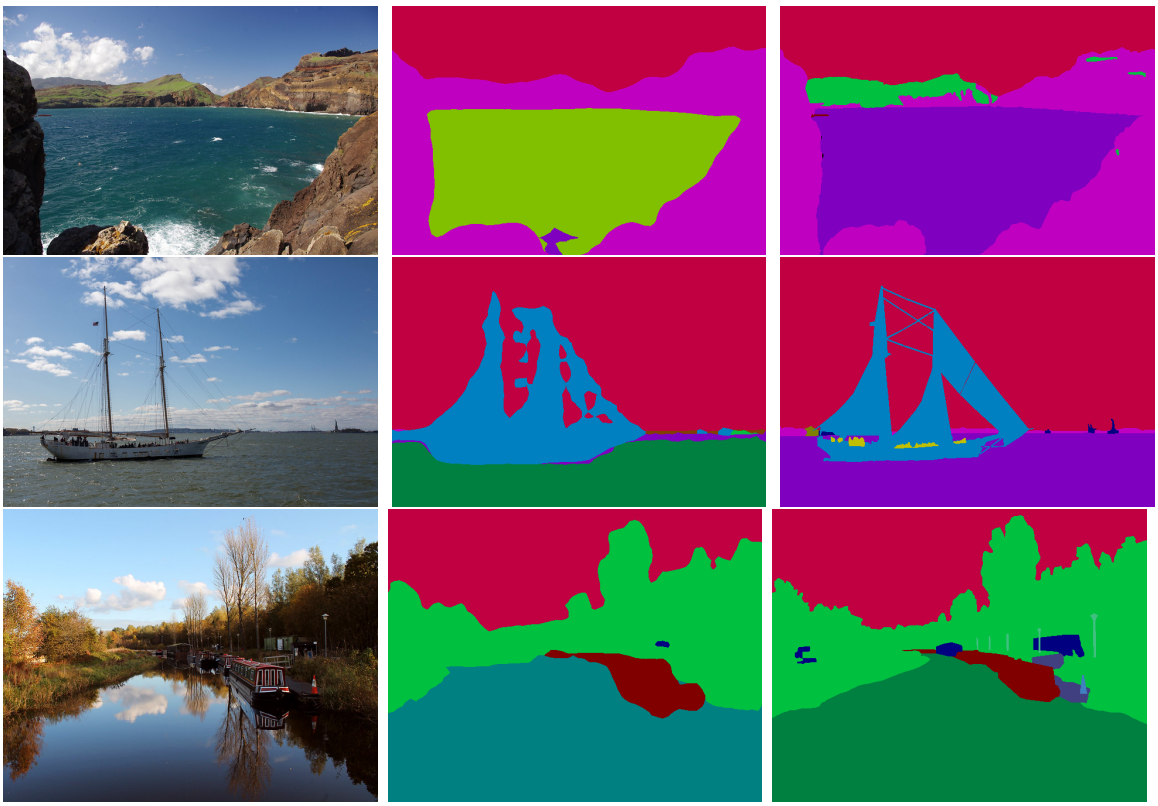


Image Ours GroundTruth

Figure 3.9 Failure cases from the ATLANTIS val set.

3.4.4 Ablation studies

We also conduct ablation studies to compare a number of different model variants of the proposed network, including the design of aquatic and non-aquatic paths, and the two feature modulations. The results are shown in Table 3.4. We can see that the design of two paths can improve the performance of waterbody image semantic

segmentation. Moreover, both the proposed low-level feature modulation (LM) and the cross-path modulation (CM) can achieve certain performance gains in terms of acc and mIoU.

Table 3.4 Ablation study of each proposed component of AQUANet on the ATLANTIS dataset.

Two Paths	LM	CM	A-acc	A-mIoU	acc	mIoU
			67.27	44.53	73.28	38.81
√			67.73	47.89	75.29	40.28
√	√		68.11	47.90	75.29	40.28
√		√	67.21	46.63	75.81	40.57
√	√	√	68.85	48.42	76.18	40.83

Compared with other state-of-the-art semantic segmentation models, AQUANet provides the highest mIoU and accuracy. In addition, by considering just labels that include water content, AQUANet still works better than others (mIoU=50.34%). In this regard, OCNet provides the second highest performance (mIoU=47.89%) and GCNet also provides the best results for canal, lake and reservoir. These results approved AQUANet is well customized for analysis of waterbodis and water-related scenes. However, considering mIoU as a more informative metric for semantic segmentation task, all approaches could only achieve 36.11%∼42.22% mIoU on the proposed dataset. It shows that the semantic segmentation of waterbodies and related objects is a challenging task and needs more research.

3.4.5 ATeX Experimental results

We further train ten well-known classification models including VGG (Simonyan and Zisserman 2014), ResNet (He et al. 2016), SqueezeNet (Iandola et al. 2016), DenseNet (Huang et al. 2017), GoogLeNet (Szegedy et al. 2015), ShuffleNet v2 (Ma et al. 2018), MobileNetV2 (Sandler et al. 2018), ResNeXt (Xie et al. 2017), Wide

ResNet (Zagoruyko and Komodakis 2016) and EfficientNet (Tan and Le 2019) on the proposed ATeX dataset. All models are implemented using PyTorch. The cross-entropy loss function is applied for training networks. We train all the networks with the same 30 training epochs, SGD optimizer with a momentum of 0.9 and weight decay of 0.0001, and batch size is set to 64. For all networks, the learning rate is first set to 2.5×10^{-4}, then it is adjusted based on the decaying rate of the resulting loss function during training. Table 3.5 shows the training time and learning rate for each model over certain 30 epochs.

Table 3.5 The performance result on ATeX test set by well-known classification models.

Networks	Time [mm:ss]	LR	Val Acc.	Test Prec.	Test Recall	Test F1
Wide-ResNet-50-2	06:56	2.5E-4	91	77	75	75
VGG-16	04:38	2.5E-4	90	75	72	72
SqueezeNet	00:47	7.5E-4	82	81	81	81
ShuffleNet V2 $\times$1.0	01:46	1.0E-2	90	90	90	90
ResNeXt-50-32 $\times$ 4d	03:15	2.5E-4	90	77	75	75
ResNet-18	01:28	2.5E-4	87	74	72	72
MobileNet V2	01:35	2.5E-4	88	74	72	72
GoogLeNet	01:51	5.0E-3	89	88	88	88
EffNet-B7	12:42	1.0E-2	90	91	91	91
EffNet-B0	02:38	7.5E-3	91	90	90	90
DenseNet-161	06:15	2.5E-4	91	81	79	79

Three common performance metrics including Precision, Recall, and F1-score are reported to evaluate the performance of the models on ATeX. Table 3.5 shows the weighted average (averaging the support-weighted mean per label) of these three metrics on the test set. Accordingly, EffNet-B7, EffNet-B0, and ShuffleNet V2 $\times$1.0 provide the best results. Considering training time, ShuffleNet V2 $\times$1.0 can be presented as the most efficient network.

3.5 Conclusion

In this paper, we introduced ATLANTIS, a large-scale dataset for semantic segmentation of waterbodies and water-related scenes, by carefully collecting images of the diverse area from the internet (Flickr) and providing high-quality annotations with the help of annotators majoring in water resources engineering. We further provided a comprehensive analysis of the characteristic of ATLANTIS and reported the performance of the current state-of-the-art by training and testing the networks on our dataset. A novel baseline network AQUANet is also proposed for waterbody image semantic segmentation and achieves the best performance on ATLANTIS. Additionally, we constructed ATLANTIS Texture (ATeX) dataset which is derived from ATLANTIS for classification and texture analysis of water. The performance of several baseline classification networks on ATeX was also evaluated and reported.

In general, digital image processing of water and water-related objects has been a complex task due to the visual challenges which are inherent in water. ATLANTIS includes images and categories beyond a specific purpose and does not focus on a certain surrounding environment for a specific purpose or limited applications. Covering such broad and diverse water and water-related categories, ATLANTIS poses significant challenges for semantic segmentation which we believe will boost new insights in both water resources engineering and computer vision communities.

Chapter 4

ATeX: A Benchmark for Image Classification of Water in Different Waterbodies Using Deep Learning Approaches [1] [2] [3]

[1]Erfani, S.M.H. and Goharian, E., 2022. Journal of Water Resources Planning and Management, 148(11), p.04022063. Reprinted here with permission of the publisher.

[2]Erfani, S.M.H. and Goharian, E., 2023. Vision-based texture and color analysis of waterbody images using computer vision and deep learning techniques. Journal of Hydroinformatics, 25(3), pp.835-850. Reprinted here with permission of the publisher.

[3]Received the 2023 award for the 'Best Research Oriented Paper' of the Journal of Water Resources Planning and Management.

Artificial Intelligence (AI) and machine learning (ML) techniques have been commonly used among water resources research communities over the past decades for different purposes, such as prediction, forecasting, and classification (Maier and Dandy 2000). Recently, deep learning (DL), as a subset of ML, has made major advances in solving problems that the AI community has not been able to resolve for many years (LeCun, Bengio, and Hinton 2015). Since 2012, as a sign of progress, DL has continuously gained more popularity for solving computer vision (CV) and visual recognition challenges (Schmidhuber 2015). These signs of success have motivated the use of DL across a wider research scope beyond computer science (Razavi 2021). DL is capable of discovering complex structures in high-dimensional data, and is therefore, applicable to various real-world problems (LeCun, Bengio, and Hinton 2015) such as autonomous driving (Cordts et al. 2016). In the water resources engineering field, although the application of ML techniques is so diverse (Mosavi, Ozturk, and Chau 2018), the adoption of DL has so far been gradual (Shen 2018). One of the main reasons for such slow progress, in particular for DL vision-based models, is indeed "water" itself.

Water scene images provide substantial information compared to the conventional water measurement methods (Lo et al. 2015), but training deep learning models on these images to derive meaningful information still remains a challenge. It is because water has certain inherent nature making the image processing tasks very difficult. In some forms and settings, water preserves intrinsic properties, such as transparency, shapelessness, and colorlessness, which brings more complications to the image processing of water and related objects. These properties can further be affected by surrounding illumination sources, and flow conditions such as sun glints, water surface reflections, turbidity, and turbulence. Moreover, different waterbodies, such as rivers and canals, or lakes and reservoirs, may have similar visual characteristics that make the task of classification of waterbody even harder.

In addition to the aforementioned inherent complexities of water for image analysis, the lack of a public dataset for water-related images significantly impedes the research community to apply ML and DL-based approaches to water resources problems. Existence of large-scale image datasets in other fields, such as ImageNet (Deng et al. 2009), PASCAL VOC (Everingham et al. 2010), Labeled Faces in the Wild (Huang et al. 2008) and more recently ADE20K (Zhou et al. 2019), Mapillary Vistas Dataset (Neuhold et al. 2017) and BDD100K (Yu et al. 2020), provide a great opportunity for researchers to develop DL-based models for real-world applications. For example, in the medical field, an exclusive large collection of annotated medical image datasets of various clinically relevant anatomies is publicly available which facilitates the development of DL-based semantic segmentation models in this field (Simpson et al. 2019; Mzurikwao et al. 2020).

Our extensive search to find a relevant image dataset that adequately focuses on water and water-related objects led to a few image datasets with limited applications. (Gebrehiwot et al. 2019) collected a small number of top-view waterbody datasets (100 images) taken by Unmanned Aerial Vehicles (UAVs). This dataset contains only four classes (i.e., water, building, vegetation, and road). (Sazara, Cetin, and Iftekharuddin 2019) introduced a larger dataset (253 images) while it just includes segmentation of flooded regions in images within certain types of the surrounding environments. More recently, (Sarp et al. 2020) put together a dataset that consists of 441 annotated images of flooded roads. However, all these datasets pose multiple limitations either in terms of the number of images, the diversity of the water categories, and the specific application (semantic segmentation) they focused on. This is mainly due to the difficulties associated with the mass collection of various water-related images, which can be a very laborious and time-consuming task. There is no specific repository providing relevant images and also team members are required to have prior knowledge in the water resources engineering field to be able to correctly

select relevant images for each label. This calls for the development of an image dataset, which

- Covers a wide range of natural and built waterbodies, and doe not consider "water" just as a general class,
- Emphasizes classification and texture analysis of water images, which are vital backbones for developing ML and DL-based models for many other CV tasks,
- Includes images beyond a specific purpose, and does not focus on a certain surrounding environment for the limited application, and
- Contains a large enough number of images for training, validation, and testing DL models.

As our first effort and to cope with the lack of a water dataset, we have previously developed ATLANTIS, a benchmark for semantic segmentation of waterbody images, which covers a wide range of fully annotated natural and man-made (artificial) water objects in images, such as seas, lakes, rivers, reservoirs, canals, and piers. ATLANTIS includes 5,195 pixel-wise annotated images which are divided into 3,364 training, 535 validation, and 1,296 testing images. In addition to the 35 waterbody and water-related objects, this dataset covers 21 general labels, such as car, vegetation, road, etc.

In this chapter, we introduced a new benchmark, ATeX (ATLANTIS TeXture) for classification and texture analysis of water (in different waterbodies) affected by different surrounding environments. Classification is one of the core problems in CV that, despite its simplicity, has a large variety of practical applications. Moreover, many other seemingly distinct CV tasks (such as object detection, and segmentation) can be reduced to image classification. For the first time, this dataset has covered a wide range of waterbodies such as estuary, swamp, glacier, puddles, etc. as shown in Figure

4.1. ATeX includes patches with 32×32 pixels of 15 waterbodies. ATeX consists of 12,503 patches split into 8,753 for training, 1,252 for validation and 2,498 for testing. Moreover, this chapter aimed to investigate vision-based texture and color analysis techniques on the water to find visual features which can distinguish water in different waterbodies. Texture information is applicable to configuring the architecture of the CNN-based models as the recognition strategy of CNNs follows local to global features in different layers of the forward pass. Thus, three different approaches, including two conventional methods (based on the hand-craft features) and a Deep Learning (DL) model, were built to extract texture features of water. The quality of extracted features was then evaluated using K-Nearest Neighbors (KNNs). This section represents a portion of the results from (Erfani and Goharian 2023). Furthermore, ten well-known deep learning models including VGG-16 (Simonyan and Zisserman 2014), ResNet-18 (He et al. 2016), SqueezeNet (Iandola et al. 2016), DenseNet-161 (Huang et al. 2017), GoogLeNet (Szegedy et al. 2015), ShuffleNet V2×1.0 (Ma et al. 2018), MobileNetV2 (Sandler et al. 2018), ResNeXt-50-32 $\times$ 4d (Xie et al. 2017), Wide ResNet-50-2 (Zagoruyko and Komodakis 2016), EfficientNet-B0 and EfficientNet-B7 (Tan and Le 2019) are trained on ATeX images and results are reported and discussed in the following sections.

4.1 Dataset Discription

Water does not preserve the same texture, form, and visual features in different environments. The physical and chemical properties of water affect water appearance of different waterbodies. Turbidity, color, temperature, suspended particles, and dissolved substances are among those water characteristics that potentially play important role in water appearance. Moreover, water depth, flow rate, and regular form of its container (such as the form of a reservoir or channel) are among those properties which can dictate the flow regime and appearance of water. Water has also a

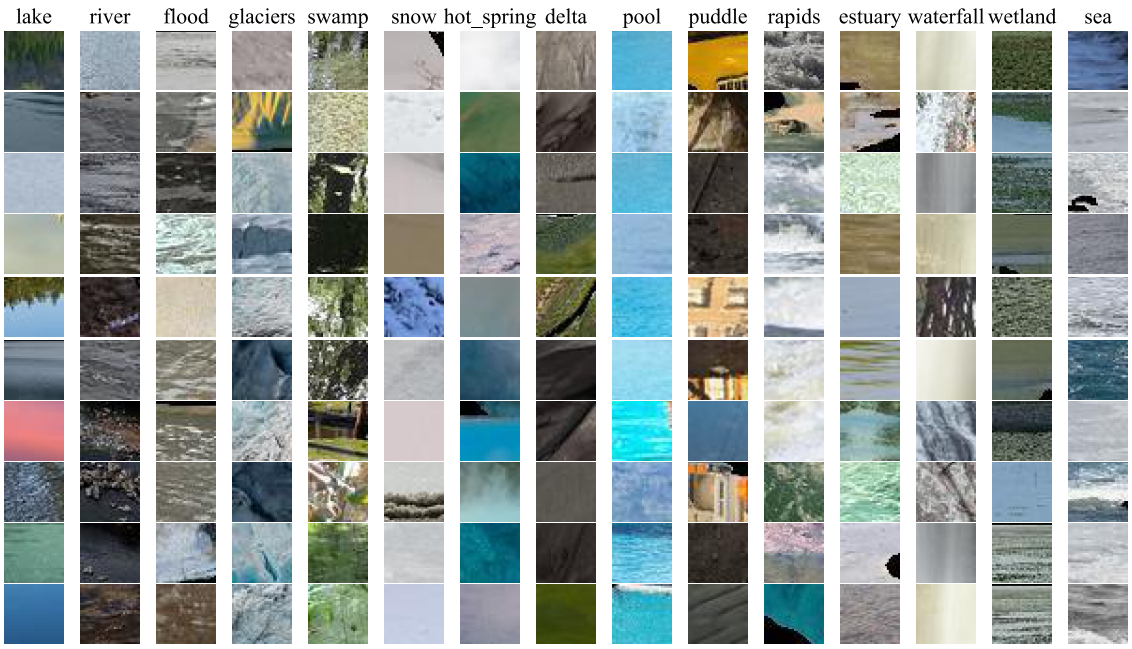


Figure 4.1 Samples of ATeX patches. Water in different waterbodies displays different image textures.

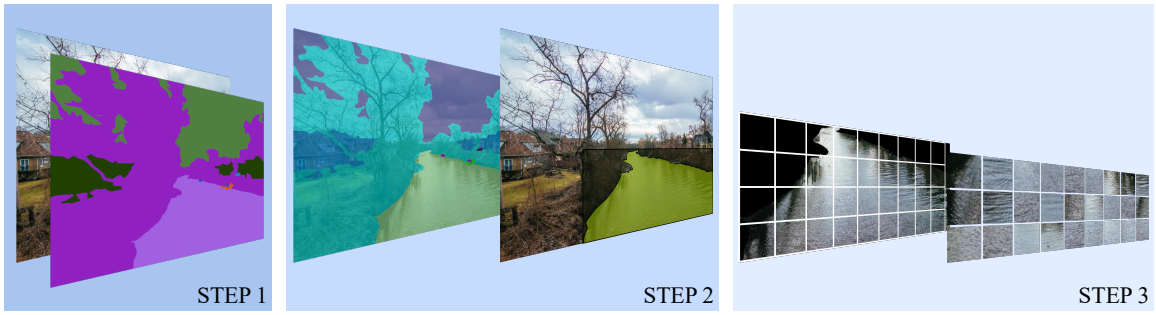


Figure 4.2 ATeX patches are derived from ATLANTIS (ArTificiaL And Natural waTer-bodIes dataSet). Waterbodies' parts are extracted from images using a ground-truth mask (Step 1), the irrelevant pixels are cut based on waterbodies' coordination (Step 2), and the outputs are cropped 32×32 to create ATeX patches (step 3).

reflective surface. For example, under laminar flow conditions, depending on ambient light, the consistent reflection can be dominant, while under turbulent flow, unique features of the flow regime such as eddies result in varying reflection from the water surface (e.g., white water effect). Moreover, turbulent regime plays a critical role in terms of accretion and transport of sediment as well as contaminant mixing and dispersion in rivers having a direct effect on water turbidity and visual appearance.

Considering different water features and their combinations, water can appear in completely different forms and colors. The ATeX dataset is designed and developed with the goal of representing various textures in which water usually appears in different waterbodies. ATeX's images are derived from ATLANTIS (ArTificiaL And Natural waTer-bodIes dataSet). ATLANTIS is a semantic segmentation dataset including 5,195 pixel-wise annotated images that covers a wide range of natural and artificial waterbodies such as sea, lake, river, reservoir, canal, and pier. ATLANTIS images are manually annotated by trained students with adequate prior knowledge of water resources and hydraulic structures. ATeX will be also utilized in DL-based model development for digital image processing of water. Training and transfer learning of Deep Neural Network (DNN) architectures that serve as the backbones of many modern ML algorithms using ATeX can be applied in turbidity measurement, water quality estimation (Peterson, Sagan, and Sloan 2020), land-use/land-cover and flood mapping (Gebrehiwot et al. 2019), and better surface water monitoring and measurement (Moy de Vitry et al. 2019). Figure 4.2 shows the pipeline through which the ATeX images are cropped from ATLANTIS images. As it is shown in this figure, there are no partial overlaps between patches. Figure 4.3 shows the frequency distribution of the number of images for each waterbody label in ATeX.

4.2 Methodology

4.2.1 Texture Representations

Facial recognition and expression task is very similar to water detection and classification. In face recognition texture representations and spatial patterns of face components (eyes, nose, lips, etc.) provide valuable information for recognition. For water, however, spatial information provides no significant information, as pattern subelements (textons) resulting from filter-based texture representations do not follow any specific spatial coordinates (Julesz 1981). Thus, in this case, the texture of

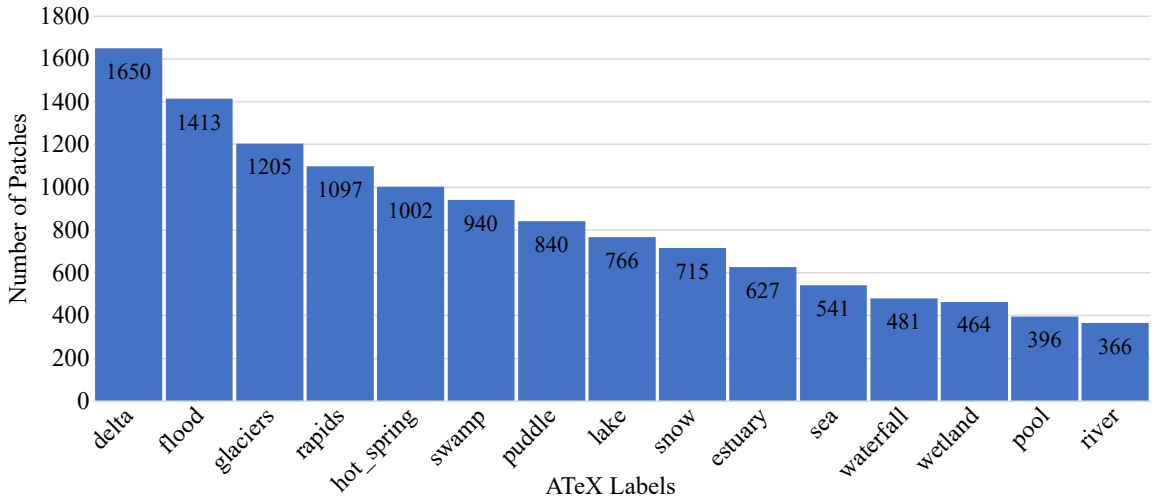


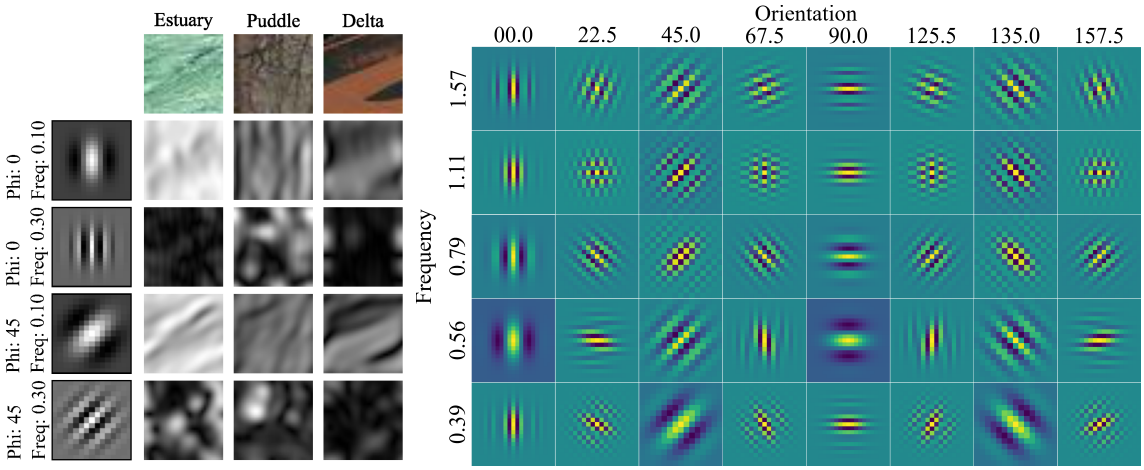
Figure 4.3 The frequency distribution of the number of images for 15 waterbodies.

the water is the only source of information for the accurate classification of waterbodies. In this section, Gabor kernels filter and local binary patterns (LBP) descriptor are used to extract texture information. The quality of the resulting texture representations is then evaluated and compared with the KNNs classification method in the following section.

Gabor Filters

Gabor wavelets have been commonly used for the task of face recognition (Shen and Bai 2006; Vinay et al. 2015). Frequency and orientation representations of Gabor filters are claimed to be very similar to those of human visual system (Olshausen and Field 1996). They have been found to be particularly appropriate for texture representation and discrimination. In the spatial domain, a 2D Gabor filter is a Gaussian kernel function that is modulated by a sinusoidal plane wave. The Gabor wavelet representation facilitates the recognition without correspondence (hence, no need for manual annotations) as it captures the local structure which corresponds to spatial frequency (scale), spatial localization, and orientation selectivity. As a result, the Gabor wavelet representation is not sensitive to changes caused by illumination

changes and subtle nuances (Liu and Wechsler 2002). The texture representations resulting from Gabor filters on ATeX waterbody patches are shown in the figure 4.4a.



(a) (b)

Figure 4.4 (a) The visual results of four different scales and orientation Gabor filters on three waterbody patches. (b) The real part of the Gabor kernels at five scales and eight orientations with the following parameters: $\sigma = \pi$, $k_{max} = \pi/2$, and $f = \sqrt{2}$.

Gabor wavelets (kernels, filters) in this paper are defined based on (Liu and Wechsler 2002) as follows:

$$\psi_{\mu,\nu}(z) = \frac{\|k_{\mu,\nu}\|^2}{\sigma^2} e^{(\|k_{\mu,\nu}\|^2 \|z\|^2 / 2\sigma^2)} \left[e^{ik_{\mu,\nu} z} - e^{-\sigma^2/2} \right] \tag{4.1}$$

where μ and ν define the orientation and scale of the Gabor kernels, $z = (x, y)$, and $\|.\|$ denotes the norm operator. Wave vector $k_{\mu,\nu}$ is defined as follows:

$$k_{\mu,\nu} = k_\nu e^{i\phi_\mu} \tag{4.2}$$

where $k_\nu = k_{max}/f^\nu$ and $\phi_\mu = \pi\mu/8$. Maximum frequency, k_{max}, is the spacing factor between kernels in the frequency domain. Five different scales, $\nu \in \{0, \ldots, 4\}$, and eight orientations, $\mu \in \{0, \ldots, 7\}$ are applied with the following parameters: $\sigma = 2\pi$, $k_{max} = \pi/2$, and $f = \sqrt{2}$. The kernels exhibit desirable characteristics of spatial frequency, spatial locality, and orientation selectivity. Figure 4.4b shows different combinations of frequency and orientation of the filters applied in this study.

In this study, we set $\sigma = \pi$ to decrease the size of the kernels and alleviate computational complexity, because 2D Gabor filters are Gaussian-based, so the values of a Gaussian function at a distance larger than 3σ from the mean are small enough to be ignored (Gonzalez 2009). The experiments have been separately run for RGB, grayscale, and HSV color space of ATeX patches to compare texture representations within each color space. The following steps discuss the experimental procedure for grayscale images as it is shown in Figure 4.5:

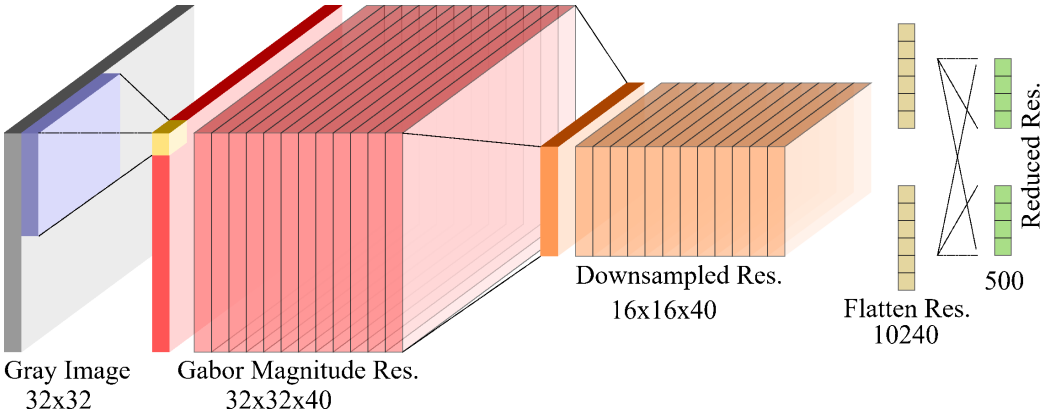


Figure 4.5 Augmented feature tensor pipeline for Gabor Kernels.

1. First, ATeX patches are imported as gray values (N×32×32) where N represents the number of patches.

2. Multi-scale and multi-orientation Gabor filters are applied and corresponding Gabor magnitude responses are obtained. Then, all responses are concatenated to build an augmented feature tensor for each patch N×32×32×40.

3. Each augmented feature tensor is downsized in space from 32×32 to 16×16 using "MaxPooling" operator which results in size of N×16×16×40.

4. Finally, all 10,240 features, resulting from 16×16×40, for each patch, are reduced to the top 500 dimensions of the data with the highest variance using Principal Component Analysis (PCA).

The same procedure is repeated for RBG and HSV color spaces, while the first step is just different to represent the color channels of images. In HSV (RGB) experiments, input patches have an additional dimension N×32×32×3, which represents Hue, Saturation, and Value (Red, Green, and Blue) of each patch. Accordingly, Gabor filters are constructed dimensionally compatible, but Gabor magnitude responses and resulting augmented feature tensor have the same dimensions as described for the grayscale patches.

Local Binary Patterns

Local binary patterns (LBP) is a type of visual descriptor used for classification in computer vision. The original LBP operator is introduced by (Ojala, Pietikäinen, and Harwood 1996). The operator labels the pixels of an image by comparing the 3×3 surrounding neighborhood of each pixel with the center value using a binary system. The corresponding location of the pixel on the binary map gets 1 if the value of the surrounding pixel is more than the center pixel and it gets 0 vice versa. Then the histogram of the labels can be used as a texture descriptor. Figure 4.6a shows the basic LBP operator (Ahonen, Hadid, and Pietikäinen 2004).

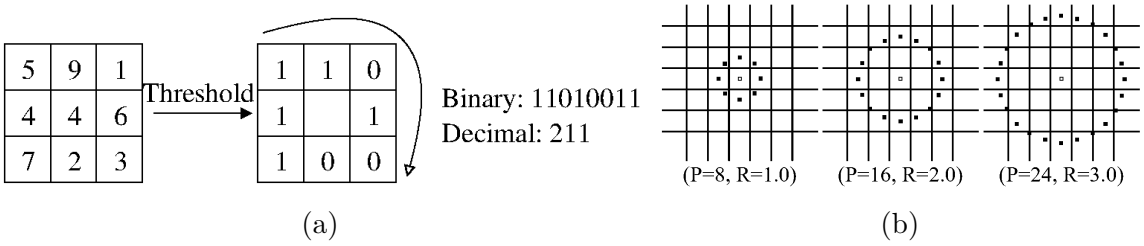


Figure 4.6 (a) The basic LBP operator Ahonen, Hadid, and Pietikäinen 2004. (b) Circularly symmetric neighbor sets for different (P, R) (Ojala, Pietikainen, and Maenpaa 2002).

The LBP operator can be extended to include more neighbor pixels Ojala, Pietikainen, and Maenpaa 2002. The circular approach and bilinearly interpolating the pixel values enable any radius and number of surrounding pixels. In this approach, we will

use the notation (P, R) which means P sampling points on a circle of radius of R. Figure 4.6b shows different sampling points for different radius.

Another extension to the original LBP operator considers so-called "uniform" patterns (Ojala, Pietikainen, and Maenpaa 2002). A Local Binary Pattern is called uniform if it contains at most two bitwise transitions from 0 to 1 or vice versa when the binary string is considered circular. For example, `00000000`, `00011110` and `10000011` are "uniform" patterns.

When surrounding pixels are all black or all white, then that image region is flat. Groups of consecutive black or white pixels are considered "uniform" patterns. "Uniform" patterns can be interpreted as corners or edges. If pixels switch back and forth between black and white, the pattern is considered "non-uniform". The following notation is used for the LBP operator in this study: $LBP^{u^2}_{P,R}$. The subscript represents using the operator in a (P, R) neighborhood, and superscript u^2 stands for using only uniform patterns and labeling all remaining patterns with a single label. Figure 4.7 shows an example of $LBP^{u^2}_{24,3}$ on the river delta patch from ATeX, where the flat, edge-like, and corner-like regions of the image are highlighted on images and histograms.

The histogram of labeled image $f_l(x, y)$ is defined as

$$H_i = \sum_{x,y} I\{f_l(x, y) = i\}, i = 0, \ldots, n-1 \tag{4.3}$$

where n is the total number of different labels produced by the LBP operator and

$$I\{A\} = \begin{cases} 1 & \text{A is True} \\ 0 & \text{A is False.} \end{cases} \tag{4.4}$$

Different parameters, i.e., sampling points P, radius R, and uniform or non-uniform pattern, offer different resulting histograms. So it is important to adjust the parameters based on the problem. In this study, sampling points, radius, and pattern

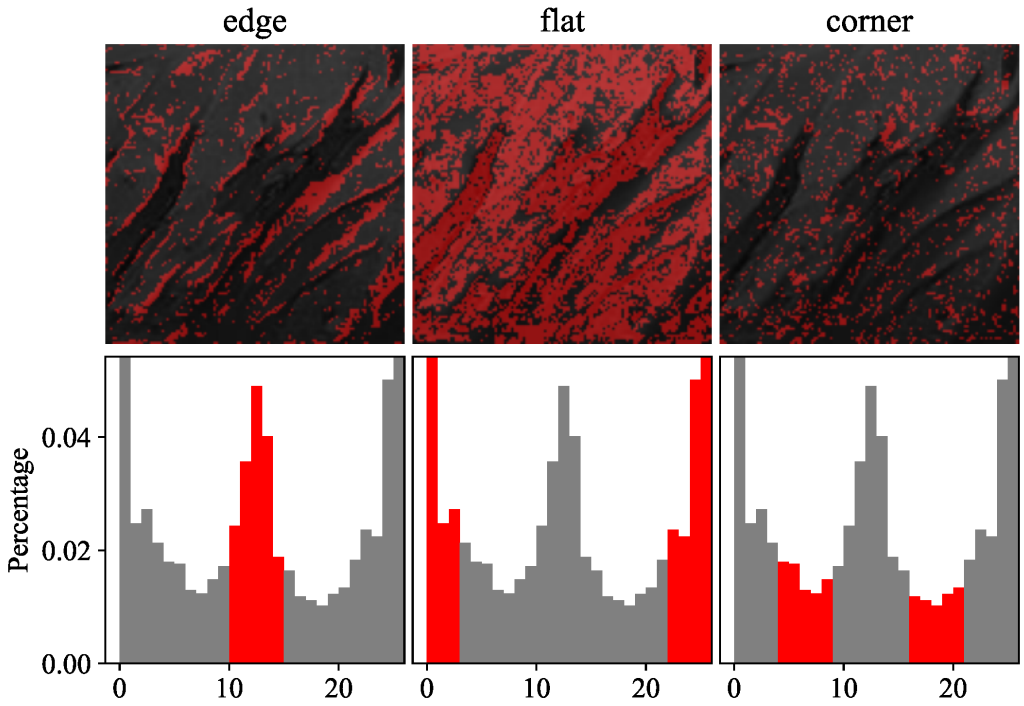


Figure 4.7 Different patterns are highlighted on both image and histogram resulting from LBP response.

are considered 8, 1, and "uniform," i.e., $LBP^{u^2}_{8,1}$, respectively. Another consideration is histograms cannot preserve the spatial information across the image. So, for the face recognition problem, it is suggested to implement regional LBP for different parts of the face to preserve the spatial information (Ahonen, Hadid, and Pietikäinen 2004). In the case of water, however, due to the irregularity of water features in spatial coordinates, regional LBP would not be effective. Moreover, several possible dissimilarity measures have been proposed for histograms. Log-likelihood, Chi-square, and Kullback-Leibler Divergence are provided in this study according to the following equations:

- Log-likelihood statistic:

$$L(P, Q) = -\sum P \log Q \tag{4.5}$$

- Chi-square statistic:

$$\chi^2(P,Q) = \sum \frac{(P-Q)^2}{P+Q} \tag{4.6}$$

- Kullback-Leibler Divergence:

$$D_{KL}(P\|Q) = \sum P \ln \frac{P}{Q} \tag{4.7}$$

Where, P and Q are two probability distributions.

Deep Learning-based Representations

Preliminary classification results on low-level vision-based methods (section **??**) showed the important role of texture information in the better performance of KNNs in classifying different waterbodies. In this section, we investigate the performance of the DL-based model in feature extraction. DL models are capable of automatically learning patterns from raw data through multiple layers of processing (LeCun, Bengio, and Hinton 2015). It is because in these models, each layer transforms the input representation into a higher-level representation which let the deeper layers learn more important aspects of the raw data and discard irrelevant variations (higher-level representation) (Eltner et al. 2021).

We train ShuffleNet V2×1.0 (Ma et al. 2018), a DL-based classification model designed for mobile devices with very limited computing power, on ATeX dataset. The PyTorch pre-trained ShuffleNet V2×1.0 is fine-tuned on 32×32 patches with 30 training epochs, we use SGD optimizer with a momentum of 0.9 and weight decay of 0.0001, and the learning rate and batch size are set to 1.0×10^{-2} and 64, respectively.

ShuffleNet V2 is an efficient Convolutional Neural Network (CNN) architecture inspired by ShuffleNet (Zhang et al. 2018). ShuffleNet is a network architecture widely adopted in low-end devices such as mobiles. In ShuffleNet architecture, "bottleneck" building block (He et al. 2016) is modified by two new operations, *pointwise* group convolution and *channel shuffle*, to greatly reduce computation cost while maintaining

accuracy. *Pointwise* group convolution is introduced to reduce computation complexity of 1×1 convolution (bottleneck). *Channel shuffle* operation is also provided to overcome the side effects brought by group convolutions (Figure 4.8a).

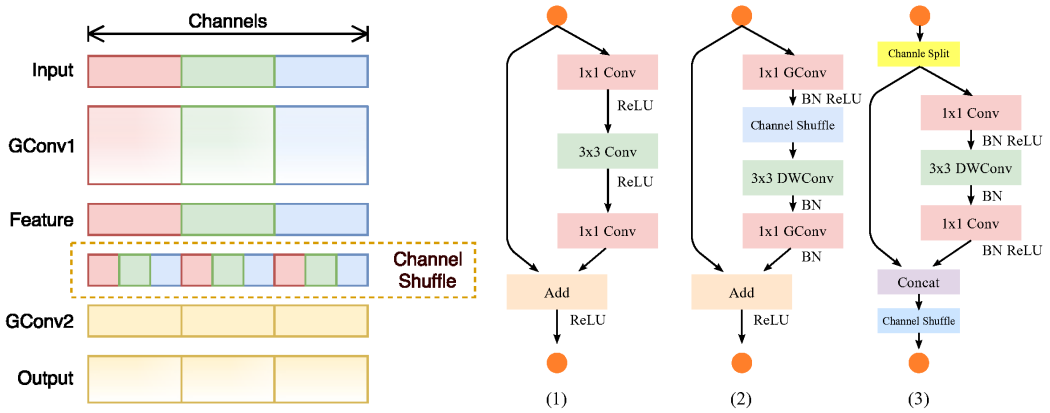


Figure 4.8 (a) Channel shuffle with two stacked group convolutions. GConv stands for group convolution (Zhang et al. 2018). (b) ShuffleNet Units. 1) bottleneck unit (He et al. 2016); 2) ShuffleNet unit with depthwise convolution (DWConv) (Chollet 2017; Howard et al. 2017), pointwise group convolution (GConv) and channel shuffle Zhang et al. 2018; 3) ShuffleNet V2 unit (Ma et al. 2018) using channel split operator.

In ShuffleNet V2, the ShuffleNet unit is modified and a simple operator called "channel split" is introduced at the beginning of each unit (Figure 4.8b). The two 1×1 convolutions are no longer group-wise (unlike (Zhang et al. 2018)) and the same "channel shuffle" operation as in (Zhang et al. 2018) is used to enable information communication between the two branches.

4.2.2 Application of Deep Learning Models

Deep learning algorithms are capable of learning patterns from raw data through multiple layers of processing based on artificial neural networks (ANNs). Each layer transforms the input features into higher-level (more complicated) features. Considering different types of problems, the deeper layers detect and learn more important features from the raw input data (related to the problem) while discarding irrelevant information (Eltner et al. 2021). The advancement in Deep Learning has been

constructed primarily over a particular algorithm, Convolutional Neural Networks (ConvNets). ConvNets are very similar to ordinary ANNs. They are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. The whole network still expresses a single differentiable score function; from the raw image pixels on one end to class scores at the other end, and has a loss function (e.g. SVM/Softmax) on the last (fully-connected) layer.

ConvNets have been successfully used in a variety of computer vision tasks. ConvNets can be used for classification, making predictions for the whole input (Krizhevsky, Sutskever, and Hinton 2012; Russakovsky et al. 2015; Szegedy et al. 2015; Simonyan and Zisserman 2014; He et al. 2015; Huang et al. 2017; Hu, Shen, and Sun 2018; Zoph et al. 2018) or object detection (localization), which provides not only the classes but also additional information regarding the spatial location of those classes (Girshick et al. 2014; Ren et al. 2015; He et al. 2017), and finally, semantic segmentation which achieves fine-grained inference by making dense predictions inferring labels for every pixel, so that each pixel is labeled with the class of its enclosing object or region (Long, Shelhamer, and Darrell 2015; Chen et al. 2017a; Badrinarayanan, Handa, and Cipolla 2015; Noh, Hong, and Han 2015; Lin et al. 2017; Yuan and Wang 2018; Zhao et al. 2017; Yuan, Chen, and Wang 2019).

In ConvNet architecture, it is explicitly assumed that the inputs are images, which allows encoding certain properties (pixel intensity values) into the architecture. These customized architectures make the forward function more efficient to implement and vastly reduce the number of parameters in the network. Unlike a regular ANN, the layers of a ConvNet have neurons arranged in 3 dimensions: width, height, and depth. As it is shown in Figure 4.9, the input images in ATeX are the input volume of activations, and the volume has dimensions 32×32×3 (width, height, depth respectively). The neurons in a layer are only connected to a small region of the layer before it.

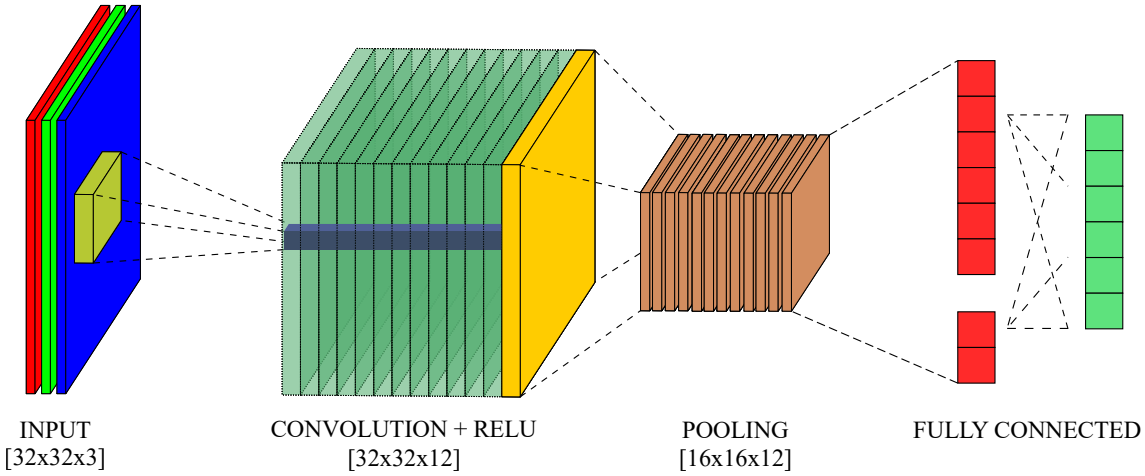


Figure 4.9 A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. In this example, the green input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, Blue channels).

The final output layer has $1\times1\times15$ dimensions for ATeX, because, by the end of the ConvNet architecture, the full image size is reduced into a single vector of class scores, arranged along the depth dimension.

A simple ConvNet consists of a sequence of layers, and every layer of a ConvNet transforms one volume of activations to another through differentiable functions. Three main types of layers are generally used to build ConvNet architectures: Convolutional Layer, Pooling Layer, and Fully-Connected Layer (similar to regular ANNs). These layers are stacked to form a full ConvNet architecture (Figure 4.9).

ConvNets transform the original image from the original pixel values to the final class scores through all the layers. Input $[32\times32\times3]$ holds the raw pixel values of the image, in this case an image with 32 width, 32 height, and three color channels of R, G, B. "Convolution" layer computes the output of neurons that are connected to the local regions of the input. The computations are dot products between their weights (of filters) and a small region they are connected to in the input volume. This may result in volume such as $[32\times32\times12]$ if we decided to use 12 filters. The CONV layer's parameters consist of a set of learnable filters. Each filter has a small

size (width and height) but extends through the full depth of the input volume. For example, a typical filter on a first layer of a ConvNet may have a size of 5×5×3 (i.e. 5 pixels width and height, and 3 because images have depth 3, the color channels). "ReLU" layer will apply an element-wise activation function, such as the $max(0, x)$ thresholding at zero. This leaves the size of the volume unchanged ([32×32×12]). The "Pooling" layer performs a downsampling operation along the spatial dimensions (width, height), resulting in volume such as [16×16×12]. The "Fully Connected" layer computes the class scores, resulting in a volume of size [1×1×15], where each of the 15 numbers corresponds to a class score, such as among the 15 categories of ATeX. In this layer (Fully Connected) computation is like ordinary Neural Networks, i.e. each neuron in this layer will be connected to all the neurons in the previous volume.

Table 4.1 summarizes the important features and properties of all CNN-based models trained on ATeX in this study. Mobilenets (Howard et al. 2017) and ShuffleNet (Zhang et al. 2018) is not used in this study, but they are presented in this table to better explain their next generations MobileNetV2 (Sandler et al. 2018) and ShuffleNet V2 (Ma et al. 2018).

Experimental settings

In order too evaluate the performance of different ConvNets on ATeX, 11 well-known pre-trained models, including VGG-16, ResNet-18, SqueezeNet, DenseNet-161, GoogLeNet, ShuffleNet V2×1.0, MobileNet V2, ResNeXt-50-32 × 4d, Wide ResNet-50-2 and EfficientNet (EffNet-B0 and EffNet-B7), are fine-tuned using the ATeX train set images. The networks are trained using a fully supervised fashion constrained by the widely-used cross-entropy loss function. For the sake of fair comparison between networks, all networks are trained for a similar 30 epochs using Stochastic Gradient Descent (SGD) optimizer method with a momentum of 0.9 and weight decay of 0.0001, and batch size of 128. The learning rate is first set to

Table 4.1 The important features of the networks trained on ATeX.

Network	Year	Significant Properties	References
VGG	2014	Deep architecture Small CONV filters (3×3) Fewer parameters	(Simonyan and Zisserman 2014)
GoogLeNet	2015	"Inception" module Split-transform-merge strategy No fully connected layers Significantly less parameters Parallel filter operations "Bottleneck", 1×1 CONV layers "Auxiliary" classification outputs	(Szegedy et al. 2015)
ResNet	2016	Introduced "Residual blocks" Train up to hundreds of layers	(He et al. 2016)
Wide ResNet	2016	Wider residual blocks Increasing width instead of depth Increasing the filter sizes Considering "dropout" Efficient GPU computations	(Zagoruyko and Komodakis 2016)
SqueezeNet	2016	Smaller network, fewer parameters "Squeeze" and "expand" layers 3×3 replaced with 1×1 filters 3×3 input channels	(Iandola et al. 2016)
ResNeXt	2017	"Cardinality" "Grouped convolutions"	(Xie et al. 2017)
DenseNet	2017	Maximized information flow Connected all layers directly	Huang et al. 2017
Mobilenets	2017	Light weight deep neural networks "Depthwise separable convolutions"	(Howard et al. 2017)
MobileNetV2	2018	Inverted residual structure Shortcut connections No non-linearities for narrow layers	(Sandler et al. 2018)
ShuffleNet	2018	Limited computing power "Pointwise group convolution" "Channel shuffle"	(Zhang et al. 2018)
ShuffleNet V2	2018	"Channel split" No "group-wise" CONV	(Ma et al. 2018)
EfficientNets	2019	Balanced depth, width, resolution Expand the network No change in architecture unit	(Tan and Le 2019)

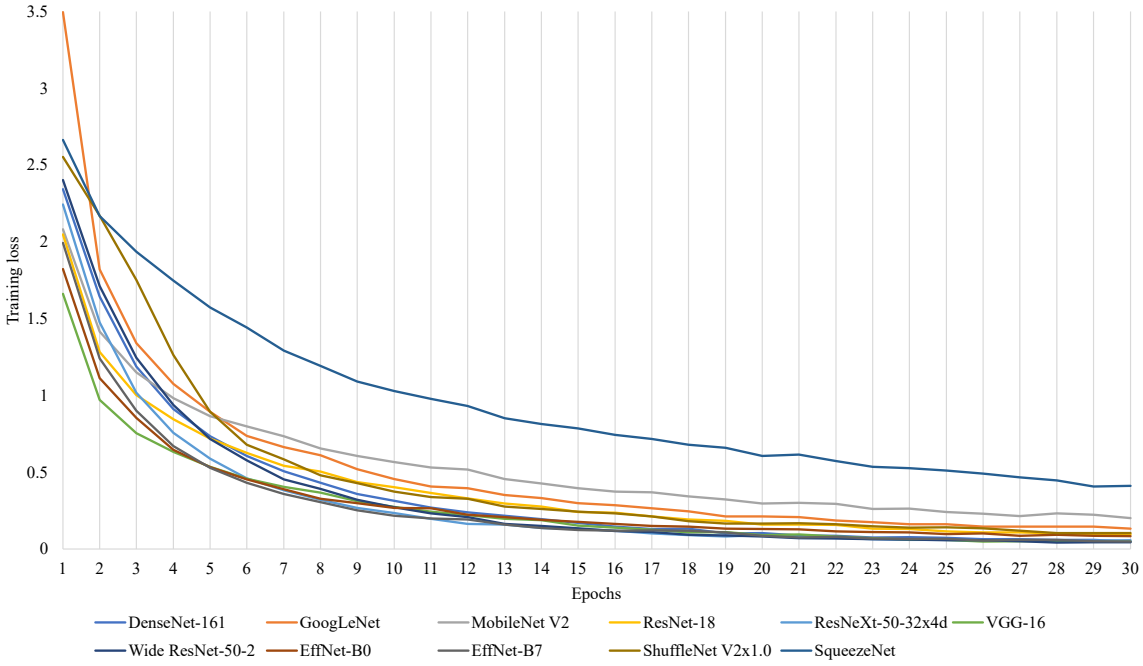


Figure 4.10 The loss function of different models over time for the training set.

2.5×10^{-4}, then it is adjusted based on a decaying rate of the loss function during training. On the second trial, each of these models is trained using a customized learning rate that works better for that particular network. Figure 4.10 shows the training loss function results over time (training epochs) for all the networks. According to Figure 4.10 loss function decay rate on training epochs looks reasonable in all networks. Table 4.3 also shows the training time (T-Time) and learning rate (LR) used for each model over 30 epochs. In the case of GoogLeNet, as an exception, the loss function imposes constrain on the final prediction P (main loss) and two intermediate features (auxiliary losses) with weights of 1.0 and 0.4 for the main and the auxiliary losses, respectively. During training, the base learning rate is set to 7.5×10^{-3} and it is decayed following the poly policy (Zhao et al. 2017).

4.3 Results and Discussion

4.3.1 Texture Representations

Features extracted from different methods are fed into customized K-Nearest Neighbors (KNNs) using different dissimilarity metrics (Euclidean [L2N], Chi-square [χ^2], Log-likelihood [LLV] and Kullback-Leibler Divergence [KLD]) for estimating the accuracy of classification on the validation set. For raw images with different color spaces, and texture features captured from Gabor magnitude responses, L2N is used as the distance metric. In the case of LBP, the dissimilarity of histograms of "patterns" resulting from LBP operation is compared using three different dissimilarity measures including χ^2, LLV, and KLD. In the case of the DL model, and in order to achieve fair results, images of the validation set are first passed into the feature layer of the pre-trained ShuffleNet V2×1.0 (on ATeX training set), then the extracted features were fed into the KNNs.

Despite the fact that KNNs is considered machine learning tools, it is a simple data-driven method that just compares the validation patches with all training ones and reports the dissimilarity vector for each validation input. So, the final results show just the performance of filters in extracting the features of patches, and the classification method does not have any significant effect on the performance of filters.

The accuracy results are reported in Table 4.2. The results are categorized based on image color space, feature extraction methods, model parameters, and metrics for dissimilarity evaluation. The number of neighbors (K) for KNNs ranges from 1 to 500. Results show the highest performance of DL model features ranging from 88% to 92% depending on the number of neighbors. After the DL model, the raw images in HSV color space, and LBP methods (independent of dissimilarity measurement method) offer the best performance with 50% and 35% accuracy, respectively. On the other hand, raw images in RGB color space provide the lowest performance with

6% accuracy. For grayscale images, Gabor magnitude responses show 29% accuracy, while the highest performance for raw grayscale images does not exceed 24% accuracy.

It is worth mentioning that considering the high computational complexity and processing time needed for convolution operation, Gabor operation is too expensive compared with LBP operation.

Table 4.2 The experimental results on ATeX validation set. Raw images, Gabor responses, and features extracted from ShuffleNet are evaluated based on L2N measurement (numbers are in percent).

K	Image			Gabor Wavelets			LBP			DL
	RGB	G	HSV	RGB	G	HSV	KLD	χ^2	LLV	
1	6	20	50	8	23	18	24	24	24	92
3	6	20	48	9	25	20	26	27	27	92
5	6	20	49	9	27	21	28	29	29	92
8	6	21	49	9	28	22	31	31	30	92
15	6	19	47	9	28	22	33	32	33	91
50	6	21	44	9	29	22	35	34	34	91
70	11	20	43	10	29	22	35	35	35	90
100	11	21	43	10	27	22	34	34	33	90
200	11	22	40	10	27	23	32	33	32	89
300	11	22	38	10	27	23	32	32	32	89
500	11	24	36	9	27	23	30	30	30	88

4.3.2 Deep Learning Models

Three common performance metrics, Precision, Recall, and F1-score are used to evaluate and compare the performance of trained models on ATeX test set images. Table 4.3 shows the weighted average (averaging the support-weighted mean per label) of these three metrics for the test set. In addition to the performance metrics, i) learnable parameters ("Params" in Table 4.3), which includes the total weights and biases of each model that are commonly used to measure the size of neural networks, ii) FLOPs which stands for floating-point operations per second and refers to the total number of multiplication-addition operations for each model, and iii) total size ("Size") which covers the memory size for a batch of input images (128×32×32×3

Table 4.3 The performance result on ATeX test set by well-known classification models.

Networks	T-Time [mm:ss]	LR	Test Set			Model Summary		
			Prec.	Recall	F1	Params	FLOPs (M)	Size (MB)
Wide-ResNet	06:56	2.5E-4	77	75	75	66,864,975	368.99	260.30
VGG	04:38	2.5E-4	75	72	72	134,321,999	567.11	515.32
SqueezeNet	00:47	7.5E-4	81	81	81	743,119	10.61	3.99
ShuffleNet	01:46	1.0E-2	90	90	90	1,268,979	5.74	6.20
ResNeXt	03:15	2.5E-4	77	75	75	23,010,639	135.00	93.01
ResNet	01:28	2.5E-4	74	72	72	11,184,207	59.52	44.19
MobileNet	01:35	2.5E-4	74	72	72	2,243,087	10.12	9.63
GoogLeNet	02:51	7.5E-3	90	90	90	5,615,279	46.10	23.18
EffNet-B7	12:42	1.0E-2	91	91	91	62,185,247	179.22	244.20
EffNet-B0	02:38	7.5E-3	90	90	90	3,616,299	12.07	15.59
DenseNet	06:15	2.5E-4	81	79	79	26,505,135	234.26	108.16

in this study), as well as forward/backward pass memory size and size of parameters are recorded for all the models (Table 4.3).

Model Evaluation Using Performance Metrics

By looking at the results from estimated performance metrics (Table 4.3), it is found that EffNet-B7, EffNet-B0, GoogLeNet, and ShuffleNet V2 $\times$1.0 perform better in comparison to the other models. Further, considering other factors such as training time, the total number of parameters, and total memory usage, ShuffleNet V2 $\times$1.0 can be selected as the most efficient network among all implemented models. Considering relatively short training time, 1 minute and 46 seconds, this model provides the second highest performance, 90 percent, on all three metrics. In the following paragraphs, the performance of each model in terms of Precision, Recall, and F1-score types is discussed in detail.

Precision

The precision of different networks is presented by heat-maps in Figures 4.11a for each waterbody class. Considering two dimensions ('actual' and 'predicted') of confusion matrix (Fawcett 2006), the precision is calculated as $TP/(TP+FP)$, where TP is the

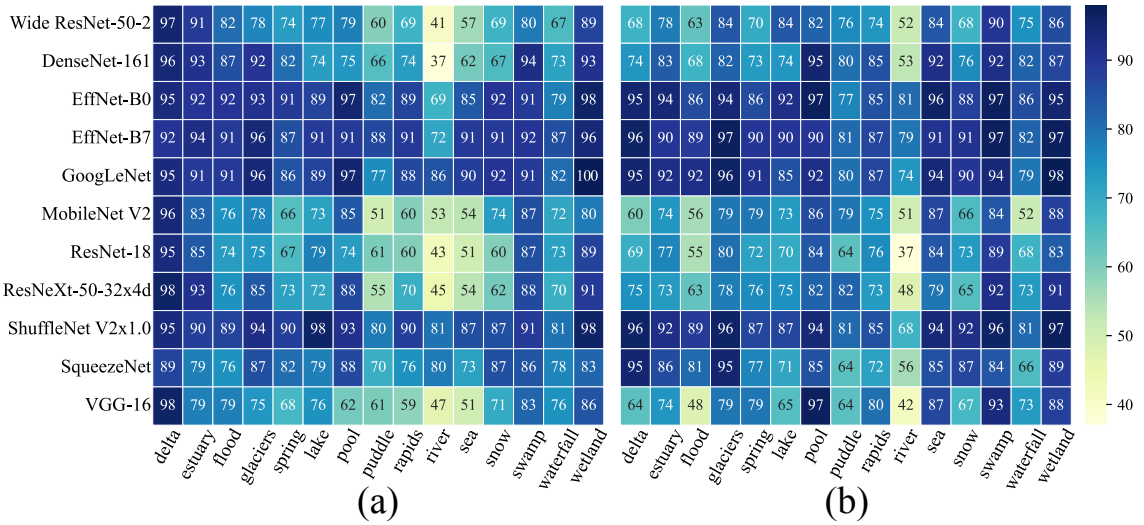


Figure 4.11 Heat-maps of (a) precision and (b) recall performance of all models on each label.

number of true positives and FP the number of false positives (Powers 2020). The precision describes how precise the model is out of the total predicted positive, e.g., how many of those are 'actual' positive. River, puddle and sea classes have the least average precision among all other classes (Figure 4.11a). As it is mentioned before, the flow regime plays an important role in identifying water texture and extracting relevant visual features. As these features are constantly changing in waterbodies such as river and sea, it makes the correct classification task sophisticated for models. In addition to turbulence, other visual features such as inconsistent texture and color over waterbody images cause more complexity. For example, derived patches of sea images near the shoreline represent white, chaotic, and wavy features, while further offshore, sea images have generally calm and still surfaces with darker colors.

Recall

The recall is calculated by $TP/(TP+FN)$, where TP is the number of true positives and FN is the number of false negatives. The recall shows the ability of a classifier to find all the positive samples, i.e., recall calculates the number of actual positives that a model captures by labeling it as a positive (True Positive). River still has the lowest

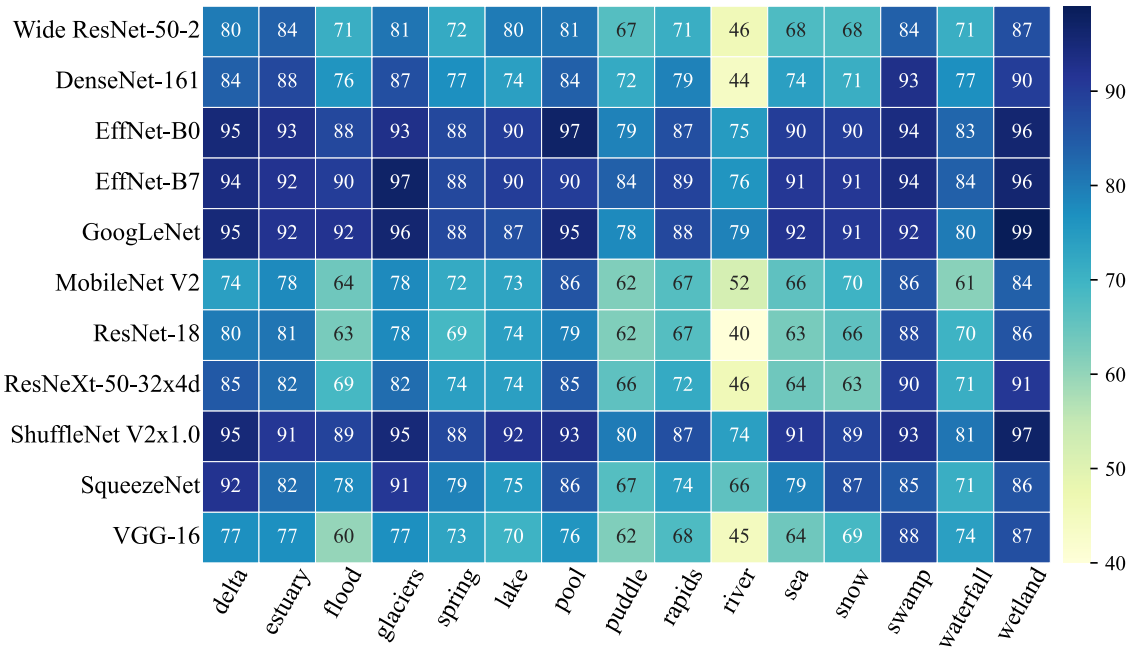


Figure 4.12 Heat-maps of the F-1score performance of all models on each label.

recall values among all other classes (Figure 4.11b). Recall scores of ResNet-18 for river is 37%, while EffNet-B0, EffNet-B7, GoogLeNet present the best results, 81%, 79% and 74% recall scores, respectively. The average of recall score for flood class is about 72%. Three networks including VGG-16, MobileNet V2 and ResNet-18 show the lowest recall scores, all below 60% for flood prediction. ShuffleNet V2 $\times$1.0 has 68% recall score for river and 89% for flood class.

F-1 score

The F-β score is interpreted as the weighted harmonic mean of the precision and recall, where an F-β score reaches its best value at 1 and worst score at 0. The F-β score weights recall more than precision by a factor of β. $\beta = 1.0$ means 'recall' and 'precision' are equally important. Figure 4.12 shows that the lowest F-1 score belongs to river for all models. Six out of eleven models, Wide ResNet-50-2, DenseNet-161, MobileNet V2, ResNet-18, ResNeXt-50-32 $\times$ 4d, and VGG-16, provide the performance measure of less than 60% for river.

4.4 Summary and Conclusion

This study introduced ATeX, the first image dataset for task classification of waterbodies with a specific focus on textural, visual, and intrinsic properties of water affected by surrounding environments. Moreover, Three different vision-based texture analysis techniques including Gabor kernels, LBP, and DL-based model were applied in this study. LBP results showed that water texture analysis is a more difficult problem compared with the task of facial recognition in computer vision. The classification results proved that the high-level image analysis method (i.e., DL-based model), outperformed much better than early vision techniques for water feature extraction. K-Nearest Neighbors results on raw images emphasized on the important role of color and color space for water. Using HSV color space increased the accuracy of classification results to 50%. Among early-vision techniques, LBP offers better results, depending on the parameters which should be adjusted based on the problem. Furthermore, several well-known deep learning models were trained on ATeX train set images. Comparing the performance of these models showed that EffNet-B7, EffNet-B0, GoogLeNet, and ShuffleNet V2×1.0 provided the best results compared to other models. However, narrowing down the search by considering additional factors i.e., training time, the total number of parameters, FLOPs and total memory usage, ShuffleNet V2×1.0 suggested better performance in the shortest time, and thus, selected as the best model trained on ATeX dataset.

In general, digital image processing of water and water-related objects has been a complex task due to the visual challenges which are inherent in water. Water naturally is shapeless and transparent, but it can dominantly be affected by ambient illumination sources, and flow regimes. Since ATeX covers a wide range of complex waterbodies, such as sea, lake, river, swamp, glacier, etc., it poses challenging research questions not only for water resources scientists but also for others who work in fields of Artificial Intelligence and Computer Vision. As the first step, ATeX provides

a new and valuable benchmark for the research community to train or pre-train Deep Learning/ Machine Learning models on this dataset for different water-related applications such as flood detection, drought monitoring, ecological research, etc. ATeX can be used in the future to develop, train and test Deep Neural Network (DNN) architectures for classification tasks which is the backbone of many modern Machine Learning and Deep Learning models. ATeX also enables this opportunity for model developers to design DNN structures solely customized for classification, detection, and semantic segmentation of water and water-related objects in natural and built environments.

Chapter 5

Eye of Horus: A Vision-based Framework for Real-time Water Level Measurement [1]

[1] Erfani, S.M.H., Smith, C., Wu, Z., Shamsabadi, E.A., Khatami, F., Downey, A.R., Imran, J. and Goharian, E., 2023. Submitted to Hydrology and Earth System Sciences (HESS).

5.1 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, Neilson, and Rasmussen 2018). Satellite data and remote sensing can complement in-situ gage data by providing information at a larger spatial scale (Alsdorf, Rodriguez, and Lettenmaier 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, Cervone, and Waters 2014; Goodchild 2007; Howe 2008). To address these limitations and enhance real-time monitoring capabilities, surveillance cameras are investigated here as a new source of data for hydrologic monitoring and flood data collection. However, this requires a significant investment in Computer Vision (CV) and Artificial Intelligence (AI) techniques to develop reliable methods for detecting water in surveillance images and translating that information into numerical data.

Recent advances in CV offer new techniques for processing image data for the quantitative measurements of physical attributes from a site (Forsyth and Ponce

2002). However, there is limited knowledge of how visual information can be used to estimate physical water parameters using CV techniques. Inspired by the principle of the float method, (Tsubaki, Fujita, and Tsutsumi 2011) used different image processing techniques to analyze images captured by closed-circuit television (CCTV) systems installed for surveillance purposes to measure the flow rate during flood events. In another example, (Kim, Han, and Hahn 2011) proposed a method for measuring water level by detecting the borderline between a staff gauge and the surface of water based on image processing of the captured image of the staff gage installed in the middle of the river. As the use of images for environmental monitoring becomes more popular, several studies have investigated the source and magnitude of errors common in image-based measurement systems, such as the effect of image resolution, lighting effects, perspective, lens distortion, water meniscus, and temperature changes (Elias et al. 2020; Gilmore, Birgand, and Chapman 2013). Furthermore, proposed solutions to resolve difficulties originating from poor visibility have been developed to better identify readings on staff gages (Zhang et al. 2019). Recently, Deep Learning (DL) has become prevalent across a wide range of disciplines, particularly in applied sciences such as CV and engineering.

DL-based models have been utilized by the water resources community to determine the extent of water and waterbodies visible in images captured by surveillance camera systems. These models can estimate the water level (Pally and Samadi 2022). In a similar vein, (Vitry et al. 2019) employed a DL-based approach to identify floodwater in surveillance footage and introduced a novel qualitative flood index, SOFI, to determine water level fluctuations. SOFI was calculated by taking the aspect ratio of the area of the water surface detected within an image to the total area of the image. However, these types of methods, which make prior assumptions and estimate water level fluctuation roughly, cannot serve as a vision-based alternative for measuring streamflow characteristics. More systematic studies adopted photogrammetry to re-

construct a high-quality 3D model of the environment with a high spatial resolution to have a precise estimation of real-world coordination while measuring streamflow rate and stage. For example, (Eltner et al. 2018; Eltner et al. 2021) introduced a method based on Structure from Motion (SfM), and photogrammetric techniques, to automatically measure the water stage using low-cost camera setups.

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

This paper presents a vision-based framework for measuring water levels from time-lapse images. The proposed framework introduces a novel approach by utilizing the iPhone LiDAR sensor as a laser scanner, which is commonly available on consumer-grade devices, for scanning and constructing a 3D point cloud of the region of interest. During the data collection phase, time-lapse images and ground truth water level values were collected using an embedded camera and ultrasonic sensor. The

water extent in the captured images was determined automatically using semantic segmentation DL-based models. For the first time, the performance of three different state-of-the-art DL-based approaches, including Convolutional Neural Networks (CNN), hybrid CNN-Transformer, and Transformers-Multilayer Perceptron (MLP), was evaluated and compared. CV techniques were applied for camera calibration, pose estimation of the camera setup in each deployment, and 3D-2D reprojection of the point cloud onto the image plane. Finally, K-Nearest Neighbors (KNN) was used to find the nearest projected (2D) point cloud coordinates to the water line on the river banks, for estimating the water level in each time-lapse image.

5.2 Deep Learning Architectures

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

CNN-based models were designed to imitate the recognition system of primates while possessing different network designs such as low-resolution representations learning (Shamsabadi, Xu, and Costa 2022; Long, Shelhamer, and Darrell 2015; Chen et al. 2017a), high-resolution representations recovering (Badrinarayanan, Handa, and Cipolla 2015; Noh, Hong, and Han 2015; Lin et al. 2017), contextual aggregation schemes (Yuan and Wang 2018; Zhao et al. 2017; Yuan, Chen, and Wang 2020), feature fusion and refinement strategy (Lin et al. 2017; Huang et al. 2019; Li et al. 2019; Zhu et al. 2019; Fu et al. 2019). CNN-based models follow local to global features in different layers of the forward pass, which used to be thought of as a general intuition of the human recognition system. In this system, objects are recognized through the

analysis of texture and shape-based clues– local and global representations and their relationship in the entire field of view. Recent research, however, shows significant differences exist between the visual behavioral system of humans and CNN-based models (Geirhos et al. 2018a), and reveal higher sensitivity of the visual systems in humans to global features rather than local ones (Zheng et al. 2018). This fact drew attention to models that focus on the global context in their architectures.

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

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

5.3 Study Area

In order to evaluate the performance of the proposed framework for measuring the water levels in rivers and channels, a time-lapse camera system has been deployed at Rocky Branch, South Carolina. This creek is approximately 6.5 km long and collects stormwater from the University of South Carolina campus and the City of Columbia. Rocky Branch is subjected to rapid changes in water flow and discharges into the Congaree River (Morsy et al. 2016). The observation site is located within the University of South Carolina campus behind 300 Main Street. An Apple iPhone 13 Pro LiDAR sensor was used to scan the region of interest (see Figure 5.1a). Although there is no official information about the technology and hardware specifications, (Gollob et al. 2021) reports the LiDAR module operates at the 8XX nm wavelength and consists of an emitter (Vertical Cavity Surface-Emitting Laser with Diffraction Optics Element, VCSEL DOE) and a receptor (Single Photon Avalanche Diode array-based Near Infrared Complementary Metal Oxide Semiconductor image sensor, SPAD NIR CMOS) based on direct-time-of-flight technology. Comparisons between the Apple LiDAR sensor and other types of laser scanners including hand-held, industrial, and terrestrial have been conducted by several recent studies (Mokroš et al. 2021; Vogt, Rips, and Emmelmann 2021). (Gollob et al. 2021) tested and reported the performance of a set of eight different scanning apps, and found three applications including 3D Scanner App, Polycam and SiteScape suitable for actual practice tests. The objective of this study is not the evaluation of the iPhone LiDAR sensor and app performance. Therefore, the 3D Scanner App (LABS n.d.) was used with the following settings: confidence = high, range = 5.0 m, masking = none, and resolution = 5 mm, for scanning and 3D reconstruction processing. The scanned 3D point cloud is shown in Figure 5.1b.

As the LiDAR scanner settings were set at the highest level of accuracy and computational demand, scanning the whole region of interest at the same time was not

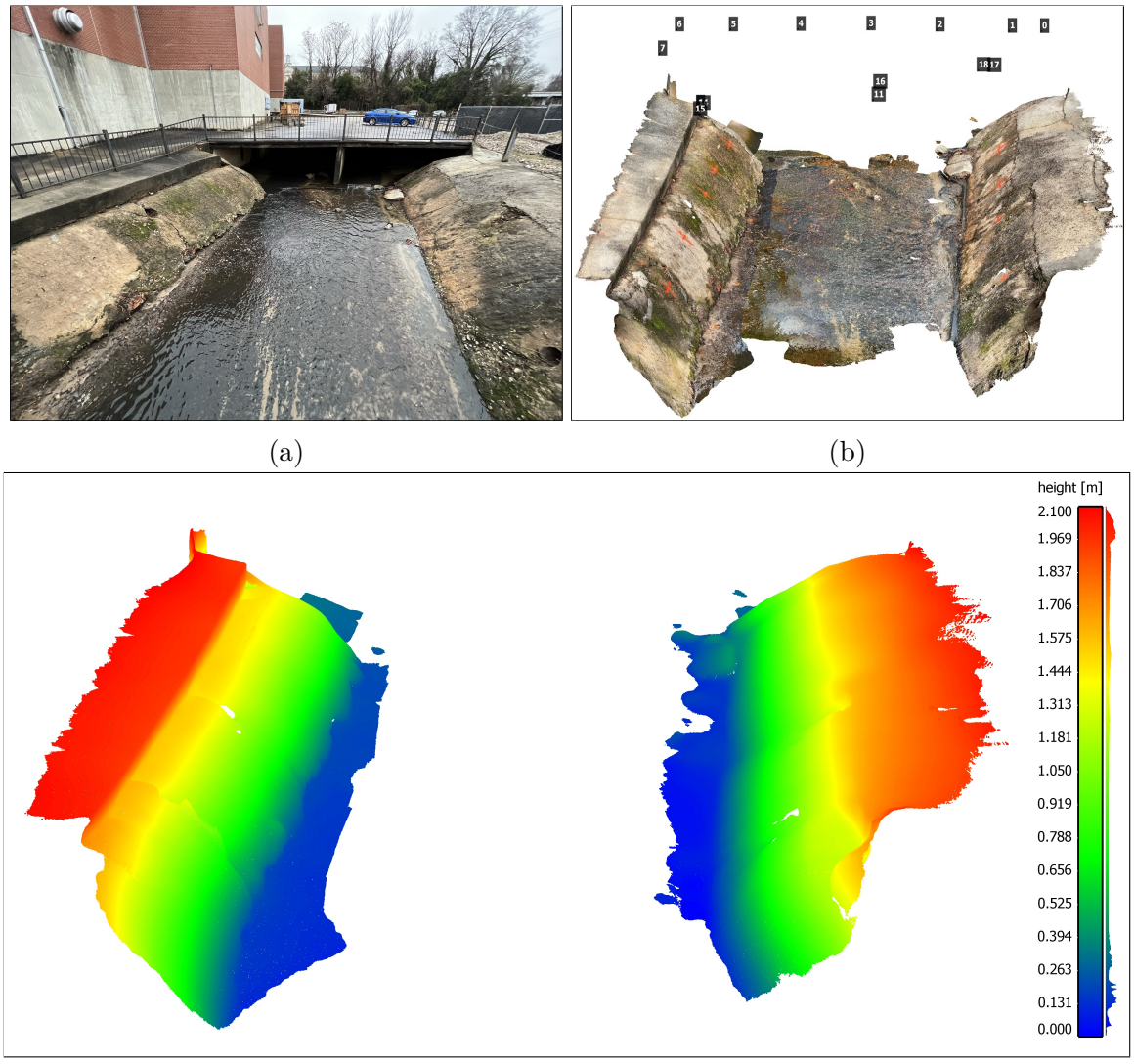
possible. So, the experimental region was divided into several sub-regions and scanned in multi-step. In order to assemble the sub-region LiDAR scans, several GCPs were considered in the study area. These GCPs were measured by a total station (Topcon GM Series). Moreover, 13 AruCo markers were installed for estimating extrinsic camera parameters in each setup deployment. Since it was not possible to accurately measure the real-world coordination of AruCo markers by the LiDAR scanner, the coordinates of the top-left corner of markers were also measured by the surveying total station. The 3D point cloud scanned for each sub-region was transformed into the total station coordinate system, and the real-world coordinates of ArUco markers were appended to the 3D point cloud for the following analyses.

5.4 Methodology

This study introduces the Eye of Horus, a vision-based framework for hydrologic monitoring and real-time water level measurements in bodies of water. The proposed framework includes three main components. The first step is designing two deployable setups for data collection. These setups consist of a programmable time-lapse camera run by Raspberry Pi and an ultrasonic sensor run by Arduino. After collecting data, the first phase (Module 1) involves configuring and training DL-based models for semantic segmentation of water in the captured images. In the second phase (Module 2), CV techniques for camera calibration, spatial resection, and calculating projection matrix are discussed. Finally, in the third phase (Module 3), an ML-based model uses the information achieved by CV models to find the relationships between real-world coordinates of water level in the captured images (see Figure 5.2).

5.4.1 Data Acquisition

Two different single-board computers (SBC) were used in this study, Raspberry Pi (Zero W) for capturing time-lapse images of a river scene, and Arduino (Nano 3.x)



(a) (b)

(c)

Figure 5.1 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).

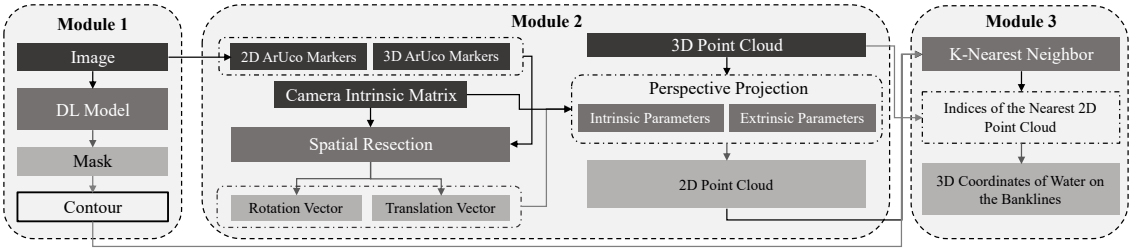


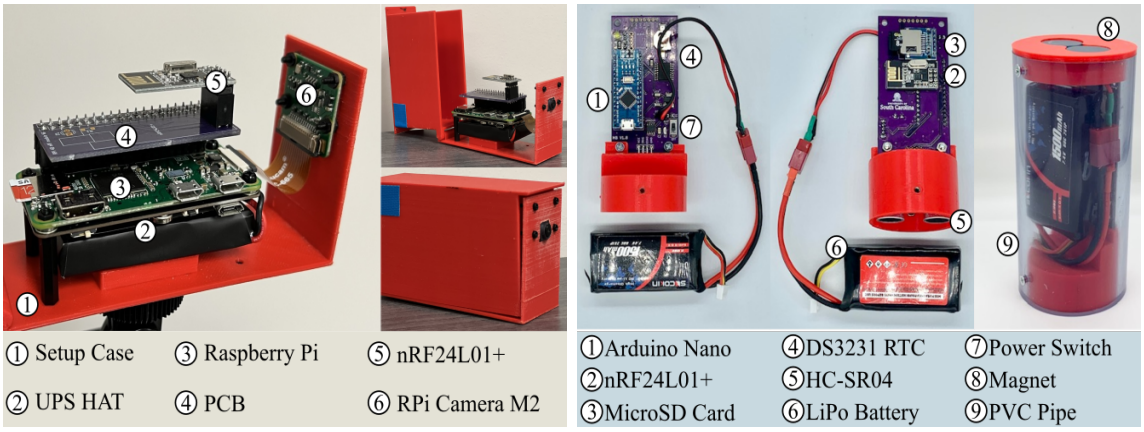
Figure 5.2 The Eye of Horus workflow includes three main modules starting from processing images captured by the time-lapse camera to estimating water level by projecting the waterline on river banks using CV techniques.

for measuring water level as the ground truth data. These devices were designed to communicate with each other, i.e., to trigger the other to start or stop recording. During capturing time-lapse images, the Pi camera device triggers the ultrasonic sensor for measuring the corresponding water level. The camera device is equipped with the Raspberry Pi Camera Module 2 which has a Sony IMX219 8-megapixel sensor. This sensor is able to capture an image size of 4,256 × 2,832 pixels. However, in this study, the image resolution was set to 1,920 × 1,440 pixels to balance image quality and computational cost in subsequent image processing steps. This setup is also equipped with a 1200 mAh UPS lithium battery power module to provide uninterrupted power to the Pi SBC (see Figure 5.3a).

The Arduino-based device records the water level. The design is based on an unmanned aerial vehicle (UAV) deployable sensor created by (Smith et al. 2022). The nRF24L01+ single-chip 2.4 GHz transceiver allows the Arduino and Raspberry Pi to communicate via radio frequency (RF). The chip 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 Arduino device and provides a contactless water level measurement. Two permanent magnets at the top of the housing attach to a ferrous structure and allow the ultrasonic sensor to be suspended up to 14 feet over the surface of the water. The device also includes a microSD card module and DS3231 real-time clock, which enable

data logging and storage on-device as well as transmission. The device is powered by a rechargeable 7.4V 1500 mAh lithium polymer battery (see Figure 5.3b).

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



(a) (b)

Figure 5.3 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.

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

5.4.2 Deep Learning Model for Water Segmentation

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

TransUNet, the second model, is a U-shaped architecture that employs a hybrid of CNN and Transformers as the encoder to leverage both the local and global contexts for precise localization and pixel-wise classification (Chen et al. 2021). In the encoder part of the network, CNN is first used as a feature extractor to generate a feature map for the input image, which is then fed into Transformers to extract long-range dependencies. The resulting features are upsampled in the decoding path and combined with detailed high-resolution spatial information skipped from the CNN to make estimations on each pixel of the input image.

SegFormer, the third model, unifies a novel hierarchical Transformer, which does not require the positional encodings used in standard Transformers, and MultiLayer Perceptron (MLP) performs efficient segmentation (Xie et al. 2021). The hierarchical Transformer introduced in the encoder of this architecture gives the model the attention ability to multiscale features (high-resolution fine and low-resolution coarse

information) in the spatial input without the need for positional encodings that may adversely affect a models performance when testing on a different resolution from training. Moreover, unlike other segmentation models that typically use deconvolutions in the decoder path, a lightweight MLP is employed as the decoder of this network that inputs the features extracted at different stages of the encoder to generate a prediction map faster and more efficiently. Two different variants, including SegFormer-B0 and SegFormer-B5, were applied in this study. The configuration of the models implemented in this study is elaborated in Table 5.1. The total number of parameters (Params), occupied memory size on GPU (Total Size), and input image size (Batch Size) are reported in Million (M), Megabyte (MB), and Batch size×Height×Width×Channel (*B, H, W, C*) respectively. BCE and BE stand for Binary Cross Entropy and Cross Entropy, respectively.

Table 5.1 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	2×500×500×3	BCE	SGD	2.50E-04
TransUNet	20.1	6,017	2×448×448×3	CE + Dice	SGD	2.50E-04
SegFormer-B0	3.7	2,217	2×512×512×3	CE	AdamW	6.00E-05
SegFormer-B5	82.0	27,666	2×1024×1024×3	CE	AdamW	6.00E-05

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

scaling, and random cropping.

5.4.3 Projective Geometry

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

To estimate the camera intrinsic parameters, OpenCV built-in was applied for camera calibration using a 2D checkerboard (Bradski 2000). Intrinsic parameters are specific to a camera. The focal length (f_x, f_y) and optical centers (c_x, c_y) can be used to create a camera matrix. The camera matrix is unique to a specific camera, so once calculated, it can be reused on other images taken by the same camera (Equation 5.1). It is expressed as a 3×3 matrix:

$$\text{camera matrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{5.1}$$

The camera extrinsic parameters were determined using the pose estimation problem which consists in solving for the rotation, and translation that minimizes the reprojection error from 2D-3D point correspondences (Marchand, Uchiyama, and Spindler 2015). For this purpose, the iterative method was applied which is based

on a Levenberg-Marquardt optimization. In this task the function finds such a pose that minimizes reprojection error, that is the sum of squared distances between the observed projections "image point" and the projected "object points." The initial solution for non-planar 3D object points needs at least six points and uses the Direct Linear Transformation (DLT) algorithm.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \overbrace{\begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}^{\mathbf{K}} \overbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}}^{[\mathbf{R}|\mathbf{t}]} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{5.2}$$

Equation 5.2 represents "Projection Matrix" consisting of two parts– the intrinsic matrix ($\mathbf{K}$) that contains the intrinsic parameters and the extrinsic matrix ($[\mathbf{R} \mid \mathbf{t}]$) that is a combination of 3×3 rotation matrix $\mathbf{R}$ and a 3×1 translation $\mathbf{t}$ vector.

2D points are represented with ArUco markers' pixel coordinates on the 2D image plane, and corresponding 3D object points are measured by the total station. Having at least six 3D-2D point correspondences, the spatial position and orientation of the camera can be estimated for each setup deployment. After retrieving all the necessary parameters, a full-perspective camera model can be generated. Using this model, the 3D point cloud is projected on the 2D image plane. The projected (2D) point cloud can represent 3D real-world coordinates of the nearest 2D pixel correspondence on the image plane.

5.4.4 Machine Learning for Image Measurements

Using the projection matrix, the 3D point cloud is projected on the 2D image plane (see Figure 5.4). The projected (2D) point cloud is intersected with the water line pixels, the output of the DL-based model (Module 1), to find the nearest point cloud coordinate. To achieve this objective, we utilize the K-Nearest Neighbors (KNN)

algorithm. Notably, the indices of the selected points remain consistent for both the 3D point cloud and the projected (2D) correspondences. As a result, by utilizing the indices of the chosen projected (2D) points, the corresponding real-world 3D coordinates can be retrieved.

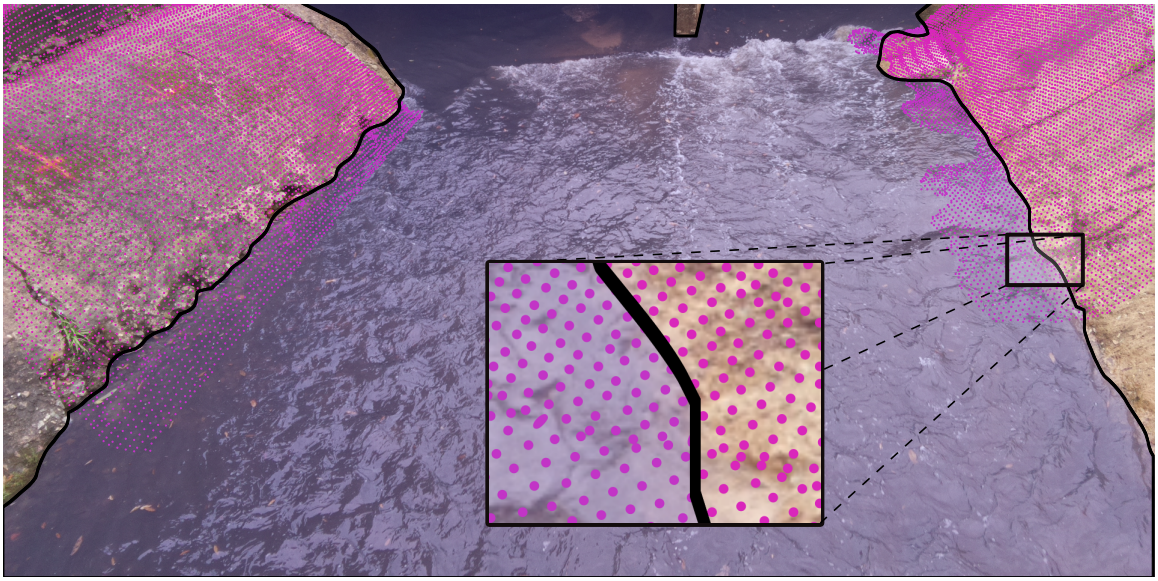


Figure 5.4 KNN is used to find the nearest projected (2D) point cloud (magenta dots) to the water line (black line) on the image plane.

5.5 Results and Discussion

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

5.5.1 DL-based Models Results

The performance of DL-based models for the task of semantic segmentation is evaluated and compared in this section. Since the proposed dataset includes just two classes, “river” and “non-river”, “non-river” was omitted from the evaluation process, and the performance of models is only reported for the “river” class of the test set. The class-wise intersection over union (IoU) and the per-pixel accuracy (ACC) were

considered the main evaluation metrics in this study. According to Table 5.2, both variants of SegFormer– SegFormer-B0, and SegFormer-B5– outperform other semantic segmentation networks on the test set. Considering the models' configurations detailed in Table 5.1, SegFormer-B0 can be considered the most efficient DL-based network, as it is comprised of only 3.7 M trainable parameters and occupies just 2,217 Megabytes of GPU ram during training. In Figure 5.5, four different visual representations of the models' performance on the validation set of the proposed dataset are presented. Since the water level is estimated by intersecting the water line on river banks with the projected (2D) point cloud, precise delineation of the water line is of utmost importance to achieve better results in the following steps. This means that estimating the correct location of the water line on creek banks in each time-lapse image plays a more significant role than performance metrics in this study. Taking the quality of water line detection into account and based on the visual representations shown in Figure 5.5, SegFormers' variants still outperform DL-based approaches. In this regard, a comparison of PSPNet and TransUNet showed that PSPNet can delineate the water line more clearly, while the segmented area is more integrated for TransUNet outputs.

Table 5.2 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%

CNNs are typically limited by the nature of their convolution operations, leading to architecture-specific issues such as locality (Geirhos et al. 2018b). Consequently, CNN-based models may achieve high accuracy on training data, but their performance can decrease considerably on unseen data. Additionally, compared to Transformer-based networks, they perform poorly at detecting semantics that requires combining

long- and short-range dependencies. Transformers can relax the biases of DL-based models inducted by Convolutional operations, achieving higher accuracy in localization of target semantics and pixel-level classification with lower fluctuations in varied situations through the leverage of both local and global cues (Naseer et al. 2021). Yet, various transformer-based networks may perform differently depending on the targeted task and the network's architecture. TransUNet adopts Transformers as part of its backbone; however, Transformers generate single-scale low-resolution features as output (Xie et al. 2021), which may limit the accuracy when multi-scale objects or single objects with multi-scale features are segmented. The problem of producing single-scale features in standard Transformers is addressed in SegFormer variants through the use of a novel hierarchical Transformer encoder (Xie et al. 2021). This approach has resulted in human-level accuracy being achieved by Segformer-B0 and -B5 in the delineation of the water line, as shown in Figure 5.5. The predicted masks are in satisfactory agreement with the manually annotated images.

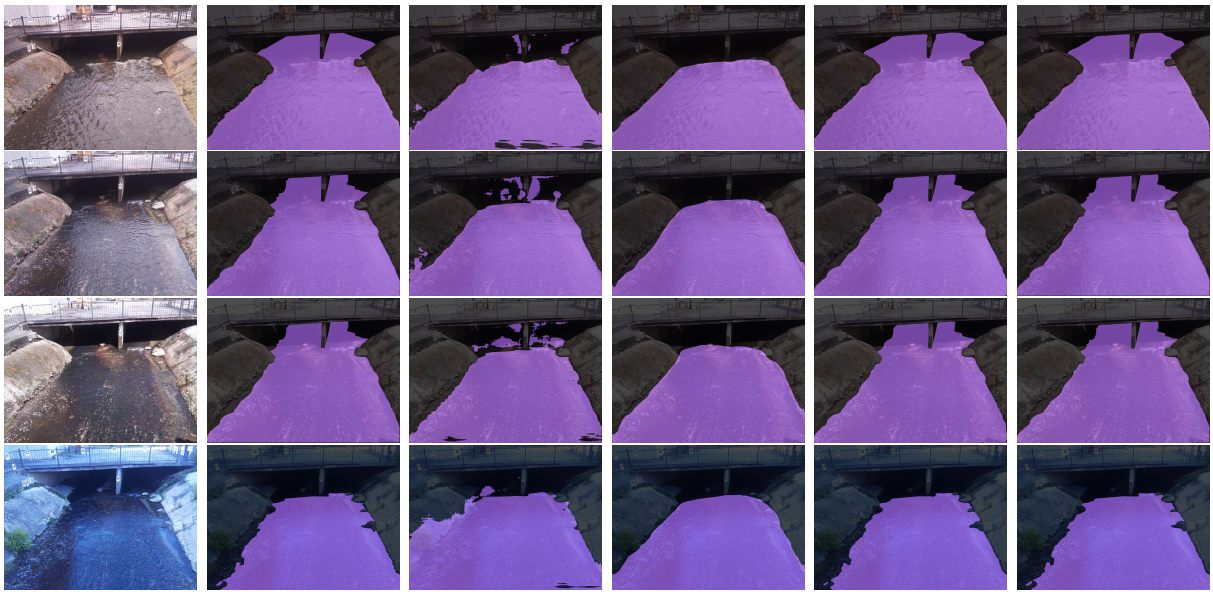


Figure 5.5 Visual representations of different DL-based image segmentation approaches on the validation dataset.

5.5.2 Water Level Estimation

This section reports the framework performance based on several deployments in the field. The performance results are separately shown for the left and right banks and compared with ultrasonic sensor data as the ground truth. The ultrasonic sensor was evaluated previously that documented an average distance error of 6.9 mm (Smith et al. 2022). Four different efficiency criteria including coefficient of determination (R^2), Nash-Sutcliffe Efficiency (NSE), Root Mean Square Error (RMSE), and Percent bias (PBIAS) are reported in Table 5.3. R^2, as the most representative metric, emphasizes how much of the observed dispersion can be explained by the prediction. However, if the model systematically over- or under-estimates the results, R^2 will still be close to 1.0 as it only takes dispersion into account (Krause, Boyle, and Bäse 2005). NSE, a traditional metric used in hydrology is also used to summarize model performance. NSE normalizes model performance into an interpretable scale and is commonly used to differentiate between 'good' and 'bad' models (Knoben, Freer, and Woods 2019). RMSE represents the square root of the average of squares of the errors, the differences between predicted values and observed values. The PBIAS of estimated water level, compared against the ultrasonic sensor data was also used to show where the two estimates are close to each other and where they significantly diverge (Lin et al. 2020).

The setup was deployed on several rainy days. In addition to Table 5.3, the results of each deployment are visually demonstrated in Figure 5.6. The scatter plots show the relationships between the ground truth data (measured by the ultrasonic sensor), and the banks of the river. The scatter plots visually present whether the camera readings overestimate or underestimate the ground truth data. Moreover, the time-series plot of water level is shown for each deployment separately. A hydrograph, showing changes in the water level of a stream over time can be a useful tool for demonstrating whether camera readings can satisfactorily capture the response of a

Table 5.3 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

catchment area to rainfall. The proposed framework can be evaluated in terms of its ability to accurately track and identify important characteristics of a flood wave, such as the rising limb, peak, and recession limb.

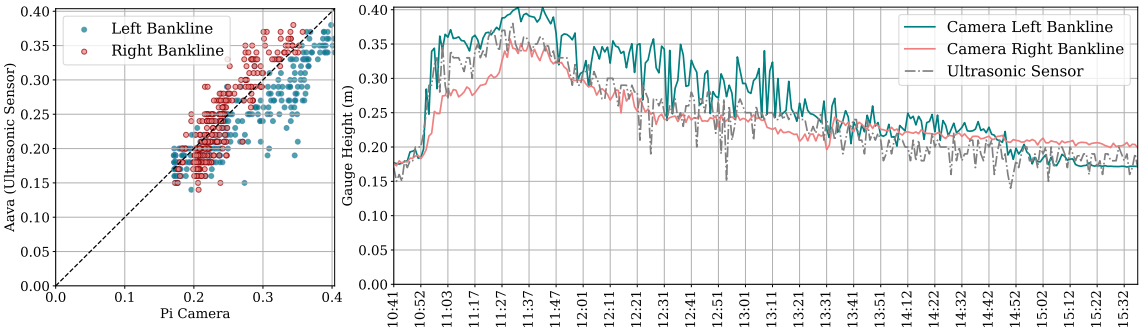
The first deployment was done on Aug 17, 2022 (see Figure 5.6a). The initial water level of the base flow and parts of the rising limb were not captured in this deployment. Table 5.3 shows that the performance results of the right bank camera readings are better than those of the left bank. R^2 for both banks was about 0.80 showing a strongly related correlation between the water level estimated by the framework and ground truth data. Figure 5.6a shows how the left and right bank camera readings perform during the rising limb; the right bank camera readings still underestimated the water level during this time frame, and during the recession limb, the left bank camera readings overestimated the water level. However, the hydrograph plot shows that both left and right bank camera readings were able to capture the peak water level.

The second deployment was done on Aug 19, 2022. In this deployment, all segments of the hydrograph were captured. According to Table 5.3, the performance of the right bank camera readings was better than the left bank one; more than 0.95

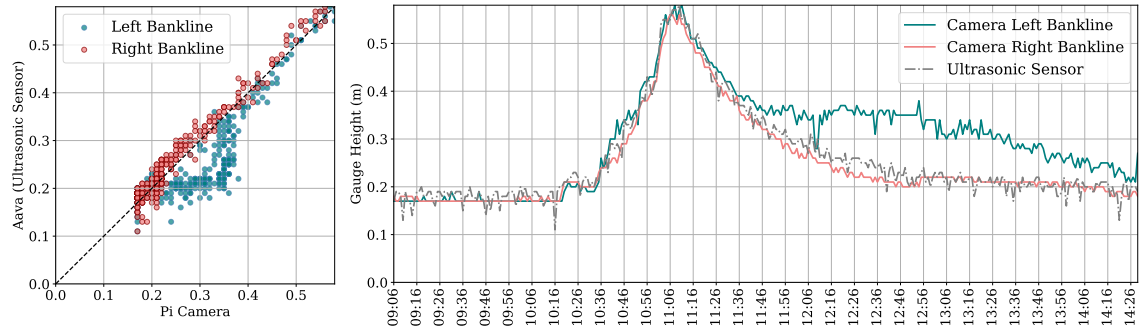
was reported for R^2 and NSE of the right bankline. Figure 5.6b shows during the rising limb and crest segment both banks estimated the water level similar to ground truth. During the recession limb, the right bank water level estimation kept coincident with ground truth, while the left bank overestimated the water level. The third deployment was on Aug 25, 2022. This time water level of the recession limb and the following base flow were captured (see Figure 5.6c). The right bank camera readings with R^2 of 0.89 performed better than the left bank. This time, left bank camera readings underestimated the water level over the recession limb, but during the following base flow, the water level was estimated correctly by cameras on both banks.

The results indicate that the right bank camera readings performed better than the left bank. Further investigation of the field conditions revealed that stream erosion had a more significant impact on the concrete surface of the left bank, resulting in patches and holes that were not scanned by the iPhone LiDAR. As a result, the KNN algorithm used to find the nearest (2D) point cloud coordinates to the water line could not accurately represent the corresponding real-world coordinates of these locations. Figure 5.7 shows a box plot and scatter plot of the estimated water level for a time-lapse image captured at 13:29 on Aug 19, 2022. The patches and holes on the left bank surface caused instability in water level estimation for the region of interest. The box plot of the left bank (Cam-L-BL) was taller than that of the right bank (Cam-R-BL), indicating that the estimated water level was spread over larger values in the left bank due to the presence of these irregularities.

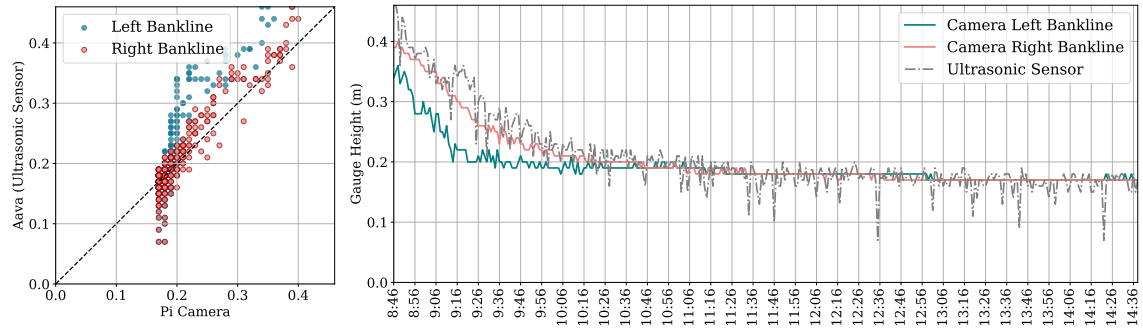
After analyzing the initial results, the deployable setups were modified to enhance the quality of data collection. The programming code of the Arduino device, Aava, was modified to measure five different records for water level, each time it is triggered by the camera device, Beena, and transmit the average distance to the Raspberry Pi device. This modification decreased the number of noise spikes in the measured data



(a)



(b)



(c)

Figure 5.6 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.

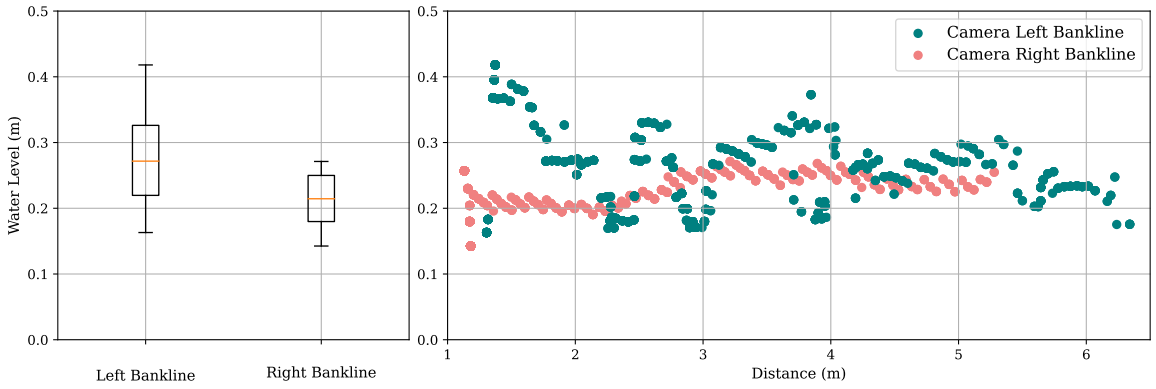


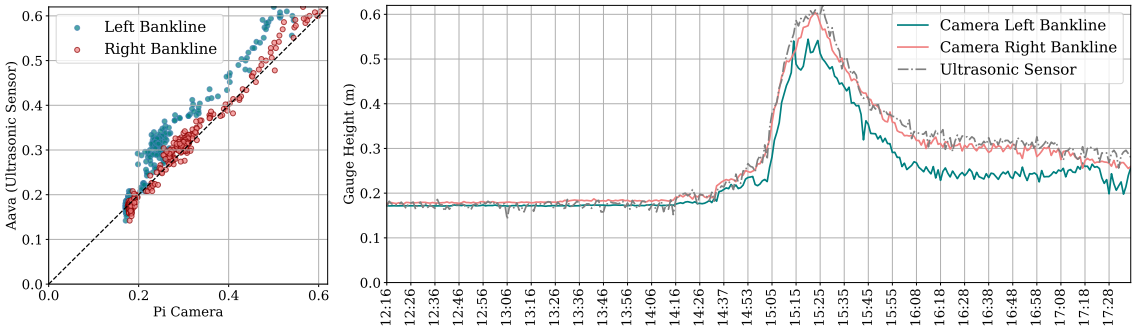
Figure 5.7 Water level fluctuation along both left and right banks for the flow regime for an image captured at 13:29 on Aug 19, 2022.

and allowed a better comparison between camera readings and ground truth data. The case of the camera device, Beena, was redesigned to protect the single board against rain without requiring an umbrella which makes the camera setup unstable in stormy weather and causes a decrease in the precision of measurements. Moreover, an opening is incorporated into the redesigned case to connect an external power bank to enhance the run time. Finally, the viewpoint of the camera was subtly shifted to the right to adjust the share of the river banks on the camera's field of view.

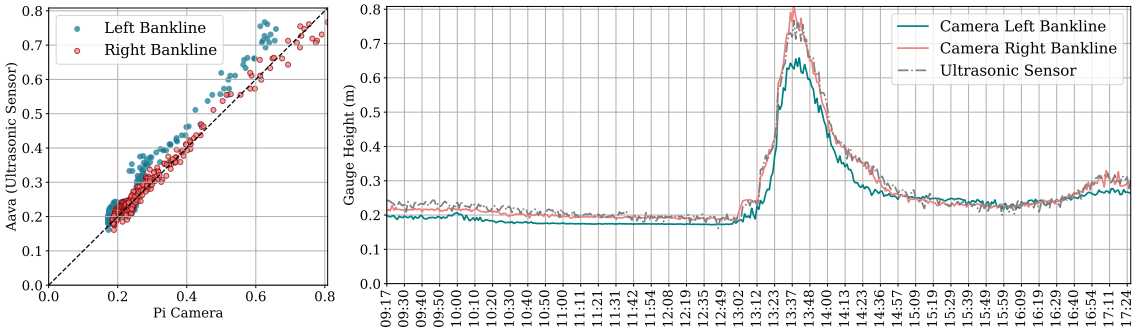
The results of the deployments on Nov 10, 2022, and Nov 11, 2022, demonstrate that modifications to the setup have significantly improved the results of the left bank (as shown in Table 5.3). NSE improved from approximately 0.55 for the first three setup deployments to over 0.80 for the modified deployments. Figure 5.8 shows the setup performances during all segments of the flood wave. The peaks were captured by the right bankline on both deployment dates, and there was no effect of noisy spikes on either camera readings or ground truth data. However, the right bank images still underestimated the water level during the rainstorms.

5.6 Conclusion

This study introduced Eye of Horus, a vision-based framework for hydrologic monitoring and measuring real-time water-related parameters, e.g., water level, from surveil-



(a)



(b)

Figure 5.8 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.

lance images captured during flood events. Time-lapse images and real water level correspondences were collected by Raspberry Pi camera and Arduino HC-SR05 ultrasonic sensor, respectively. Moreover, Computer Vision and Deep Learning techniques were used for semantic segmentation of water surface within the captured images and for reprojecting the 3D point cloud constructed with an iPhone LiDAR scanner, on the (2D) image plane. Eventually, the K-Nearest Neighbor algorithm was used to intersect the projected (2D) point cloud with the water line pixels extracted from the output of the Deep Learning model, to find the real-world 3D coordinates.

A vision-based framework offers a new alternative to current hydrologic data collection and real-time monitoring systems. Hydrological models require geometric information for estimating discharge routing parameters, stage, and flood inundation maps. However, determining bankfull characteristics is a challenge due to natural or

anthropogenic down-cutting of streams. Using visual sensing, stream depth, water velocity, and instantaneous streamflow at bankfull stage can be reliably measured.

Chapter 6

Conclusion

During flash or nuisance flooding in urban areas, real-time visual monitoring of flood events can assist decision-makers and local authorities to take better hazard reduction actions, specifically for public disaster warnings. Conventional gauge sensing systems, however, provide the water level data only in one spatial dimension which cannot accurately represent the actual runoff-land interactions. Consequently, related authorities cannot obtain sufficient visual field information for disaster control and hazard reduction. Surveillance imagery networks, on the other hand, provide spatial dynamics of the surface water extent in the monitored region. Using the vision-based framework of this dissertation, such systems will be capable to provide disaster prevention agencies with actual field information such as water stage and discharge for over-bank flow states. Such information can be used for determining the water fluctuation and measuring its elevation and flood intrusion with respect to real-world coordinates which will be more helpful to real-time flood inundation modeling and disaster-relieving operations.

Bibliography

Adam, Elhadi et al. (2014). "Land-use/cover classification in a heterogeneous coastal landscape using RapidEye imagery: evaluating the performance of random forest and support vector machines classifiers". In: *International Journal of Remote Sensing* 35.10, pp. 3440–3458.

Adnan, Rana Muhammad et al. (2019). "Daily streamflow prediction using optimally pruned extreme learning machine". In: *Journal of Hydrology* 577, p. 123981.

Ahonen, Timo, Abdenour Hadid, and Matti Pietikäinen (2004). "Face recognition with local binary patterns". In: *Eur. Conf. Comput. Vis.* Springer, pp. 469–481.

Alsdorf, Douglas, Ernesto Rodriguez, and Dennis Lettenmaier (2007). "Measuring surface water from space". In: *Reviews of Geophysics* 45.2.

Anusree, K and KO Varghese (2016). "Streamflow prediction of Karuvannur River Basin using ANFIS, ANN and MNLR models". In: *Procedia Technology* 24, pp. 101–108.

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

Bai, Shaojie, J Zico Kolter, and Vladlen Koltun (2018). "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling". In: *arXiv preprint arXiv:1803.01271*.

Bernal, Jorge et al. (2017). "Comparative validation of polyp detection methods in video colonoscopy: results from the MICCAI 2015 endoscopic vision challenge". In: *IEEE Transactions on Medical Imaging* 36.6, pp. 1231–1249.

Bjerklie, David M et al. (2003). "Evaluating the potential for measuring river discharge from space". In: *Journal of Hydrology* 278.1-4, pp. 17–38.

Bradski, G. (2000). "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools*.

Caesar, Holger, Jasper Uijlings, and Vittorio Ferrari (2018). "Coco-stuff: Thing and stuff classes in context". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1209–1218.

Cao, Yue et al. (2019). "Gcnet: Non-local networks meet squeeze-excitation networks and beyond". In: *Int. Conf. Comput. Vis. Worksh.*

Cervone, Guido et al. (2016). "Using Twitter for tasking remote-sensing data collection and damage assessment: 2013 Boulder flood case study". In: *International Journal of Remote Sensing* 37.1, pp. 100–124.

Cervone, Guido et al. (2017). "Using social media and satellite data for damage assessment in urban areas during emergencies". In: *Seeing cities through big data*. Springer, pp. 443–457.

Charoenporn, Pattama (2017). "Reservoir inflow forecasting using ID3 and C4. 5 decision tree model". In: *2017 3rd IEEE International Conference on Control Science and Systems Engineering (ICCSSE)*. IEEE, pp. 698–701.

Chen, Jieneng et al. (2021). "Transunet: Transformers make strong encoders for medical image segmentation". In: *arXiv preprint arXiv:2102.04306*.

Chen, Liang-Chieh et al. (2017a). "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 40.4, pp. 834–848.

Chen, Liang-Chieh et al. (2017b). "Rethinking atrous convolution for semantic image segmentation". In: *arXiv preprint arXiv:1706.05587*.

Cheng, M et al. (2020). "Long lead-time daily and monthly streamflow forecasting using machine learning methods". In: *Journal of Hydrology* 590, p. 125376.

Chiang, Yen-Ming, Li-Chiu Chang, and Fi-John Chang (2004). "Comparison of static-feedforward and dynamic-feedback neural networks for rainfall–runoff modeling". In: *Journal of hydrology* 290.3-4, pp. 297–311.

Chollet, François (2017). "Xception: Deep learning with depthwise separable convolutions". In: *IEEE Conf. Comput. Vis. Pattern Recog.* Pp. 1251–1258.

Chow, Ven Te, David R Maidment, and W Larry (1988). "Applied Hydrology". In: *International edition, MacGraw-Hill, Inc* 149.

Chow, Ven Te, David R. Maidment, and Mays Larry W. (1964). *Handbook of applied hydrology*. McGraw-Hill.

Cigizoglu, Hikmet Kerem (2005). "Application of generalized regression neural networks to intermittent flow forecasting and estimation". In: *Journal of Hydrologic Engineering* 10.4, pp. 336–341.

Collobert, Ronan and Samy Bengio (2001). "SVMTorch: Support vector machines for large-scale regression problems". In: *Journal of machine learning research* 1.Feb, pp. 143–160.

Cordts, Marius et al. (2016). "The cityscapes dataset for semantic urban scene understanding". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223.

Deng, Jia et al. (2009). "Imagenet: A large-scale hierarchical image database". In: *IEEE Conf. Comput. Vis. Pattern Recog.* Ieee, pp. 248–255.

Dibike, Yonas B et al. (2001). "Model induction with support vector machines: introduction and applications". In: *Journal of Computing in Civil Engineering* 15.3, pp. 208–216.

Dosovitskiy, Alexey et al. (2020). "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929*.

Duan, Shiheng, Paul Ullrich, and Lele Shu (2020). "Using convolutional neural networks for streamflow projection in california". In: *Frontiers in Water* 2, p. 28.

Elias, Melanie et al. (2020). "Assessing the influence of temperature changes on the geometric stability of smartphone-and raspberry Pi cameras". In: *Sensors* 20.3, p. 643.

Eltner, Anette and Danilo Schneider (2015). "Analysis of different methods for 3D reconstruction of natural surfaces from parallel-axes UAV images". In: *The Photogrammetric Record* 30.151, pp. 279–299.

Eltner, Anette et al. (2016). "Image-based surface reconstruction in geomorphometry–merits, limits and developments". In: *Earth Surface Dynamics* 4.2, pp. 359–389.

Eltner, Anette et al. (2018). "Automatic image-based water stage measurement for long-term observations in ungauged catchments". In: *Water Resources Research* 54.12, pp. 10–362.

Eltner, Anette et al. (2021). "Using deep learning for automatic water stage measurements". In: *Water Resources Research* 57.3, e2020WR027608.

Erdal, Halil Ibrahim and Onur Karakurt (2013). "Advancing monthly streamflow prediction accuracy of CART models using ensemble learning paradigms". In: *Journal of Hydrology* 477, pp. 119–128.

Erfani, Seyed Mohammad Hassan and Erfan Goharian (2023). "Vision-based texture and color analysis of waterbody images using computer vision and deep learning techniques". In: *Journal of Hydroinformatics* 25.3, pp. 835–850.

Erfani, Seyed Mohammad Hassan et al. (2022). "ATLANTIS: A benchmark for semantic segmentation of waterbody images". In: *Environmental Modelling & Software* 149, p. 105333.

Everingham, Mark et al. (2010). "The pascal visual object classes (voc) challenge". In: *International journal of computer vision* 88.2, pp. 303–338.

Fatichi, Simone et al. (2016). "An overview of current applications, challenges, and future trends in distributed process-based models in hydrology". In: *Journal of Hydrology* 537, pp. 45–60.

Fawcett, Tom (2006). "An introduction to ROC analysis". In: *Pattern recognition letters* 27.8, pp. 861–874.

Feng, Quanlong et al. (2019). "Multisource hyperspectral and lidar data fusion for urban land-use mapping based on a modified two-branch convolutional neural network". In: *ISPRS International Journal of Geo-Information* 8.1, p. 28.

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

Froideval, Laurent et al. (2019). "A low-cost open-source workflow to generate georeferenced 3D SfM photogrammetric models of rocky outcrops". In: *The Photogrammetric Record* 34.168, pp. 365–384.

Fu, Jun et al. (2019). "Dual attention network for scene segmentation". In: *IEEE Conf. Comput. Vis. Pattern Recog.* Pp. 3146–3154.

Gao, Ang et al. (2019). "A newly developed unmanned aerial vehicle (UAV) imagery based technology for field measurement of water level". In: *Water* 11.1, p. 124.

Garg, Rajat et al. (2022). "Land cover classification of spaceborne multifrequency SAR and optical multispectral data using machine learning". In: *Advances in Space Research* 69.4, pp. 1726–1742.

Gebrehiwot, Asmamaw et al. (2019). "Deep convolutional neural network for flood extent mapping using unmanned aerial vehicles data". In: *Sensors* 19.7, p. 1486.

Geirhos, Robert et al. (2018a). "Generalisation in humans and deep neural networks". In: *Adv. Neural Inform. Process. Syst.* 31.

Geirhos, Robert et al. (2018b). "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness". In: *arXiv preprint arXiv:1811.12231*.

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

Girshick, Ross et al. (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587.

Gollob, Christoph et al. (2021). "Measurement of forest inventory parameters with Apple iPad pro and integrated LiDAR technology". In: *Remote Sensing* 13.16, p. 3129.

Gonzalez, Rafael C (2009). *Digital image processing*. Pearson education india.

Goodchild, Michael F (2007). "Citizens as sensors: the world of volunteered geography". In: *GeoJournal* 69.4, pp. 211–221.

He, Kaiming et al. (2015). "Spatial pyramid pooling in deep convolutional networks for visual recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 37.9, pp. 1904–1916.

— (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

He, Kaiming et al. (2017). "Mask r-cnn". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969.

Hirabayashi, Yukiko et al. (2013). "Global flood risk under climate change". In: *Nature Climate Change* 3.9, pp. 816–821.

Hosseini, Seiyed Mossa and Najmeh Mahjouri (2016). "Integrating support vector regression and a geomorphologic artificial neural network for daily rainfall-runoff modeling". In: *Applied Soft Computing* 38, pp. 329–345.

Hosseiny, Hossein (2021). "A deep learning model for predicting river flood depth and extent". In: *Environmental Modelling & Software* 145, p. 105186.

Howard, Andrew G et al. (2017). "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861*.

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

Hsu, Kuo-lin et al. (2002). "Self-organizing linear output map (SOLO): An artificial neural network suitable for hydrologic modeling and analysis". In: *Water Resources Research* 38.12, pp. 38–1.

Hu, Jie, Li Shen, and Gang Sun (2018). "Squeeze-and-excitation networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141.

Huang, Chengquan, LS Davis, and JRG Townshend (2002). "An assessment of support vector machines for land cover classification". In: *International Journal of remote sensing* 23.4, pp. 725–749.

Huang, Gao et al. (2017). "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708.

Huang, Gary B et al. (2008). "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments". In: *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*.

Huang, Xiao, Cuizhen Wang, and Zhenlong Li (2018). "A near real-time flood-mapping approach by integrating social media and post-event satellite imagery". In: *Annals of GIS* 24.2, pp. 113–123.

Huang, Zilong et al. (2019). "Ccnet: Criss-cross attention for semantic segmentation". In: *Int. Conf. Comput. Vis.* Pp. 603–612.

Iandola, Forrest N et al. (2016). "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size". In: *arXiv preprint arXiv:1602.07360*.

Iqbal, Umair et al. (2021). "How computer vision can facilitate flood management: A systematic review". In: *International Journal of Disaster Risk Reduction* 53, p. 102030.

Jha, Debesh et al. (2020). "Kvasir-seg: A segmented polyp dataset". In: *International Conference on Multimedia Modeling*. Springer, pp. 451–462.

Julesz, Bela (1981). "Textons, the elements of texture perception, and their interactions". In: *Nature* 290.5802, pp. 91–97.

Kim, J, Y Han, and H Hahn (2011). "Embedded implementation of image-based water-level measurement system". In: *IET computer vision* 5.2, pp. 125–133.

King, Tyler V, Bethany T Neilson, and Mitchell T Rasmussen (2018). "Estimating discharge in low-order rivers with high-resolution aerial imagery". In: *Water Resources Research* 54.2, pp. 863–878.

Kişi, Özgür (2007). "Streamflow forecasting using different artificial neural network algorithms". In: *Journal of Hydrologic Engineering* 12.5, pp. 532–539.

Knoben, Wouter JM, Jim E Freer, and Ross A Woods (2019). "Inherent benchmark or not? Comparing Nash–Sutcliffe and Kling–Gupta efficiency scores". In: *Hydrology and Earth System Sciences* 23.10, pp. 4323–4331.

Kratzert, Frederik et al. (2018). "Rainfall–runoff modelling using long short-term memory (LSTM) networks". In: *Hydrology and Earth System Sciences* 22.11, pp. 6005–6022.

Kratzert, Frederik et al. (2019). "Toward improved predictions in ungauged basins: Exploiting the power of machine learning". In: *Water Resources Research* 55.12, pp. 11344–11354.

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

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*, pp. 1097–1105.

Kussul, Nataliia et al. (2017). "Deep learning classification of land cover and crop types using remote sensing data". In: *IEEE Geoscience and Remote Sensing Letters* 14.5, pp. 778–782.

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

Lea, Colin et al. (2017). "Temporal convolutional networks for action segmentation and detection". In: *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 156–165.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning". In: *nature* 521.7553, pp. 436–444.

Lhomme, Julien et al. (2008). "Recent development and application of a rapid flood spreading method". In.

Li, Xia et al. (2019). "Expectation-maximization attention networks for semantic segmentation". In: *Int. Conf. Comput. Vis.* Pp. 9167–9176.

Li, Zhenlong et al. (2018). "A novel approach to leveraging social media for rapid flood mapping: a case study of the 2015 South Carolina floods". In: *Cartography and Geographic Information Science* 45.2, pp. 97–110.

Lima, Aranildo R, William W Hsieh, and Alex J Cannon (2017). "Variable complexity online sequential extreme learning machine, with applications to streamflow prediction". In: *Journal of Hydrology* 555, pp. 983–994.

Lin, Guosheng et al. (2017). "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation". In: *IEEE Conf. Comput. Vis. Pattern Recog.* Pp. 1925–1934.

Lin, Peirong et al. (2020). "Global estimates of reach-level bankfull river width leveraging big data geospatial analysis". In: *Geophysical Research Letters* 47.7, e2019GL086405.

Lin, Tsung-Yi et al. (2014). "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer, pp. 740–755.

Liu, Chengjun and Harry Wechsler (2002). "Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition". In: *IEEE Trans. Image Process.* 11.4, pp. 467–476.

Liu, Ping et al. (2014). "Facial expression recognition via a boosted deep belief network". In: *IEEE Conf. Comput. Vis. Pattern Recog.* Pp. 1805–1812.

Liu, Ze et al. (2021). "Swin transformer: Hierarchical vision transformer using shifted windows". In: *Int. Conf. Comput. Vis.* Pp. 10012–10022.

Lo, Shi-Wei et al. (2015). "Visual sensing for urban flood monitoring". In: *Sensors* 15.8, pp. 20006–20029.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell (2015). "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440.

Loshchilov, Ilya and Frank Hutter (2017). "Decoupled weight decay regularization". In: *arXiv preprint arXiv:1711.05101*.

Luetzenburg, Gregor, Aart Kroon, and Anders A Bjørk (2021). "Evaluation of the Apple iPhone 12 Pro LiDAR for an application in geosciences". In: *Scientific reports* 11.1, pp. 1–9.

Ma, Ningning et al. (2018). "Shufflenet v2: Practical guidelines for efficient cnn architecture design". In: *Eur. Conf. Comput. Vis.* Pp. 116–131.

Madsen, Kaj, Hans Bruun Nielsen, and Ole Tingleff (2004). "Methods for non-linear least squares problems". In.

Maier, Holger R and Graeme C Dandy (2000). "Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications". In: *Environmental modelling & software* 15.1, pp. 101–124.

Marchand, Eric, Hideaki Uchiyama, and Fabien Spindler (2015). "Pose estimation for augmented reality: a hands-on survey". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 22.12, pp. 2633–2651.

Martinis, Sandro, Jens Kersten, and André Twele (2015). "A fully automated TerraSAR-X based flood service". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 104, pp. 203–212.

Meng, Zibo et al. (2017). "Identity-aware convolutional neural network for facial expression recognition". In: *IEEE Int. Conf. Auto. Face Gest. Recog.* IEEE, pp. 558–565.

Minaee, Shervin et al. (2021). "Image segmentation using deep learning: A survey". In: *IEEE Trans. Pattern Anal. Mach. Intell.*

Mokroš, Martin et al. (2021). "Novel low-cost mobile mapping systems for forest inventories as terrestrial laser scanning alternatives". In: *International Journal of Applied Earth Observation and Geoinformation* 104, p. 102512.

Morsy, Mohamed M et al. (2016). "Distributed stormwater controls for flood mitigation within urbanized watersheds: case study of Rocky Branch watershed in Columbia, South Carolina". In: *Journal of Hydrologic Engineering* 21.11, p. 05016025.

Mosavi, Amir, Pinar Ozturk, and Kwok-wing Chau (2018). "Flood prediction using machine learning models: Literature review". In: *Water* 10.11, p. 1536.

Mottaghi, Roozbeh et al. (2014). "The role of context for object detection and semantic segmentation in the wild". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 891–898.

Moy de Vitry, Matthew et al. (2019). "Scalable Flood Level Trend Monitoring with Surveillance Cameras using a Deep Convolutional Neural Network". In: *Hydrology and Earth System Sciences Discussions*, pp. 1–21.

Munoz, David F et al. (2021). "From local to regional compound flood mapping with deep learning and data fusion techniques". In: *Science of The Total Environment* 782, p. 146927.

Muñoz, David F et al. (2021). "Fusing Multisource Data to Estimate the Effects of Urbanization, Sea Level Rise, and Hurricane Impacts on Long-Term Wetland Change Dynamics". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14, pp. 1768–1782.

Mutlu, E et al. (2008). "Comparison of artificial neural network models for hydrologic predictions at multiple gauging stations in an agricultural watershed". In: *Hydrological Processes: An International Journal* 22.26, pp. 5097–5106.

Mzurikwao, Deogratias et al. (2020). "Towards image-based cancer cell lines authentication using deep neural networks". In: *Scientific reports* 10.1, pp. 1–15.

Naseer, Muhammad Muzammal et al. (2021). "Intriguing properties of vision transformers". In: *Adv. Neural Inform. Process. Syst.* 34, pp. 23296–23308.

Neuhold, Gerhard et al. (2017). "The mapillary vistas dataset for semantic understanding of street scenes". In: *Int. Conf. Comput. Vis.* Pp. 4990–4999.

Noh, Hyeonwoo, Seunghoon Hong, and Bohyung Han (2015). "Learning deconvolution network for semantic segmentation". In: *Int. Conf. Comput. Vis.* Pp. 1520–1528.

Nourani, Vahid et al. (2014). "Applications of hybrid wavelet–artificial intelligence models in hydrology: a review". In: *Journal of Hydrology* 514, pp. 358–377.

Ojala, Timo, Matti Pietikäinen, and David Harwood (1996). "A comparative study of texture measures with classification based on featured distributions". In: *Pattern Recognition* 29.1, pp. 51–59.

Ojala, Timo, Matti Pietikainen, and Topi Maenpaa (2002). "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 24.7, pp. 971–987.

Olshausen, Bruno A and David J Field (1996). "Emergence of simple-cell receptive field properties by learning a sparse code for natural images". In: *Nature* 381.6583, pp. 607–609.

Pal, Mahesh (2005). "Random forest classifier for remote sensing classification". In: *International journal of remote sensing* 26.1, pp. 217–222.

Palen, Leysia et al. (2010). "Twitter-based information distribution during the 2009 Red River Valley flood threat". In: *Bulletin of the American Society for Information Science and Technology* 36.5, pp. 13–17.

Pally, RJ and S Samadi (2022). "Application of image processing and convolutional neural networks for flood image classification and semantic segmentation". In: *Environmental Modelling & Software* 148, p. 105285.

Paniconi, Claudio and Mario Putti (2015). "Physically based modeling in catchment hydrology at 50: Survey and outlook". In: *Water Resources Research* 51.9, pp. 7090–7129.

Panteras, George and Guido Cervone (2018). "Enhancing the temporal resolution of satellite-based flood extent generation using crowdsourced data for disaster monitoring". In: *International Journal of Remote Sensing* 39.5, pp. 1459–1474.

Park, Taesung et al. (2019). "Semantic image synthesis with spatially-adaptive normalization". In: *IEEE Conf. Comput. Vis. Pattern Recog.* Pp. 2337–2346.

Peterson, Kyle T, Vasit Sagan, and John J Sloan (2020). "Deep learning-based water quality estimation and anomaly detection using Landsat-8/Sentinel-2 virtual constellation and cloud computing". In: *GIScience & Remote Sensing* 57.4, pp. 510–525.

Piecuch, Christopher G et al. (2018). "Origin of spatial variation in US East Coast sea-level trends during 1900–2017". In: *Nature* 564.7736, pp. 400–404.

Powers, David MW (2020). "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation". In: *arXiv preprint arXiv:2010.16061*.

Qian, Yuguo et al. (2014). "Comparing machine learning classifiers for object-based land cover classification using very high resolution imagery". In: *Remote sensing* 7.1, pp. 153–168.

Qiao, Cheng et al. (2012). "An adaptive water extraction method from remote sensing image based on NDWI". In: *Journal of the Indian Society of Remote Sensing* 40.3, pp. 421–433.

Quilty, John and Jan Adamowski (2018). "Addressing the incorrect usage of wavelet-based hydrological and water resources forecasting models for real-world applications with best practices and a new forecasting framework". In: *Journal of hydrology* 563, pp. 336–353.

Rasouli, Kabir, William W Hsieh, and Alex J Cannon (2012). "Daily streamflow forecasting by machine learning methods with weather and climate inputs". In: *Journal of Hydrology* 414, pp. 284–293.

Razavi, Saman (2021). "Deep Learning, Explained: Fundamentals, Explainability, and Bridgeability to Process-based Modelling". In: *Earth and Space Science Open Archive ESSOAr*.

Ren, Shaoqing et al. (2015). "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems*, pp. 91–99.

Russakovsky, Olga et al. (2015). "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115.3, pp. 211–252.

Sandler, Mark et al. (2018). "Mobilenetv2: Inverted residuals and linear bottlenecks". In: *IEEE Conf. Comput. Vis. Pattern Recog.* Pp. 4510–4520.

Sarp, Salih et al. (2020). "Detecting Floodwater on Roadways from Image Data Using Mask-R-CNN". In: *2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*. IEEE, pp. 1–6.

Sazara, Cem, Mecit Cetin, and Khan M Iftekharuddin (2019). "Detecting floodwater on roadways from image data with handcrafted features and deep transfer learning". In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, pp. 804–809.

Schlaffer, Stefan et al. (2015). "Flood detection from multi-temporal SAR data using harmonic analysis and change detection". In: *International Journal of Applied Earth Observation and Geoinformation* 38, pp. 15–24.

Schmidhuber, Jürgen (2015). "Deep learning in neural networks: An overview". In: *Neural networks* 61, pp. 85–117.

Schnebele, E, G Cervone, and N Waters (2014). "Road assessment after flood events using non-authoritative data". In: *Natural Hazards and Earth System Sciences* 14.4, p. 1007.

Schnebele, Emily and Guido Cervone (2013). "Improving remote sensing flood assessment using volunteered geographical data". In: *Natural Hazards and Earth System Sciences* 13.3, pp. 669–677.

Schumann, Guy et al. (2009). "Progress in integration of remote sensing–derived flood extent and stage data and hydraulic models". In: *Reviews of Geophysics* 47.4.

Sekachev, Boris et al. (Aug. 2020). *opencv/cvat: v1.1.0*. Version v1.1.0. DOI: 10.5281/zenodo.4009388. URL: https://doi.org/10.5281/zenodo.4009388.

Senthil Kumar, AR et al. (2013). "Application of artificial neural network, fuzzy logic and decision tree algorithms for modelling of streamflow at Kasol in India". In: *Water science and technology* 68.12, pp. 2521–2526.

Shamsabadi, Elyas Asadi, Chang Xu, and Daniel Dias-da Costa (2022). "Robust crack detection in masonry structures with Transformers". In: *Measurement* 200, p. 111590.

Sharma, Priyanka and Deepesh Machiwal (2021). "Chapter 1 - Streamflow forecasting: overview of advances in data-driven techniques". In: *Advances in Streamflow Forecasting*. Ed. by Priyanka Sharma and Deepesh Machiwal. Elsevier, pp. 1–50.

Shen, Chaopeng (2018). "A transdisciplinary review of deep learning research and its relevance for water resources scientists". In: *Water Resources Research* 54.11, pp. 8558–8593.

Shen, Linlin and Li Bai (2006). "A review on Gabor wavelets for face recognition". In: *J. of Pattern Anal. Appl.* 9.2, pp. 273–292.

Shortridge, Julie E, Seth D Guikema, and Benjamin F Zaitchik (2016). "Machine learning methods for empirical streamflow simulation: a comparison of model accuracy, interpretability, and uncertainty in seasonal watersheds". In: *Hydrology and Earth System Sciences* 20.7, pp. 2611–2628.

Simonyan, Karen and Andrew Zisserman (2014). "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556*.

Simpson, Amber L et al. (2019). "A large annotated medical image dataset for the development and evaluation of segmentation algorithms". In: *arXiv preprint arXiv:1902.09063*.

Sit, Muhammed et al. (2020). "A comprehensive review of deep learning applications in hydrology and water resources". In: *Water Science and Technology* 82.12, pp. 2635–2670.

Smith, Corinne et al. (2022). "UAV rapidly-deployable stage sensor with electropermanent magnet docking mechanism for flood monitoring in undersampled watersheds". In: *HardwareX* 12, e00325. DOI: 10.1016/j.ohx.2022.e00325.

Smith, Laurence C and Tamlin M Pavelsky (2008). "Estimation of river discharge, propagation speed, and hydraulic geometry from space: Lena River, Siberia". In: *Water Resources Research* 44.3.

Sun, Alexander Y, Dingbao Wang, and Xianli Xu (2014). "Monthly streamflow forecasting using Gaussian process regression". In: *Journal of Hydrology* 511, pp. 72–81.

Szegedy, Christian et al. (2015). "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9.

Tan, Mingxing and Quoc Le (2019). "Efficientnet: Rethinking model scaling for convolutional neural networks". In: PMLR, pp. 6105–6114.

Tanim, Ahad Hasan et al. (2022). "Flood Detection in Urban Areas Using Satellite Imagery and Machine Learning". In: *Water* 14.7. ISSN: 2073-4441. URL: https://www.mdpi.com/2073-4441/14/7/1140.

Tavani, Stefano et al. (2022). "Smartphone assisted fieldwork: Towards the digital transition of geoscience fieldwork using LiDAR-equipped iPhones". In: *Earth-Science Reviews* 227, p. 103969.

Teng, J et al. (2015). "Rapid inundation modelling in large floodplains using LiDAR DEM". In: *Water Resources Management* 29.8, pp. 2619–2636.

Teng, J. et al. (2017). "Flood inundation modelling: A review of methods, recent advances and uncertainty analysis". In: *Environmental Modelling & Software* 90, pp. 201 –216. ISSN: 1364-8152. DOI: https://doi.org/10.1016/j.envsoft.2017.01.006. URL: http://www.sciencedirect.com/science/article/pii/S1364815216310040.

Tian, Ying-Li, Takeo Kanade, and Jeffrey F Cohn (2005). "Facial expression analysis". In: *Handbook of face recognition*. Springer, pp. 247–275.

Tikhamarine, Yazid, Doudja Souag-Gamane, and Ozgur Kisi (2019). "A new intelligent method for monthly streamflow prediction: hybrid wavelet support vector regression based on grey wolf optimizer (WSVR–GWO)". In: *Arabian Journal of Geosciences* 12.17, pp. 1–20.

Tong, Yan, Wenhui Liao, and Qiang Ji (2007). "Facial action unit recognition by exploiting their dynamic and semantic relationships". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 29.10, pp. 1683–1699.

Townsend, Philip A and Stephen J Walsh (1998). "Modeling floodplain inundation using an integrated GIS with radar and optical remote sensing". In: *Geomorphology* 21.3-4, pp. 295–312.

Tsubaki, Ryota, Ichiro Fujita, and Shiho Tsutsumi (2011). "Measurement of the flood discharge of a small-sized river using an existing digital video recording system". In: *Journal of Hydro-environment Research* 5.4, pp. 313–321.

Turnipseed, D Phil and Vernon B Sauer (2010). *Discharge measurements at gaging stations*. Tech. rep. US Geological Survey.

Vapnik, Vladimir N (1999). "An overview of statistical learning theory". In: *IEEE transactions on neural networks* 10.5, pp. 988–999.

Vaswani, Ashish et al. (2017). "Attention is all you need". In: *Adv. Neural Inform. Process. Syst.* 30.

Vinay, A et al. (2015). "Face recognition using Gabor wavelet features with PCA and KPCA-a comparative study". In: *J. of Procedia Comput. Sci.* 57, pp. 650–659.

Vitry, Matthew Moy de et al. (2019). "Scalable flood level trend monitoring with surveillance cameras using a deep convolutional neural network". In: *Hydrology and Earth System Sciences* 23.11, pp. 4621–4634.

Vogt, Maximilian, Adrian Rips, and Claus Emmelmann (2021). "Comparison of iPad Pro®'s LiDAR and TrueDepth capabilities with an industrial 3D scanning solution". In: *Technologies* 9.2, p. 25.

Wang, Xintao et al. (2018). "Recovering realistic texture in image super-resolution by deep spatial feature transform". In: *IEEE Conf. Comput. Vis. Pattern Recog.* Pp. 606–615.

Webster, Peter J (2013). "Improve weather forecasts for the developing world". In: *Nature* 493.7430, pp. 17–19.

Westoby, Matthew J et al. (2012). "'Structure-from-Motion'photogrammetry: A low-cost, effective tool for geoscience applications". In: *Geomorphology* 179, pp. 300–314.

Wu, CL, Kwok Wing Chau, and Yok Sheung Li (2009). "Predicting monthly stream-flow using data-driven models coupled with data-preprocessing techniques". In: *Water Resources Research* 45.8.

Xiang, Zhongrun, Jun Yan, and Ibrahim Demir (2020). "A rainfall-runoff model with LSTM-based sequence-to-sequence learning". In: *Water resources research* 56.1, e2019WR025326.

Xie, Enze et al. (2021). "SegFormer: Simple and efficient design for semantic segmentation with transformers". In: *Adv. Neural Inform. Process. Syst.* 34, pp. 12077–12090.

Xie, Saining et al. (2017). "Aggregated residual transformations for deep neural networks". In: *IEEE Conf. Comput. Vis. Pattern Recog.* Pp. 1492–1500.

Xu, Yonghao et al. (2019). "Advanced multi-sensor optical remote sensing for urban land use and land cover classification: Outcome of the 2018 IEEE GRSS data fusion contest". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 12.6, pp. 1709–1724.

Yan, Jining et al. (2020). "Temporal convolutional networks for the advance prediction of ENSO". In: *Scientific reports* 10.1, pp. 1–15.

Yin, Jie et al. (2015). "Using social media to enhance emergency situation awareness". In: *Twenty-fourth international joint conference on artificial intelligence*.

Yin, Minghao et al. (2020). "Disentangled non-local neural networks". In: *Eur. Conf. Comput. Vis.* Springer, pp. 191–207.

Yu, Fisher et al. (2020). "Bdd100k: A diverse driving dataset for heterogeneous multitask learning". In: *IEEE Conf. Comput. Vis. Pattern Recog.* Pp. 2636–2645.

Yuan, Yuhui, Xilin Chen, and Jingdong Wang (2019). "Object-contextual representations for semantic segmentation". In: *arXiv preprint arXiv:1909.11065*.

— (2020). "Object-contextual representations for semantic segmentation". In: *Eur. Conf. Comput. Vis.* Springer, pp. 173–190.

Yuan, Yuhui and Jingdong Wang (2018). "Ocnet: Object context network for scene parsing". In: *arXiv preprint arXiv:1809.00916*.

Zagoruyko, Sergey and Nikos Komodakis (2016). "Wide residual networks". In: *arXiv preprint arXiv:1605.07146*.

Zhang, Xiangyu et al. (2018). "Shufflenet: An extremely efficient convolutional neural network for mobile devices". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6848–6856.

Zhang, Zhen et al. (2019). "In-situ water level measurement using NIR-imaging video camera". In: *Flow Measurement and Instrumentation* 67, pp. 95–106.

Zhang, Zhengyou (2000). "A flexible new technique for camera calibration". In: *IEEE Transactions on pattern analysis and machine intelligence* 22.11, pp. 1330–1334.

Zhang, Zhengyou et al. (1998). "Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron". In: *IEEE Int. Conf. Auto. Face Gest. Recog.* IEEE, pp. 454–459.

Zhao, Guoying and Matti Pietikainen (2007). "Dynamic texture recognition using local binary patterns with an application to facial expressions". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 29.6, pp. 915–928.

Zhao, Hengshuang et al. (2017). "Pyramid scene parsing network". In: *IEEE Conf. Comput. Vis. Pattern Recog.* Pp. 2881–2890.

Zheng, Yufeng et al. (2018). "Processing global and local features in convolutional neural network (cnn) and primate visual systems". In: *Mobile Multimedia/Image Processing, Security, and Applications 2018*. Vol. 10668. SPIE, pp. 44–51.

Zhou, Bolei et al. (2019). "Semantic understanding of scenes through the ade20k dataset". In: *Int. J. Comput. Vis.* 127.3, pp. 302–321.

Zhu, Zhen et al. (2019). "Asymmetric non-local neural networks for semantic segmentation". In: *Int. Conf. Comput. Vis.* Pp. 593–602.

Zoph, Barret et al. (2018). "Learning transferable architectures for scalable image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8697–8710.

Zscheischler, Jakob et al. (2020). "A typology of compound weather and climate events". In: *Nature Reviews Earth & Environment*, pp. 1–15.