May 2021

Application of Deep Learning for Imaging-Based Stream Gaging

Ryan Lee Vanden Boomen
University of Wisconsin-Milwaukee

APPLICATION OF DEEP LEARNING FOR IMAGING-BASED STREAM GAGING

by

Ryan Vanden Boomen

A Thesis Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in Computer Science

at

The University of Wisconsin-Milwaukee

May 2021

ABSTRACT

APPLICATION OF DEEP LEARNING FOR IMAGING-BASED STREAM GAGING

by

Ryan Vanden Boomen

The University of Wisconsin-Milwaukee, 2021
Under the Supervision of Professor Zeyun Yu and Professor Qian Liao

In the field of water resources management, one vital instrument utilized is the stream gage. Stream gages monitor and record flow and water height within some water body. The United States Geological Survey maintains a network of stream gages at many locations across the country. Many of these sites are also equipped with webcams monitoring the state of the water body at the moment of measurement. Previous studies have outlined methods to approximate stream gage data remotely with limitations such as the requirement of detailed depth information for each site. This study seeks to create a process for training a deep neural network that will utilize the webcam and stream gage data to generate a water height prediction based on the visual state of the water body. The goal of this study is to outline a training process and model that can be utilized on a variety of different sites while only requiring that location's existing webcam and stream gage data. This paper outlines the experiments on stream gages located at the Clear Creek in Iowa, Auglaize River in Ohio, and Milwaukee River in Wisconsin. The process outlined utilizes transfer learning and well-known image classification models as a basis for a generalized river height regression model. Across the training, validation, and deployment experiments, the developed process shows great success in creating an accurate model for various sites of different conditions and river clarity. The results of this study show confidence in future studies utilizing remote stream gaging with image regression.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

USGS stream gage stations selected for this study. Discharge and stage of the 10-year flow are estimated with annual peak flow data following the Log-Pearson type III distribution model. Data sourced from United States Geological Survey in USGS Water Data for the Nation, 2020, Retrieved from https://waterdata.usgs.gov/nwis/rt/

I. Introduction

Stream Gaging

Water Resource Management is the field of analyzing, tracking, and planning water resources for the best usage. As we are facing an ever-worsening climate crisis, the importance of water as a vital resource will increase exponentially. The goal of water resource management is to develop methods to monitor and optimally allocate water resources across the various entities that require the resource. Therefore, it is important to accurately measure and determine the status of natural water bodies in a geographic region as it will provide analysts the ability to understand how water availability and excess will affect the region.

Stream gaging is an instrument that is vital in tracking the flow of a water body. Stream gages are typically capable of tracking the flow discharge and water height in a river or stream. The data that is collected from stream gages are important for analyzing water quality and assessing flood conditions. Stream gages can measure the height of a water body. The water height is also typically known as a stage measurement. The instrument will also measure the water flow discharge using a “rating-curve” that is calibrated individually for each site. Both measurements provide insight into possible flooding or drought conditions of the stream.

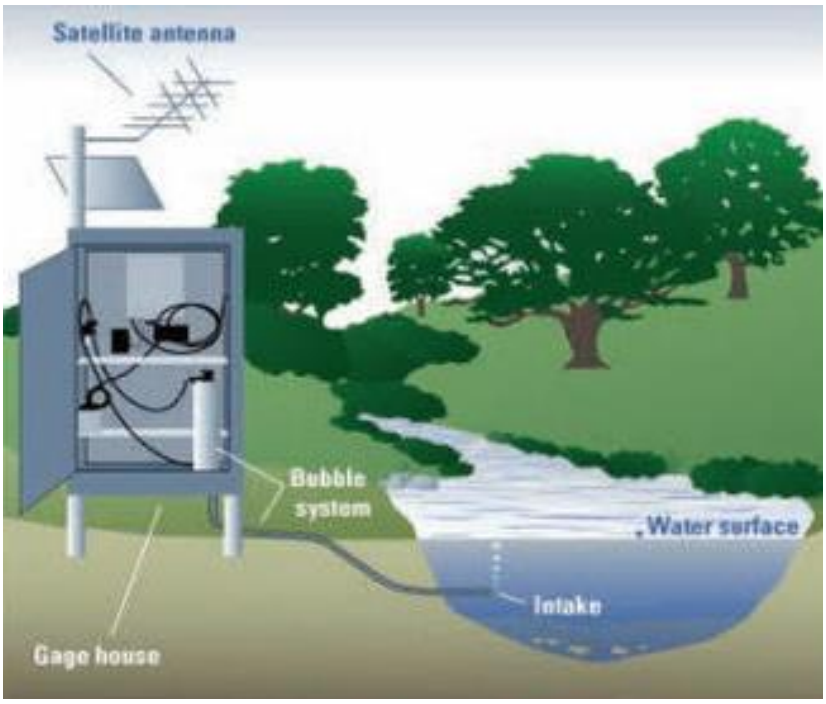

Figure 1: Stream Gage System Design

Illustrated design of a stream gage system. Reprinted from United States Geological Survey in *What is a streamgage?*, 2021, Retrieved from https://www.usgs.gov/centers/ut-water/science/what-a-usgs-streamgage

In the United States, most stream gages are managed by the United States Geological Survey (USGS). Over 10,000 stream gages are active in the USGS network. However, the number of stream gages is considered too sparse in many regions to get a full picture of water conditions. According to USGS, average operation cost was about $14,000 annually per typical continuous stream gage (Norris et al., 2007) with most cost associated with site inspections and field experiments. Therefore, a need arises in minimizing the cost for current stream gaging setups. In addition, physical management of stream gaging sites can result in hazardous conditions especially during extreme weather events such as flooding. Minimizing cost and human risk to stream gaging methods while maintaining a standard of reliability is a task that will increase the quality of stream gaging networks over time.

Remote Sensing Approaches

One possible solution for minimizing cost in stream gaging networks is by implementing remote or non-contact sensing methods in water sites. As mentioned earlier, most of the cost comes from the need for site inspections and field experiments to ensure accuracy for the physical instruments. Remote sensing solutions would ideally incur less wear and therefore maintenance over time. While a remote sensing method would also require a calibration process, the remote method would require calibration less often due to the decrease in maintenance. Since a remote method does not have contact with the water body, the risks associated with working in hazardous conditions are also lessened. Therefore, there is potential in developing remote sensing methods to solve the difficulties in stream gaging methods.

It has been demonstrated that radars can be applied to detect both water depth and surface velocity, thus the discharge (Costa et al., 2000; Melcher et al., 2002). River stage and slope can be measured by through satellite-based remote sensing, and the discharge can be subsequently estimated with raring relations or hydraulic models (Durand et al., 2016; Kouraev et al.; 2004, Bjerklie et al., 2003). Large-Scale Particle Image Velocimetry (LSPIV) is a computer vision-based method which can resolve 2D flow velocity distribution over a river surface. Many studies have shown that LSPIV can provide a low-cost nearfield remote sensing solution to stream gaging (Muste, 2008; Tauro et al., 2018; Ran et al., 2016). Despite great potential, limitations exist for LSPIV to be an automated and self-contained approach for continuous gaging solution. Additional depth measurements are still necessary for a proper image orthorectification and for discharge calculation (Li et al., 2019).

Images of river surface present rich features which may correlate with important hydraulic parameters. Studies demonstrated that surface velocity or temperature structures originate from turbulence production due to bottom friction can be used to estimate water depth distribution, thus discharge can be calculated without addition bathymetry measurements (Johnson and Cowen, 2016; Puleo et al., 2012; Legleiter et al., 2017; Jin and Liao, 2019). While these studies attempted to correlate surface images or extracted features with hydraulic processes through the mechanics of turbulence, image features may also contain information of the flow via some unknown mechanisms or in ways that are not able to be quantified easily through traditional statistical analysis. A person may visually sense the change of surface features, such as the water level, surface area, turbidity, ripples, and waves, as the river flow rate varies. It is therefore reasonable to hypothesize that a correlation between these features with hydraulic parameters can be directly established through Machine Learning.

Utilizing ground-level imaging data shows potential as a catalyst for the development of a remote stream gaging method. In addition to their vast stream gaging network, the USGS also maintains a network of webcams located at many stream gage sites. Each webcam provides a high-resolution view of the current state of that site's water body. The webcams are designed to capture an image within a given interval (typically 15 minutes to an hour) and send the image to the USGS servers. The captured images will correlate with a recorded measurement on that site's stream gage. Therefore, with the stream gage data and a corresponding image, an analyst can get a full understanding of the conditions of a certain site.

The webcam images and site data can also be utilized to create a new remote sensing method. As I have discussed, there is the possibility for a correlation between the visual features

of a water body and the water stage height. Since both the stage height and image data for a site exist, the stage height data could be used to train the interpretation of a given image. One could utilize Machine Learning on the USGS stream gage data to create a Machine Learning remote stream gage.

Study Proposal

Utilizing the USGS stream gage data, this study will outline a method for developing an Image Regression Stream Gaging model that predicts river height from webcam data. This way, river height readings can be received remotely without dependence on a physical measurement system. Therefore, this system will prove to be an effective remote tracking method. A DNN Stream Gaging model will require previous image data and stream gage readings to train the model. Once that data is gathered, the system will follow a generalized process to create a trained model for that webcam site. The necessary data is made available publicly from the USGS. Therefore, since training is a general process, and the data is available from a centralized source, the Image Regression Stream Gaging model would require little effort for implementing on different locations. This study proposes a method for stream gage remote sensing that is location-agnostic. This method will allow for the trivial creation of many similar Image Regression Stream Gaging models located at different sites. Image Regression Stream Gaging therefore acts as a method for automated, self-contained, and accurate ground-level Remote Sensing Stream Gaging solution.

An Image Regression model will output a real number value. This method will generate approximations of the actual river height value for some site. These approximations should lie within relative accuracy of the true height value. While error is an inescapable occurrence in an

Image Regression model, approximating the general trend in water stage height over time proves efficacy in this method as a remote stream gaging method.

This study acts as a proof of concept for the idea of an Image Regression Stream Gaging system. Utilizing a select subset of locations available from the USGS, a method is proposed for training an Image Regression Neural Network for a specific site. These Neural Network models will be based on popular Transfer Learning methods. Therefore, this study hopes to create relatively accurate results that indicate initial success and convergence in this general training method. I wish to show that this process will create accurate stream gage results for a sample of USGS sites given sufficient data.

II. Related Work

Machine Learning in Water Resources Management

Machine Learning as a concept involves the creation of methods that generate mathematical approximations utilizing previous occurrences. The idea of having a learning model means that the insights generated come from trends found in previous data. A Machine Learning model needs to be trained on previous data. This previous data is commonly known as training data. It is necessary for training data to represent a diverse array of conditions within the dataset that the model will predict. Machine Learning models typically lack the ability to effectively extrapolate and make interpretations on data with foreign conditions. Ideally, to maximize the amount of conditions a model is trained on, the number of samples that is added to the training data should also be maximized. However, since evaluation of a Machine Learning model is dependent on how the model will interpret future data, training data must be separated from the dataset and disqualified from any experimental evaluation. The trade-off between training and evaluation data is a struggle that can be best alleviated by working with datasets with a substantial number of samples. Thus, Machine Learning is most effective in fields and domains where data is abundant.

Machine Learning is not an unfamiliar concept to the field of Water Resource Management. Many studies have been effective in using Machine Learning to form insights using water data from various instruments (Adamala, 2017; Pan et al., 2019; Pan et al., 2020; Kreyenberg et al., 2019; Hu et al. 2019). Machine Learning has been utilized in Remote Water Level Sensing studies previously as well. Some studies have utilized DNNs to create accurate readings on satellite images of rivers and other water bodies (Miao et al., 2018; Chen et al.,

2018). A few studies have utilized images from social media to find water level from objects present in the image (Chaudhary et al., 2019; Lopez-Fuentes et al., 2017). One study proposes a method for segmenting a ground-level, real-time video containing a river and calculating the height from this segmentation map (Jafari et al., 2020). However, there has yet to be a study investigating the application of an Image Regression model on ground-level river image data. Therefore, this study's investigation into an Image Regression Stream Gaging method is novel.

Since Water Resource Management is a data-oriented field, Machine Learning has been a popular tool for studies in the field. Research in this field have been able to gather large volumes of data from various instruments. Utilizing this data, new Machine Learning methods have created new insights and better methods for analyzing and distributing water resources. Water Resource Management is an emerging domain for applied Machine Learning research.

Deep Neural Networks

As an effective Machine Learning technique, Deep Learning Neural Networks (DNNs) are a popular method for generating predictions on non-linear data. A Neural Network is a data structure consisting of layers of weighted nodes. Each node is weighted based on training and given an activation function. At each node, a linear equation is computed based on the input and the weights. The result of this equation is then augmented using the activation function. After leaving the node, the result continues through the network. A Neural Network can already become very complicated with a few nodes and layers. However, a simple Neural Network can make respectable approximations of non-linear data.

A Deep Neural Network is an expansion of the Neural Network structure. A Deep Neural Network will utilize many layers and nodes to make a prediction. This deepening of a Neural

Network structure can lead to models with millions of trainable weights. Due to the scale of Deep Neural Network models, the computational requirements of training and utilizing DNNs is intense. In addition, parsing and understanding the resulting linear equation is impossible for humans. Therefore, DNNs are difficult to work with due to their computational requirements and difficulty for understanding. However, DNNs have been found to be effective in finding correlations and trends across various forms of data. DNNs are a popular Machine Learning method due to its effectiveness

Many researchers have found it effective to construct and train DNNs using image data (Wang et al., 2019). Due to the non-linear state of most image data, DNNs have been one of the most popular Machine Learning techniques for creating accurate models from visual data. A few studies have already utilized DNNs to make predictions on satellite images of water bodies (Ling et al., 2019; Cho et al., 2019). As we know from various LSPIV studies, water metrics such as river height and discharge hold a strong correlation with features found in visual data. Intuitively, given enough data, humans would also be able to estimate the water height of a body only based on the visual data. Therefore, applying DNNs to create an Imaging-based Stream Gaging system shows high potential in finding accurate predictions for water body metrics.

Transfer Learning

One of the most popular problems for Imaging-based DNNs is the task of Image Classification. Given a dataset of images and predefined labels, Image Classification tasks one with creating a model that would predict or "classify" an input image with the correct label. The most prominent application of Image Classification is ImageNet (Deng et al., 2009). ImageNet is a data set containing over 14 million images categorized into 1,000 labels. Due to its size and

diversity in labels, ImageNet is a common benchmark for Image Classification DNNs. One common trait of successful ImageNet models is their flexibility in training and predicting specialized data.

Therefore, a common Image Classification method is to utilize a popular ImageNet model with pre-trained weights, modify the DNN to fit a new set of image data and labels, and continue training the model with specialized data. The idea behind this method is to leverage previous knowledge and accelerate the training process. This method is known as Transfer Learning. Transfer Learning from ImageNet allows a model to quickly find important features during the training process (Ribani et al., 2019). While inventing custom DNNs to solve the formulated regression problem may be more accurate, the creation of custom methods can be costly and ineffective, particularly because the number of images is limited. Therefore, a Transfer Learning method using effective Image Classification Deep Neural Networks was implemented.

A Convolutional Neural Network (CNN) is a category of Neural Network models that are commonly utilized with spatially dependent two-dimensional data. CNNs are the ideal choice for creating Image Prediction models (Wang et al., 2019). CNNs rely on the usage of two types of layers: Convolutional and Pooling. A Convolution node utilizes a two-dimensional filter to transform the input data. Typically, the filter is a square mask that is defined by the model. The weights on this filter are found during the training process. The goal of the Convolution layer is to reveal important features, such as “edges” in the data. Therefore, the output of a Convolution layer is typically described as a feature map. The Pooling layers exist to reduce size of the outputted feature maps. Pooling compresses the values of one group of pixels (a kernel) into one

pixel. Common pooling methods include taking the maximum or mean values of all pixels in the kernel. Outside of improving performance down the line, a main advantage of pooling is the act of reducing noise in a feature map. Pooling acts as form of "focusing" the model on what has been found to be important. Typically, a Convolutional Neural Network will utilize an interleaved combination of Convolution and Pooling layers to output a final set of feature maps. Then, the final feature maps are typically flattened into a one-dimensional vector. Fully Connected layers will then operate and create a prediction on this vector based on previously trained data. A Fully Connected node will perform a linear equation based on the input data and trained weights for that node. This linear equation is then inputted into an activation function that acts as the output of the node. By filtering and examining feature maps, a CNN is especially effective at finding deep correlations between discrete features in images and the training values for that image.

Typically, Imaging CNNs will contain millions of trainable weights and require significant computation time to train. Training a CNN from scratch would require substantial time to interpret the input images correctly and approach anywhere near the global minimum error. Utilizing Transfer Learning sets a starting point for the model before training. ImageNet models have already been trained on recognizing general shapes. This general knowledge can easily transfer to any imaging problem. Transfer Learning can be a vital task in creating an imaging model.

Image Regression

While a Classification method is effective for assigning a discrete label to an element, limitations can appear when utilizing continuous data. With continuous data, such as stream gage

stage or discharge, one would typically "bin" the data within specific thresholds. Each bin is assigned a discrete label. Binning is a common way of discretizing continuous data. Regardless of the number of bins utilized, this method results in a model that can only estimate within a threshold. Error can arbitrarily occur at the border between two thresholds. Therefore, binning continuous data is not ideal for problems that require accuracy. It is preferable to utilize a regression model on a continuous dataset.

In the previous section, I discussed applying Transfer Learning with popular Image Classification models. Image Classification models output what are known as “one-hot” vectors. The output is an array of real numbers between 0 and 1. The length of the array is equal to the number of classes. Each value in the array corresponds with how confident the model believes this image belongs to this class. Therefore, the largest value is the assigned class for that image.

While this method will create a discrete label for some data, it is creating that label by discretizing an array of scalar values from the one-hot array. Therefore, the mathematical basis exists to remove the discretizing process and thus transform the Classification model into a Regression model. I simply must modify the output and fully connected layers to convert to a Regression model. Where Classification is the act of predicting a discrete label, Regression will predict a scalar value. For Image Classification models, studies have shown effective results performing this task with accurate results (Fischer et al., 2015). Therefore, the Transfer Learning method can also be effective at predicting continuous data. It is thus possible for an Image Regression model to predict metrics given water image data.

III. Method

Data

As a data-oriented method, a major limitation to a Deep Neural Network is the quality and variance in the data offered. Since DNNs are not suited to extrapolation, it is ideal to train a neural network on a diverse dataset that will best represent all possible features of the entire dataset. For example, the color of a water body can vary based on the angle of the sun reflecting on the body. In addition, the amount of foliage and snow can vary depending on the season. Therefore, for this problem, it is ideal to find data that would provide high quality representation to all seasons and times of day. The best way to ensure that the dataset has a diverse representation is to gather images from an automated system that will run for a long period of time. This system will then be able to gather images during regular intervals. Thankfully, the USGS already provides such a system.

All USGS stream gage stations have instruments that measures the stage of the stream and other additional parameters such as turbidity, temperature, and dissolved oxygen, etc. Many USGS stations include live webcam images of the nearby river. Each webcam captures an image of the water body at a specific time interval. Each image is then uploaded and made publicly available on the USGS website. The frequency of the image captures and period that the webcam is active is dependent on the site. In addition, the camera angle, the position of the camera along the river, and the resolution of the images captured vary across the sites.

Since the USGS provides webcam data at the same intervals as their stream gage data, they were chosen as the data source for this study. Synchronizing the stream gage data and the

image would be ideal to train a DNN to fit a prediction on an image to the actual stage height value. As a constantly maintained system, this system was able to capture data daily from March 2020 to January 2021. While a site may be unavailable for some days, the USGS stream gage dataset represents all times of day throughout every season during this time. Thus, the USGS stream gage dataset was the clear choice for this study to test a remote stream gage.

For this study, the variation between images is important. One could intuit that two images with different water height will have different shoreline features. The goal of the DNN training is to train a model to find those features. In addition, the vegetation and lighting in each image will vary based on the season, time of day, and weather. These variations are important as they will appear in any hypothetical stream gaging system. When looking at images between two different sites, the camera angle, vegetation, and body shape can vary wildly. By human intuition, it is difficult to find similar features between two sites as surface images cannot easily indicate the height under the water. Therefore, difficulties can appear when training with data from two different sites. To study the application of image regression DNNs on river image data of varying stages, it was ideal to create separate models for each site. This method allows us to have a best-case scenario for training a model to predict the stage or discharge of a river at a given time. Since the camera angle and general environmental features would be almost identical across all images in the dataset, the model would focus on features that have a stronger correlation to the flow condition. By creating separate models on different sites, the possibility for confusion is decreased while proposing a plausible remote stream gaging system.

The USGS Upper Midwest Water Science Center hosts a website titled “Storm Summary Timelapse Dashboard”, which displaces live webcam images from 285 USGS stream gage

stations (https://apps.usgs.gov/sstl/). After investigating various sites from the dashboard, three webcam sites were selected for this study according to data availability and variability of image features and hydrologic conditions: (1) Clear Creek near Coralville, IA; (2) Milwaukee River near Cedarburg, WI; and (3) Auglaize River near Kossuth, OH. Summary of the three stations including their locations, drainage areas, maximum river stage of available data and the estimated 10-year flows are given in Table 1.

Table 1: USGS Selected Location Demographics

Site Name	USGS station ID	Location (Latitude, Longitude)	Drainage Area (mi^2)	Maximum stage (ft)	10-year flow discharge (ft^3/s)	10-year flow stage (ft)
Clear Creek near Coralville, IA	05454300	(41°40′36″, 91°35′55″)	98.1	16.4	4,710	13.4
Milwaukee River near Cedarburg, WI	04086600	(43°16′49″, 87°56′33″)	607	14.0	5,064	12.2
Auglaize River near Kossuth, OH	04185935	(40°41′09″, 84°15′56″)	201	16.7	NA	NA

USGS stream gage stations selected for this study. Discharge and stage of the 10-year flow are estimated with annual peak flow data following the Log-Pearson type III distribution model. Data sourced from United States Geological Survey in *USGS Water Data for the Nation*, 2020, Retrieved from https://waterdata.usgs.gov/nwis/rt/

Since only small subset of webcam data from the previous couple days are available through USGS's website, a Python script was developed to gather the newly available images. This script was deployed daily between March 2020 and January of 2020. This process was interrupted for about two weeks in October as the website underwent an update. In addition, images with ice covering the river surface were excluded from the analysis. Some images captured at the Clear Creek site in September and November 2020 were also excluded as images appeared blurry, possibly due to the stained camera lens. At the end of this process, 2,861 images from the Milwaukee River site, 6,777 images from the Clear Creek site, and 2,348 images from the Auglaize River site were acquired for this study. Each image was captured in 1920×1080 resolution and contained full RGB information. Three sample images from each site are shown in Figure 1, taken during the spring, summer and fall of 2020, respectively, are shown in Figure

1 to demonstrate the change of surrounding environment and the riparian vegetation, and the potential impacts on image analysis.

Figure 2: USGS Selected Location Image Samples

Sample webcam images from the three selected USGS stream gage sites (left pane: early spring; middle pane: summer; right pane: fall). Reprinted from United States Geological Survey in *Storm Summary Timelapse Dashboard*, 2020, Retrieved from https://apps.usgs.gov/sstl/

When searching for testing sites, two additional factors were considered as important for properly testing the prediction model: the existence of relative markers and the obfuscation of the water body. Relative markers exist as static elements in the environment. A relative marker should exist in the same location in every image. Relative markers may include rocks, roads, bridge piers, trees, and signs. When analyzing water image data, humans commonly utilize a marker to visually judge relative water height and are known to be effective with Imaging DNNs. Therefore, it is important to test on sites where markers are abundant and where markers are sparse. When a portion of the water body is obfuscated, a prediction model will lose the ability to

analyze a water body using the markers that are currently not visible. Therefore, when the water body is covered by a tree, car, or another obstruction, important data is lost. Considering sites where features are occasionally obstructed is important to show the flexibility in a prediction model and that the model will not fixate on unreliable features. As shown in Figure 1, the three selected sites have different characteristics of relative markers and visual obfuscation.

Clear Creek images show a sizeable body of water lying along a natural park. The webcam on this site provides a centered view of a large amount of the water body. The overpass in the image provides a strong relative marker for comparison to the height of the stream. Due to the clear and unobstructed view to a large amount of the stream and the clear relative markers surrounding it, this location was assumed to be the best-case dataset to river stage prediction.

The webcam along the Milwaukee River site provides a perpendicular angle to the river. One interesting note for this dataset is the shoreline that will flood and be lost in higher water levels. One may consider the shoreline as a primary marker in judging water height. In this case, the model may seek to find different markers in higher stages. One other interesting feature is the tree that is obfuscating the overpass during the summertime. The overpass should be a solid marker for the model. However, in different seasons, the amount of obfuscation may increase. Therefore, this dataset will pose a common practical challenge for this method.

The webcam view at the Auglaize River site is like that of the Milwaukee River site with a perpendicular camera angle and obfuscating trees. However, this camera is at a higher angle and covers a nearby road. This method poses different challenges in having multiple river-covering elements. Therefore, this dataset will test whether the model will be able to make predictions with minimal consistent markers.

Pre-processing was performed on the images before use in the study. Each image was rescaled to 420x420 pixels to be compatible with the selected Transfer Learning model. Due to changes in weather and prolonged outdoor exposure, the camera may shift in position and rotation over time. To account for this possibility in the model, a random rotation, shift, and shear transformation were applied to each image. Each image was randomly rotated between -5 and +5 degrees. Shifting occurs on each image randomly between -4% and 4% in the x and y dimensions. Shearing occurs within the range of -0.5 and 0.5 degrees on each image. These pre-processing steps help create a more robust and accurate model when faced with new data.

To properly train the model based on the images gathered, river stage and discharge at the time each image is captured is required. This information is publicly available through the USGS water data sever (https://waterdata.usgs.gov/nwis). A Python script was developed to extract time series of stage and discharge at each test site from the "Current/Historical Observations" data source, which reports measurements at a 15-minute interval. Stage values at the time of images acquisition were obtained through interpolation of the time series and were subsequently "labeled" to each image. Conceptually, the Machine Learning model can be trained to report either the stage or discharge. However, examination of data at all three sites found that discharge values were not uniquely related to stage values, suggesting that additional corrections were applied by USGS. Moreover, since the stage is directly measured and discharge is an estimation based on an empirical rating relation, only stage measurements were used for training and validation in this study.

Given this dataset, I can define the parameters for this experiment. First, I want to know if it is possible for an Imaging Regression model to train an image to predict water stage height.

Therefore, a training process is initiated and monitored. The goal for this step is to determine if a popular training algorithm can work to minimize error in predicted height over time. In addition, it is not only important to see if the model will see more accurate results on the data being trained over time, but it is also vital to see how the model reacts to new data. Therefore, after each training iteration, a validation dataset should be utilized to see how the error on untrained data changes over time. With a theoretical remote stream gaging method, there becomes a point where, once the method is deployed, the model would exclusively be seeing new data. I am curious to see how the model would stand up to new data over time. Therefore, a deployment experiment is also defined. This experiment will generate a time series for water height prediction over a period. A strong correlation between the predicted and actual height will show potential for this study as a remote stream gaging method.

With the data retrieved, the prediction of river stage based on image information can be formulated as a regression problem. In this study, the primary objective is to study whether a Deep Neural Network can find any strong correlation between the image data and the river stage. Figure 2 shows the distribution of the "labeled" river stages of all image data at the three test sites. The dataset presented a "lopsided" distribution skewed right, with most data representing baseflow conditions and fewer were high flow conditions after precipitation events. To augment the dataset, stage values were shifted by subtracting the mean value at each site. Each dataset was split to three sets for training, validation, and deployment, respectively. First, the dataset was divided in a "before" set and an "after" set based divided on a specific date in the Summer of 2020. This dividing date differs from each site due to dataset composition and the number of images gathered. From the "before" set, 90% of the images were reserved for the training set, while the remaining images were assigned to validation during the training process. In the "after"

set, 90% of the images were reserved for the deployment experiment and isolated from the training process. To account for potential extremes in the “after” set, the remaining 10% were randomly assigned to the training set. The idea is that this will help in situations where global minimum and maximums exist in the deployment set while not being a comprehensive peek at the deployment set. As a result, numbers of images available for DNN training were 3772, 1905 and 1239 for the Clear Creek, Milwaukee River and Auglaize River, respectively. For validation, 370 Clear Creek images, 185 Milwaukee River images, and 144 Auglaize River images were selected. The deployment set consists of 3005 images for Clear Creek, 955 for Milwaukee River, and 1109 for Auglaize River.

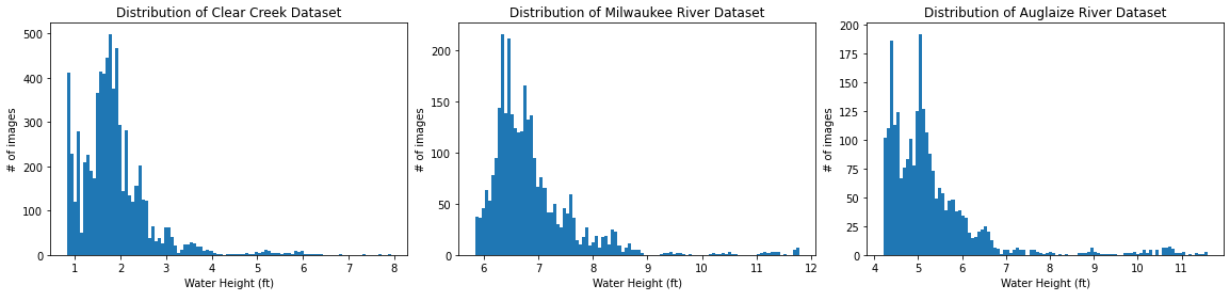

Figure 3: Stream Gage Water Height Histogram

Histograms outlining the number of images for the actual water height value. Height values were discretized within 100 total bins uniform across data range. (left pane: Clear Creek; middle pane: Milwaukee River; right pane: Auglaize River). Data sourced from United States Geological Survey in *Storm Summary Timelapse Dashboard*, 2020, Retrieved from https://apps.usgs.gov/sstl/

Neural Network Model

Given the data, an Imaging Deep Neural Network model is proposed. As stated earlier, utilizing Transfer Learning and Image Regression, I can modify a popular Image Classification model to fit this problem. While inventing custom DNNs to solve the formulated regression problem may be more accurate and necessary in future study, the creation of custom methods can be costly and ineffective, particularly because the number of images was limited in this

preliminary study. Therefore, a Transfer Learning method using effective Image Classification Deep Neural Networks was implemented. The two base models initially chosen were VGG-16 (Simonyan et al., 2015) and InceptionV3 (Szegedy et al., 2015). Both models are popular Convolutional Neural Networks that were designed for the ImageNet classification benchmark. Both models have previously shown strong results in being modified for regression tasks. Training and validation experiments with river image data showed that VGG-16 model performed slightly better than InceptionV3 in terms of stability and accuracy. Therefore, only VGG-16 model results are presented and discussed in this study.

VGG-16 is a convolutional neural network structure invented by Karen Simonyan and Andrew Zisserman in 2014. This model gained reputation for its success in the 2014 ImageNet Large Scale Visual Recognition Challenge (ISLVRC) and has become a very popular model for Image Classification tasks. As illustrated in Figure 3, VGG-16 has a relatively simple structure with 16 layers. The first 13 layers consist of clusters of sequentially smaller convolutional layers connected by a “max” pooling operation. The last 3 layers are Fully-Connected and lead to a “softmax” classification layer. One of the main advantages of VGG-16 is the relative simplicity of the network. The model can be easily considered as a linear progression of blocks of decreasing size. Each block takes in an image, creates feature maps, and then uses a pooling layer to shrink those maps. One setback to this model however is the large amount of weights for each node and the overall size of the model. VGG-16 contains over 138,423,208 weights. Therefore, while the model typically performs well in classification tasks, the model requires one to consider the trade-offs between performance and ease of understanding.

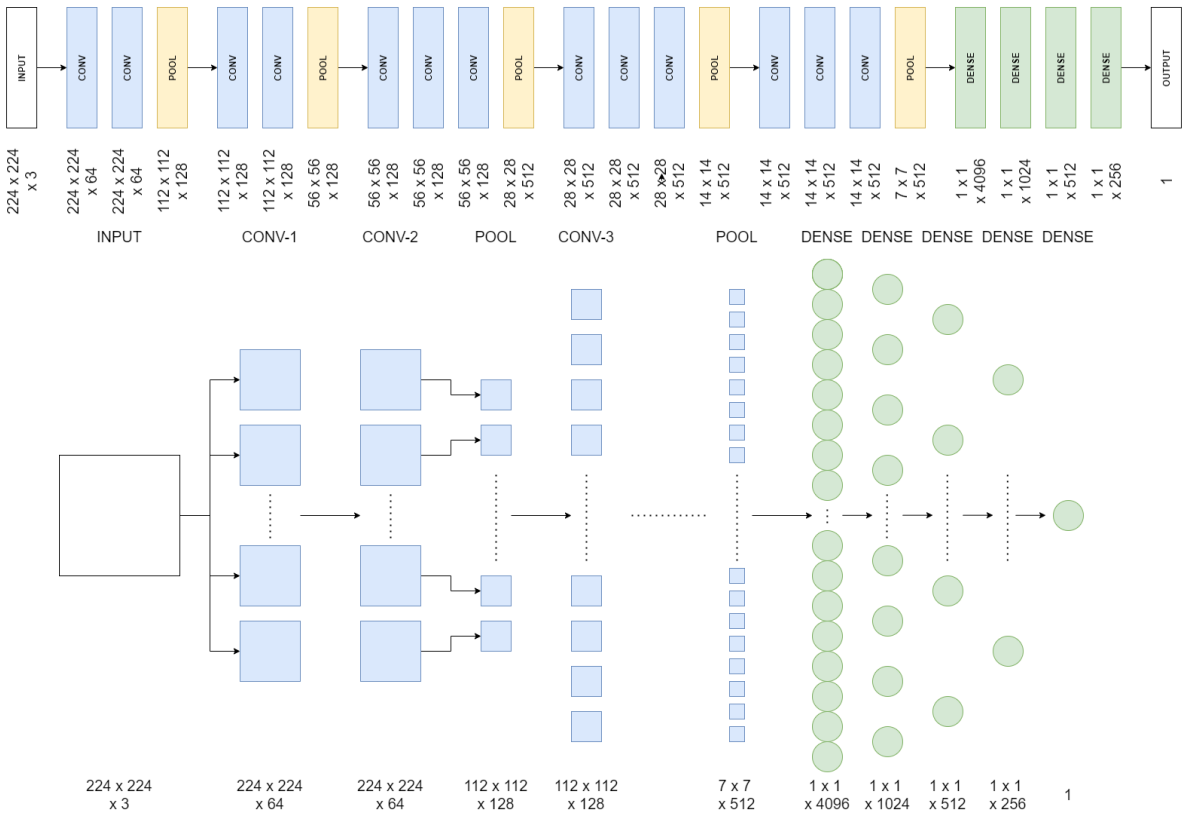

Figure 4: VGG-16 Neural Network Structure

Top: sketch of neural network layers. Bottom: zoomed-in network structure (blue squares: feature maps, circles: fully-connected nodes)

Since these models exist as Image Classification methods, they are originally designed to output an array of class label probabilities. To modify this model to output a continuous scalar value, the output layers of each network were removed and replaced with new output layers designed for regression models. These output layers typically exist after the convolutional and pooling layers in a network and help define the output of the network. At the end of each model, four fully-connected layers were added. The first three fully-connected layers contain a decreasing number of nodes and an activation function, which acts as the output for each node given input from the previous layer. The first two fully-connected layers have a Sigmoid activation function:

$$\frac{1}{1 + e^{-x}} \tag{1}$$

where x is the weighted summation of the nodes from the previous layer. The weights are determined by the training process. The Sigmoid activation function is a popular activation for regression tasks as it is a monotonic function that returns a value between 0 and 1. The third activation function contains a ReLU activation function:

$$\max(0, x) \tag{2}$$

ReLU is a popular activation function due to its ability to predict non-linear data with a constant derivative. The final fully-connected layer contains 1 node with no activation function. The output of this node is the scalar value representing the river stage of an image. This node acts as the weighted sum of the previous nodes. With these modifications, the Transfer Learning models are now suited to be trained to calculate the real number river stage value of a given site's image data.

Training Process

To train a Deep Neural Network, I wish to find the weights for the neural network that minimize the error. Due to the complexity of the problem, the equation cannot be differentiated and I must rely on a numerical approximation of the weights. Therefore, an iterative method is ideal to minimize error in the model prediction.

Gradient Descent is a popular iterative optimization method. Gradient descent is the act of iteratively using training data on a model to calculate the error of the model's prediction. Gradient Descent then uses the gradient of the loss function to adjust the weight parameters of the model. However, in a neural network, computing the gradient of the loss function requires additional consideration as the model has multiple layers and nodes. A process known as backpropagation is performed to adjust the weight of a model. This method iterates from the output backwards to adjust the weights of the model to minimize the error. This way, gradient descent was able to be utilized to train the method.

One parameter that is important for the gradient descent process is the learning rate. The learning rate determines the amount that the gradient descent algorithm can change in an iteration. A large learning rate can result in an ability to miss the global minimum, while a small step size will result in a longer training process and may get lost in local minimum. Therefore, picking a learning rate is very important to configuring the training process. However, it can be difficult to find a valid learning rate and usually involves trial and error. As such, it is quite popular to utilize dynamic optimization algorithms that will dynamically modify the learning rate based on the results of the previous gradient descent iteration. Adam is an optimization algorithm that adaptively changes the learning rate of the algorithm based on past gradients (Kingma et al.,

2015). Adam is popular for quickly minimizing loss. However, it is typically not as accurate as slower methods. This trade-off is acceptable for this study.

The final key element in this method is to determine a loss function. A loss function describes the amount of error that is created by the prediction. Ideally, a perfect model would minimize the loss function to 0. Since I are working with a regression problem, I can directly take the difference in the predicted and real values. From there, I can modify the difference in to accommodate for specific statistical trends. The loss function for this model is Mean Squared Error. Mean Squared Error (MSE) is found from:

$$MSE = \frac{1}{N}\sum_{i}(y_i - \hat{y}_i)^2 \tag{3}$$

Where N is the number of elements, i is each element, $\hat{y}_i$ is the training value for each element, and y_i is the predicted value for that element. Therefore, this loss function is essentially the average squared difference of the true and predicted value for each element. This function is common in training and evaluating regression models. Mean Squared Error is especially good at punishing predictions with large amounts of error. Since I want to see all predictions within a reasonable threshold of the real value, MSE is ideal. The goal of this algorithm is to then find a minimum value for the MSE of the river height predictions.

To begin the training process, I start by normalizing the water stage height data by shifting each value such that the mean is 0. Then, images are loaded with the random augmentations applied upon load. After initializing the modified VGG-16 regression model with ImageNet weights, I can then initiate the iterative method. Utilizing Gradient Descent, Adam, and Backpropagation, I iteratively fit the training set to the model. After each epoch, I then make

predictions on the validation set. The MSE results for the training set and the validation set are written before epoch (iteration) begins. This process continues for 20 epochs. After training, the model weights that minimize validation loss 20 epochs are saved and evaluated further. This implementation was created in Python using TensorFlow and Keras. Training was performed using CUDA on a NVIDIA GeForce RTX 2080 Super XC Ultra from EVGA with 8 GB of VRAM.

IV. Results

Training

When training, it is important to analyze the loss value during each epoch. As an iterative process, training a neural network should result in a decrease in error over time. Training using gradient descent seeks to optimize weights to minimize the loss function. Therefore, a significant decrease in error over time in a training process helps to indicate that the model is capable of fitting to a dataset. In addition, the stability of a minimizing loss is important too. If the loss value sees significant spikes while training, those issues can indicate that the model is struggling to escape local minimum and converge. Thus, seeing a loss value that is maintaining a relatively stable descent towards minimum will indicate that a model is fitting well to the data.

The figure below describes the results of the training process for each dataset, i.e., the change of MSE with training epochs. One epoch is measured by a full passthrough of the training dataset. After each epoch, the validation dataset is also evaluated for comparison. Therefore, in each training run, the figure demonstrates how the model will reasonably train to minimize error on each dataset. In the Clear Creek dataset, the minimum MSE score for the validation set is 0.0046 ft^2 after 20 epochs. For the Milwaukee River dataset, the final MSE is 0.0152 ft^2. On the Auglaize River, the final MSE score is 0.0116 ft^2.

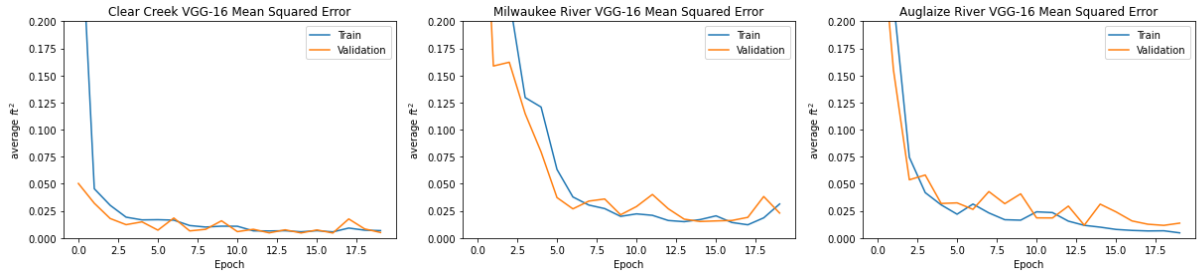

Figure 5: Training Mean Squared Error over time

Mean Squared Error of image regression experiment over each epoch. (blue: training dataset error, orange: validation dataset error)

With the highest amount of image data, the model was able to work towards minimizing error on the Clear Creek dataset efficiently. The training process had major decreases in loss in the first few epochs. This convergence shows that the models were able to easily find features in the dataset that correlates with the river stage. This dataset was considered the "best case" scenario regarding finding visible markers, and this assumption was confirmed by the convergence curve shown in the figure.

Compared to the Clear Creek case, it appeared that the Auglaize River dataset required more epochs for the model to converge. As it was expected that due to the obstructed view of the river, the training process requires more time to train and find the best features. In addition, it is possible that the relative lack of data compared to Clear Creek could slow down the training process as well.

On the Milwaukee River dataset, the slowest rate of convergence occurs. The model struggled to find the correct features. In addition, there is a greater difference in the training and the testing dataset. As such, the model is also struggling to make general predictions on the dataset. Certain features, such as the shallow camera angle and the lost shoreline in higher levels,

may have increased difficulty in this model's ability to train. However, the results are still decent as the error is within a reasonable level of precision for the scope this study.

From the figures, one encouraging feature is the stability of convergence over time. If an epoch results in a higher MSE than the previous epoch, then the model had previously chosen incorrect features to focus on. The model can still escape a local minimum due to incorrect features by targeting a global minimum over iterations. However, if the model has many "spikes" in loss, this trend may indicate a general difficulty finding a global minimum. This difficulty can lead to a slower training speed and a possible selection of unreliable features. Therefore, it is clear that the VGG16 model, while containing a few "spikes", maintains a consistent downward trend towards the global minimum. It is clear that this model creates accurate prediction with average error within a few tenths of a feet. Therefore, the training process is effective at fitting the image data to predicting water stage.

Saliency Mapping

When working with a trained DNN, it is often difficult to understand how a model generates a prediction. The dense nature and high parameter count of a DNN lead most researchers to consider a model as a "black box". In the case of this study, it is nearly impossible to understand what features in the image are influencing the river stage prediction by simply observing the weights and output of a model. Therefore, understanding how a model is trained requires one to employ a technique that will simplify the trained features of a model into an understandable and reasonably accurate report.

For imaging DNNs, one popular method for visualizing a model is saliency mapping (Simonyan et al., 2014). Saliency mapping can be utilized to generate a "heatmap" of the most

important pixels in an image. Therefore, by analyzing the heatmap, the features that are highly weighted by the models prediction can be identified. The method is generated by taking an input image and the generated river stage prediction for that image. For each pixel, the derivative of the stage prediction given the pixel can be evaluated.

Simonyan et al. (2014) define the derivative w, given a score S, and an image I:

$$w = \frac{\boldsymbol{dS}}{\boldsymbol{dI}}. \tag{4}$$

The derivative w is a three-channel matrix with the same shape as the image I. Given w, the map M is calculated from:

$$M = max_c(w_c) \tag{5}$$

where $c = 1,2,3$ indicating the three channels.

A saliency map can be useful to understand how the model makes a prediction on a single image. To assess statistically important features, a mean saliency map can be calculated. With the river stage prediction model, saliency maps were generated for all of the images in the validation dataset for each location. Then, for each location, the mean value for each pixel across every saliency map is calculated to generate a mean saliency map. Figure X shows the mean saliency maps for the Clear Creek, Auglaize River, and Milwaukee River sites:

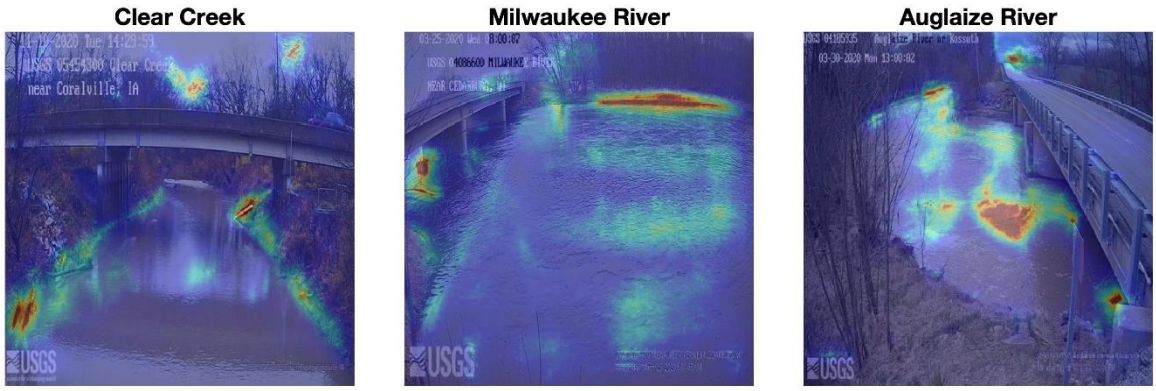

Figure 6: Mean Saliency Maps on testing sites

Mean Saliency Maps for each of the experiment sites. Each heatmap includes a randomly selected superimposed image to show a general correlation with the features of a given image. Blue indicates that the pixel is not important to the prediction, while red indicates a high level of importance on determining water stage height. (left: Clear Creek, center: Milwaukee River, right: Auglaize River)

For the Clear Creek mean saliency map, the primary set of activated pixels corresponds to the shoreline in the image. The shoreline is apparently a good indicator of the water surface elevation. Therefore, by focusing on the shoreline, it can be interpreted that the model is correctly choosing a feature that is relevant to the river stage. On the Auglaize River mean saliency map, it appears that the heatmap flares in the center of the river. These pixels correspond to a raised spot of land in the water body. By looking at how much of this spot of land is covered, the model may determine is the water surface level. In the Milwaukee River mean saliency map, the heatmap is again activated along the far shoreline of the image. In addition, the heatmap activates near the overpass bridge in the image. The amount of exposure in the bridge pier will also correlate with the water surface level. In summary, most activated features in the mean saliency maps appear to correlate with visual cues any human observer would rely on to estimate the river stage.

In all three mean saliency maps, a fair number of activated pixels that may not seem to correlate with river stage. It is possible that these activated features act as constant markers in

every image. The usage of these markers may aid the model in determining distance, scale, and rotation. However, it is also possible that these activated pixels may be noise resulting from the training process. Further training and experimentation with increased data size may reduce the influence of noise on the saliency maps.

Deployment

With a trained model, it is important to consider how that model will react to foreign data. As a proposed remote stream gaging method, the water stage image regression model must react well to new data as it is deployed. Therefore, a deployment experiment will indicate the efficacy of predictions over the course of a period. Since the model was trained on the training set, it is expected for the deployment set to experience a decrease in accuracy. However, the error on the deployment set should still be within a reasonable threshold compared to the loss on the training set.

The VGG-16 models trained with the "before" image set were applied to predict river stage with the "after" set at each site. Results of the "after" set are presented in this study as a demonstration of deployment of the DNN model for predicting future river stages with images. Figures 6~8 show time series of stage measurements (black line), and results of training (blue dots) and prediction (red dots) between March 2020 and January 2021 at the three test sites. Relations between the measured gage data and training/prediction results are also shown in these figures, along with a 1:1 relation (dashed line). In general, deployment tests demonstrated solid performance of the trained model at all three sites, except for a few "outliers". Taking the square root of the MSE as an indicator of "error" for model performance, "errors" of training results were 0.068 ft, 0.124 ft, and 0.108 ft for Clear Creek, Milwaukee River and Auglaize River,

respectively. “Errors” of deployment results grew slightly, which were 0.090 ft, 0.180 ft, and 0.150 ft, respectively. Increased prediction “errors” compared to the training may partly attributed to the seasonal change of surrounding environment and riparian vegetation. The “errors” of prediction at the Auglaize River site showed greater “excursions” in the stage range between 8 and 10 ft, where very few data existed in the training set (see Figure 2). The variation of “errors” among the three test sites reflected the expected level of challenges according to visual inspection of images on features related to river stage estimation. It is generally expected that both training and prediction “errors” can decrease as data size increases with more images available at various river stages.

It should also be noted that the same experiments were conducted without pre-processing images by random shift, rotate and shear transformations. It was found that training and prediction “errors” were not substantially different from those with transformations. This suggests that the DNN method is insensitive to random perturbations to the imaging system. With a conventional image processing approach, one may extract geometric information, such as the river stage, through explicit feature detection. Additional processes such as image registration are required to account for disturbance, which are generally specific to the scene captured.

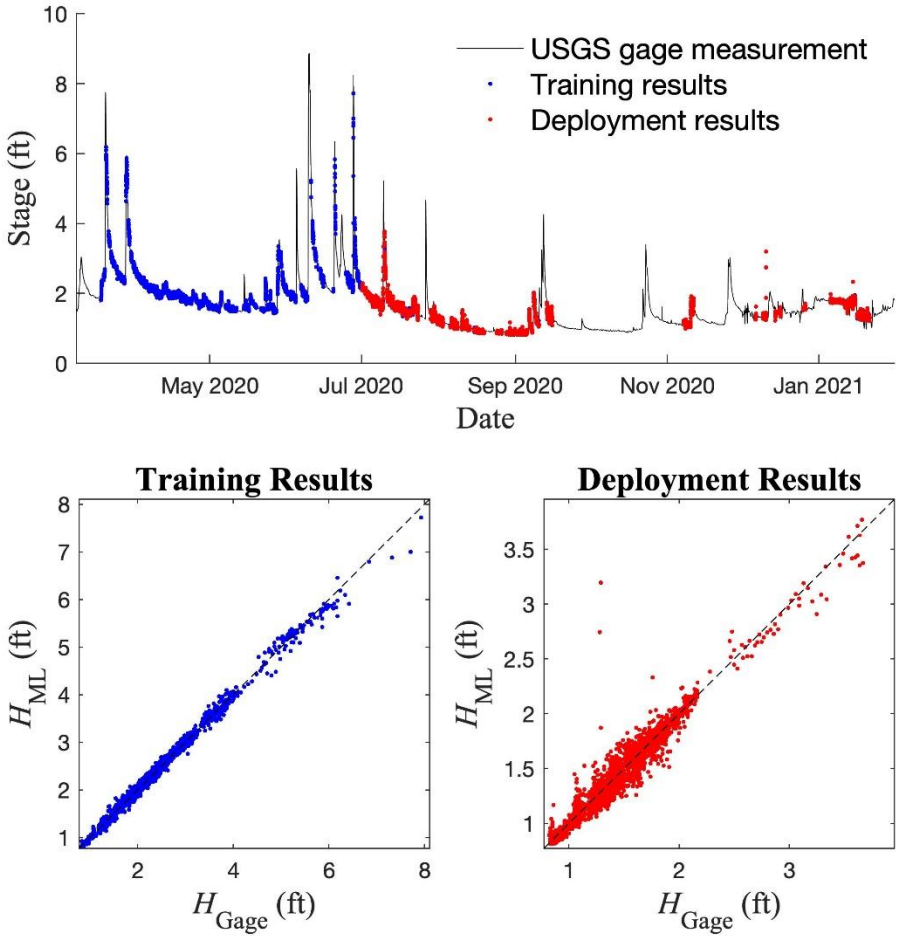

Figure 7: Clear Creek Deployment Timeseries

Time Series and Gage vs ML graph for the Clear Creek dataset. The Time Series graph outlines predictions made between March 2020 and January 2021. Training results (blue dots) of the "before" set separates with prediction results (red dots) on July 1st, 2020. Gage vs ML graphs show the difference in the actual stage height value (Gage) and the predicted value (ML). No error in a prediction would result in the plot lying on the dotted line. (top: Time Series, bottom-left Training set Gage vs ML graph, bottom-right: Deployment set Gage vs ML graph)

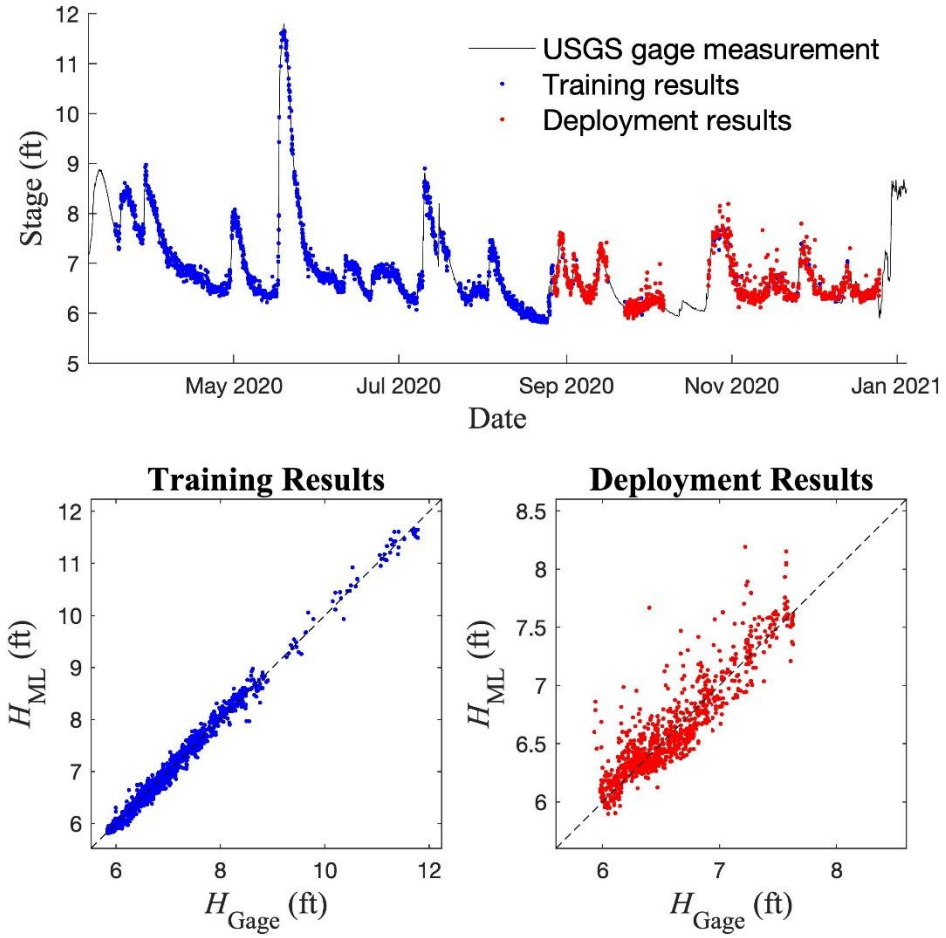

Figure 8: Milwaukee River Deployment Timeseries

Time Series and Gage vs ML graph for the Milwaukee River dataset. The Time Series graph outlines predictions made between March 2020 and January 2021. Training results (blue dots) of the "before" set separates with prediction results (red dots) on August 27th, 2020. Gage vs ML graphs show the difference in the actual stage height value (Gage) and the predicted value (ML). No error in a prediction would result in the plot lying on the dotted line. (top: Time Series, bottom-left Training set Gage vs ML graph, bottom-right: Deployment set Gage vs ML graph)

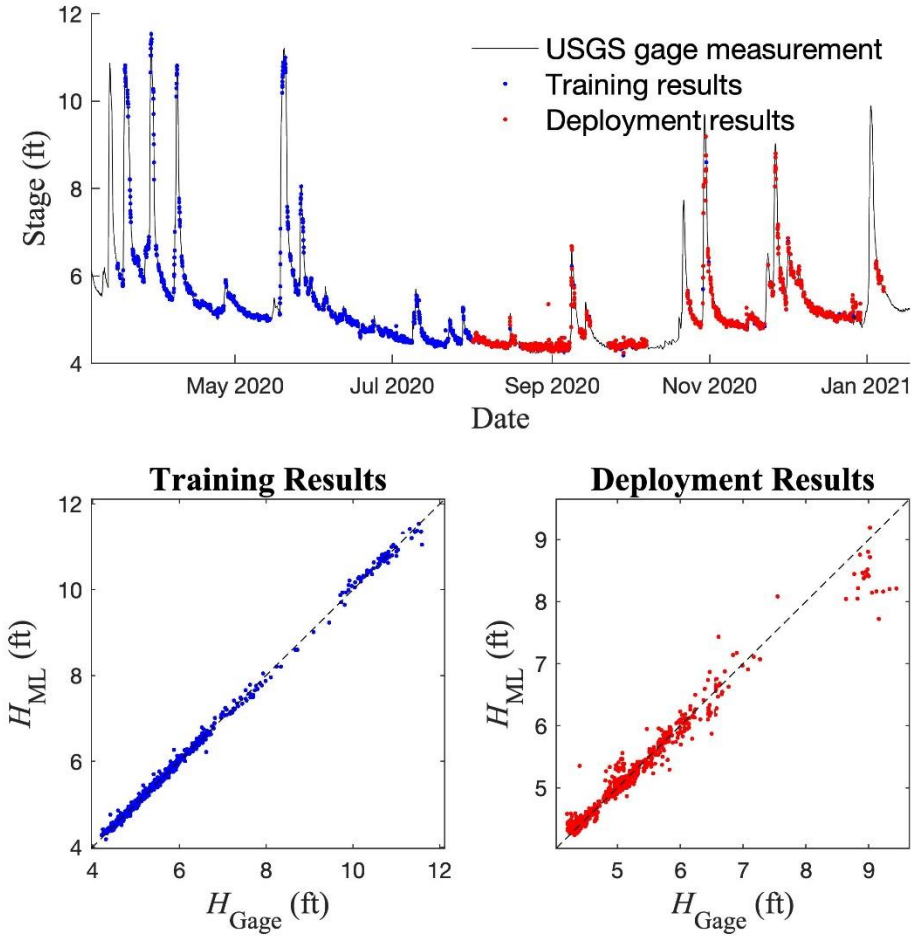

Figure 9: Auglaize River Deployment Timeseries

Time Series and Gage vs ML graph for the Auglaize River dataset. The Time Series graph outlines predictions made between March 2020 and January 2021. Training results (blue dots) of the "before" set separates with prediction results (red dots) on August 1st, 2020. Gage vs ML graphs show the difference in the actual stage height value (Gage) and the predicted value (ML). No error in a prediction would result in the plot lying on the dotted line. (top: Time Series, bottom-left Training set Gage vs ML graph, bottom-right: Deployment set Gage vs ML graph)

As expected, the deployment set predictions are not as accurate as the predictions on the training set. However, all the deployment set predictions still correlate with the general trend of the time series. As noted earlier, the error increase between training and deployment sets is reasonable. Therefore, from this deployment experiment, it is clear that the model created will continue to create relatively accurate predictions on new data. This result increases the viability in this method as a remote stream gaging method.

Limitations

In selecting the three sites for this study, while locations were selected that may have a considerable amount of difficulty, the sites had clear and identifiable features correlating with water height. In the USGS stream gage network, there were many sites where the angle and location of the webcam do not allow for easy identification for the water height. Also, the three sites chosen were in the Midwest and may not be subject to weather conditions that affect stream gage locations in other regions. Therefore, issues may arise when utilizing this method as-is on these potentially troubling areas.

One limitation of the DNN method is its inability to extrapolate output. As noted, the model can only predict conditions it has already seen. Therefore, in the case of an extreme weather condition, the model will only predict results close to the most representative training sample. In this case, error in the model's prediction can be quite large as the extreme weather exceeds the range of known values. Consideration of the range of prediction in a trained model will be important as the model can only fit according to the data it is trained to.

Considering the DNN model, many shortcuts were made to achieve these results. By utilizing transfer learning and training parameters intended to find fast convergence, it is likely

that this method has no optimally minimized error. The shortcuts may have harmed the possible accuracy upside of the DNN model in service of increasing the likelihood for finding plausibility as a remote stream gage method. This trade-off was important to show this method as a proof-of-concept idea. However, it is likely that better results are possible.

V. Conclusion

This study serves as a proof-of-concept that river stage can be directly predicted from static images. Specifically, it was demonstrated that, for a sub-optimal dataset, the transfer learning approach, i.e., a Deep Learning model designed for a separated problem domain, was able to identify image features that are relevant to the river stage, thus predict its value following a regression model. The presented DNN regression model on stream gage webcam data shows potential as a general non-contact method of measuring river flow data, which minimizes the cost and hazards associated with the current operation. As a data-driven approach, the supervised training procedure can be universally applied to other sites without specific knowledge, parameter tuning, or manual calibration of a specific site. The study also demonstrated that the DNN regression approach is robust against random perturbations to the imaging system as the output river stage values are insensitive to artificially generated random translation, rotation, and shear. These features simplify river stage/discharge measurements, compared to other imaging-based solutions, such as the LSPIV method. This study demonstrates the strength of machine learning, i.e., a "black-box" approach that directly link image input to a desired output as long as sufficient data are available for the learning process.

The idea of utilizing Image Regression on stream gaging webcam data shows potential as a general, remote, method of measuring flow data. A DNN Regression model has the potential to create accurate measurements on high definition image data. With this method, one can safely gather data at scale without the need for physical measurements. This method will increase the ease of creating consistent remote Stream Gaging solutions for various locations. As a general method, given the data, models that report on many different locations can be created with

similar parameters. This approach will allow us make insight from detailed stream gage data without the limitations of LSPIV methods. Therefore, this study and future work on this topic will lead to the development of a generalized, remote, river gaging method that minimizes the cost and hazards of current methods.

Future Work

Going forward, increasing the diversity and quality of datasets utilized on this method will help uncover any potential issues and increase prediction accuracy. As stated previously, possible limitations arose in the sites chosen for this study. Therefore, the next step would be to perform the water height image regression method on more difficult sites. It is important to determine the effectiveness of this method as it becomes harder to find distinct features that correlate with water height. This experiment would reveal potential issues with this method and open the door for including other Image Processing methods to increase the clarity of important features. In addition, this study utilized data from March 2020 to January 2021. Therefore, while all seasons are represented in the dataset, there may be features exclusive to the month of February that may cause issues. Also, one year of data will not be sufficient for possible extreme weather scenarios. Therefore, collecting more data to create larger datasets will help increase the possible range of prediction, and fully represent the location for future deployment.

There are a few paths to improve our methodology for future research. The first path would be to rethink our model implementation. One possible consideration would be to train our models without any pre-trained weights inherited ImageNet. While the training process may fail and cost relatively more resources, untrained starting weights may help the model see features that would be ignored when pretrained on ImageNet. Therefore, studying untrained weights may

increase accuracy in the long run. Additionally, if a larger sample of images are available, designing a new deep neural network may that would suit this problem better. The transfer learning models are designed for recognizing features that are more complicated than those in webcam images of a stream gage. As such, designing a simpler model may increase performance while having minimal effect on error.

Future research may consider additional methods that could augment the image classification regression model. One such method can involve the consideration of water surface patterns in motion video data to add another dimension of river flow hydraulics. Similar to the concept of LSPIV in which river surface flow speed can be resolved from a sequence of images, a DNN regression model on multiple frames of a motion video may have a stronger correlation with hydraulic parameters including stage, water depth and discharge. On the other hand, a DNN model can be implemented as a preprocess of a LSPIV approach. Models trained from machine learning may help to automate processes such as river surface segmentation, geometric calibration, and image orthorectification. It is also possible to resolve surface velocity distribution directly with DNN learning. Future research may also explore applications of DNN models on estimating river water qualities from images, such as the turbidity, BOD, and colored dissolved organic matters (CDOM), etc.

References

Adamala, S. (2017) An Overview of Big Data Applications in Water Resources Engineering. Machine Learning Research, 2(1), 10-18. doi:10.11648/j.mlr.20170201.12.

Bjerklie, D. M., Dingman, S. L., Vorosmarty, C. J., Bolster, C. H., & Congalton, R. G. (2003). Evaluating the potential for measuring river discharge from space. Journal of Hydrology, 278(1-4), 17-38.

Chaudhary, P., D'Aronco, S., Moy de Vitry, M., Leitão, J. P., & Wegner, J. D. (2019)._ Flood-Water Level Estimation from Social Media Images, ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, 42, 5–12. doi:10.5194/isprs-annals-IV-2-W5-5-2019.

Chen, Y., Fan, R., Yang, X., Wang, J., & Latif, A. (2018). Extraction of Urban Water Bodies from High-Resolution Remote-Sensing Imagery Using Deep Learning". Water 10(5), 585. doi:10.3390/w10050585.

Cho, E., Jacobs, J. M., Jia, X., & Kraatz,S. (2019). Identifying subsurface drainage using satellite big data and machine learning via Google Earth Engine. Water Resources Research, 55, 8028-8045. doi:10.1029/2019WR024892.

Costa, J. E., Cheng, R. T., Haeni, F. P., Melcher, N., Spicer, K. R., Hayes, E., Plant, W., Hayes, K., Teague, C., & Barrick, D. (2006). Use of radars to monitor stream discharge by noncontact methods. Water Resources Research, 42(7).

Deng, J., Dong, W., Socher, R., Li, L., Li, K., & Fei-Fei, L. (2009) ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition, 248-255. doi: 10.1109/CVPR.2009.5206848.

Durand, M., Gleason, C.J., Garambois, P.A., Bjerklie, D., Smith, L.C., Roux, H., et al., (2016). An intercomparison of remote sensing river discharge estimation algorithms from measurements of river height, width, and slope. Water Resources Research, 52 (6), 4527–4549. https://doi.org/10.1002/2015WR018434

Fischer, P., Dosovitskiy, A., & Brox, T. (2015). Image orientation estimation with convolutional networks. German Conference on Pattern Recognition, 368-378. doi:10.1007/978-3-319-24947-6.

Hu, P., Tong, J., Wang, J., Yang, Y., & d. Oliveira Turci, L. (2019). A hybrid model based on CNN and Bi-LSTM for urban water demand prediction. 2019 IEEE Congress on Evolutionary Computation (CEC), 1088-1094. doi: 10.1109/CEC.2019.8790060.

Jafari, N.H., Li, X., Chen, Q., Le, C.-Y., Betzer, L.P., & Liang, Y. (2021). Real-time water level monitoring using live cameras and computer vision techniques, Computers and Geosciences, 147, doi: https://doi.org/10.1016/j.cageo.2020.104642.

Jin, T., & Liao, Q. (2019). Application of large scale PIV in river surface turbulence measurements and water depth estimation. Flow Measurement and Instrumentation, 67, 142-152.

Johnson, E. D., & Cowen, E. A. (2016). Remote monitoring of volumetric discharge employing bathymetry determined from surface turbulence metrics. Water Resources Research, 52(3), 2178-2193.

Kingma, D., & Ba, J. (2015) Adam: A Method For Stochastic optimization. International Conference on Learning Representations, arXiv:1412.6980.

Kouraev, A. V., Zakharova, E. A., Samain, O., Mognard, N. M., & Cazenave, A. (2004). Ob'river discharge from TOPEX/Poseidon satellite altimetry (1992–2002). Remote sensing of environment, 93(1-2), 238-245.

Kreyenberg, P. J., Bauser, H. H., & Roth, K. (2019). Velocity field estimation on density-driven solute transport with a convolutional neural network.Water Resources Research, 55, 7275–7293. doi:10.1029/2019WR024833.

Lang, M., Pobanz, K., Renard, B., Renouf, E., & Sauquet, E. (2010). Extrapolation of rating curves by hydraulic modelling, with application to flood frequency analysis. Hydrological Sciences Journal–Journal des Sciences Hydrologiques, 55(6), 883-898.

Lathuilière, S., Mesejo, P., Alameda-Pineda, X., & Horaud, R. (2020), A Comprehensive Analysis of Deep Regression. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(9), 2065-2081. doi: 10.1109/TPAMI.2019.2910523.

Le Coz, J., Renard, B., Bonnifait, L., Branger, F., & Le Boursicaud, R. (2014). Combining hydraulic knowledge and uncertain gaugings in the estimation of hydrometric rating curves: A Bayesian approach. Journal of Hydrology, 509, 573-587.

Legleiter, C. J., Kinzel, P. J., & Nelson, J. M. (2017). Remote measurement of river discharge using thermal particle image velocimetry (PIV) and various sources of bathymetric information. Journal of Hydrology, 554, 490-506.

Li, W., Liao, Q., & Ran, Q. (2019). Stereo-imaging LSPIV (SI-LSPIV) for 3D water surface reconstruction and discharge measurement in mountain river flows. Journal of Hydrology, 578, 124099.

Ling, F., Boyd, D., Ge, Y., Foody, G. M., Li, X., Wang, L., et al. (2019). Measuring river wetted width from remotely sensed imagery at the subpixel scale with a deep convolutional neural network.Water Resources Research, 55, 5631–5649. doi:10.1029/2018WR024136.

Lopez-Fuentes, L., van de Weijer, J.,Bolanos, M., & Skinnemoen, H. (2017). Multi-modal deep learning approach for flood detection, Working Notes Proceedings of the MediaEval 2017 Workshop, 1–3. urn:nbn:de:0074-1984-7.

Melcher, N. B., Costa, J. E., Haeni, F. P., Cheng, R. T., Thurman, E. M., Buursink, M., ... & Hayes, K. (2002). River discharge measurements by using helicopter-mounted radar. Geophysical Research Letters, 29(22), 41-1.

Miao, Z., Fu, K., Sun, H., Sun, X., and Yan, M. (2018). Automatic Water-Body Segmentation From High-Resolution Satellite Images via Deep Networks. IEEE Geoscience and Remote Sensing Letters, 15(4), 602-606. doi: 10.1109/LGRS.2018.2794545.

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

Norris, J. M., Lewis, M., Dorsey, M., Kimbrough, R., Holmes, R. R., & Staubitz, W. (2007). Qualitative comparison of streamflow information programs of the US Geological Survey and three non-federal agencies. US Geological Survey Open-File Report, 2007, 1426.

Pan, B., Hsu, K., AghaKouchak, A.,& Sorooshian, S. (2019). Improving precipitation estimation using convolutional neural network.Water Resources Research, 55, 2301–2321. doi:10.1029/2018WR024090.

Pan, M., et al. (2020) Water Level Prediction Model Based on GRU and CNN. IEEE Access, 8, 60090-60100. doi:10.1109/ACCESS.2020.2982433.

Puleo, J. A., McKenna, T. E., Holland, K. T., & Calantoni, J. (2012). Quantifying riverine surface currents from time sequences of thermal infrared imagery. Water Resources Research, 48(1).

Ran, Q. H., Li, W., Liao, Q., Tang, H. L., & Wang, M. Y. (2016). Application of an automated LSPIV system in a mountainous stream for continuous flood flow measurements. Hydrological Processes, 30(17), 3014-3029.

Ribani, R., & Marengoni, M. (2019) A Survey of Transfer Learning for Convolutional Neural Networks. 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T), 47-57. doi: 10.1109/SIBGRAPI-T.2019.00010.

Simonyan, K., Vedaldi, A., & Zisserman, A. (2014) Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, International Conference on Learning Representations, arXiv:1312.6034

Simonyan, K., & Zisserman, A. (2015) Very deep convolutional networks for large-scale image recognition, International Conference on Learning Representations, arXiv:1409.1556.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2818-2826. doi: 10.1109/CVPR.2016.308.

Tauro, F., Petroselli, A., & Grimaldi, S. (2018). Optical sensing for stream flow observations: A review. Journal of Agricultural Engineering, 49(4), 199-206.

Turnipseed, D. P., & Sauer, V. B. (2010). Discharge measurements at gaging stations (No. 3-A8). US Geological Survey.

Wang, W., Yang, Y., Wang, X., Wang, W., Li, J. (2019). Development of convolutional neural networkand its application in image classification: a survey. Optical Engineering, 58(4), 040901. doi: 10.1117/1.OE.58.4.040901.