Aerial Fluvial Image Dataset for Deep Semantic Segmentation Neural Networks and Its Benchmarks

Zihan Wang and Nina Mahmoudian, *Member, IEEE*

***Abstract*—Classification of aerial imagery is essential for water channel surveillance and waterfront land cover characterization. It is also beneficial to long-duration collaborative autonomous navigation of both unmanned aerial vehicles (UAVs) and autonomous surface vehicles (ASVs) to fulfill unmanned hydrologic data collection, environmental inspection, and disaster warning tasks. Deep semantic segmentation networks trained on aerial imagery have shown great results, however, they require finely labeled data. Existing aerial image datasets contain mostly urban scenes or fluvial images taken from ground level or collected from the Internet, there are no datasets that incorporate aerial and fluvial scenes with detailed annotation from different perspectives or include waterborne obstacles. To tackle this problem, aerial fluvial image dataset (AFID) is presented with multiple camera perspectives of fluvial scenes and is semantically labeled with emphasis on water and waterborne obstacles. Deep neural networks for binary (water and nonwater) semantic segmentation, with 12 different combinations of five encoders and three decoding architectures, are trained and tested in a curriculum learning scheme. Model performance is benchmarked on AFID, and the accuracy-efficiency tradeoff is discussed with the conclusion that the Unet architecture with a mix transformer encoder achieves the best segmentation performance with moderate computational consumption. The AFID dataset is publicly available to facilitate future work on developing new lightweight semantic segmentation models. Our immediate future plan will focus on the coordination of air and surface-water autonomous systems for navigable water detection and obstacle avoidance in high-risk challenging environments.**

***Index Terms*—ASV, autonomous navigation, dataset, deep neural network, oceans and water, semantic segmentation, sensing platforms, UAV.**

I. Introduction

AUTONOMOUS surface vehicles (ASVs) are adept at autonomous mission planning, path following, and obstacle avoidance in open environments, such as lakes, oceans, and slow flowing rivers [1], [2], [3], [4], [5], [6], [7], [8], [9]. However, narrow or meandering rivers and creeks have flow that is more rapid, obstacle laden, and likely to change rapidly over geographic and temporal spans. Changes in water level along the course of a river and over times of heavy rain or prolonged drought exposes shoals, debris, downed trees, and other obstacles that pose difficulties for ASVs. Furthermore, the meandering nature of such rivers limits the line of sight of ASVs therefore limiting reaction and planning time for safe obstacle avoidance. In addition, ASVs also have difficulty in navigating braided river branches due to their limited line-of-sight distance. In a fluvial system comprised of multiple diverging and converging shallow channels that diverge and rejoin the main channel, autonomous agent without any *a priori* information must actively explore each channel until either an obstacle impedes the agents path or the main river branch is found.

To overcome the reaction latency of obstacle avoidance and the inefficiencies in river branch selection during autonomous fluvial environment traversal, camera equipped UAVs can be utilized to form global contextualized maps of fluvial scenes ahead of the planned ASV path. However, due to UAVs' inability to operate for a long duration of time, recharging and selective deployments is required for use as a solution to the ASV river branch selection problem. Cooperatively, the heterogeneous aerial and aquatic robotic system can fulfill long-range and long-duration robust autonomous fluvial navigation tasks with minimum human intervention. In addition, this cooperation is also beneficial for energy-efficient mapping from the water surface for open and easily navigable fluvial environments, and from the air for complex or obstacle laden fluvial scenes.

Such cooperation is not uncommon in the field of robotics. In the ground domain, heterogeneous cooperation between UAVs and AGVs already has been used for many applications, such as surveillance [10], [11], [12], rescue [13], environmental monitoring [14], [15], environmental mapping [16], and object transportation [17]. In the surface domain, the UAV-ASV cooperation schemas have been used in maritime surveillance [18], [19] and rescue missions in littoral [20] and flooded urban [21] environments. In inland waterway scenes the RIVERWATCH [22] system uses a UAV and ASV designed for automatic monitoring of riverine environments. Aerial imagery of fluvial scenes can be turned into 2-D fluvial maps with localized and identified paths and objects, which can be relayed to surface vehicles for use in path generation for various mission modalities.

While cooperative heterogeneous robotic fleets have shown promise in their ability to work together to navigate complex environments, the issue still remains how to interpret sensor data, such as visual imagery, into contextual information capable of being used for route planning. Traditional computer vision methods have been used to extract meaningful information from image data [20], [21] in such instances. However, deep learning methods require less expert analysis and fine-tuning, provide superior flexibility across domains, and can achieve greater

Manuscript received 3 February 2023; revised 13 April 2023; accepted 7 May 2023. Date of publication 11 May 2023; date of current version 29 May 2023. This work was supported by the Office of Naval Research (ONR) under Grant N00014-20-486 2085. *(Corresponding author: Nina Mahmoudian.)*

The authors are with the School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: wang5044@purdue.edu; ninam@purdue.edu).

Digital Object Identifier 10.1109/JSTARS.2023.3275068

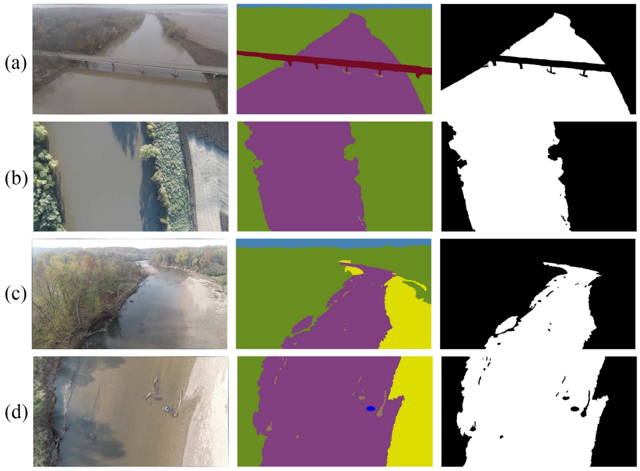


Fig. 1. RGB images captured from Wabash River and Wildcat Creek in slanted and nadir views with corresponding multiclass annotated masks and binary masks (water as 255 and nonwater as 0). Row **a** and **b**: Wabash River in slanted view and nadir view, Row **c** and **d**: Wildcat Creek in slanted view and nadir view. River (purple), boat (dark blue), bridge (red), sky (light blue), vegetation (green), dry sediment (yellow) and waterborne obstacle (brown) classes exist in these images.

accuracy in tasks like image classification, object detection, and semantic segmentation [23].

Due to the data-hungry nature of deep neural networks (DNNs), there is a growing need for annotated publicly available datasets to train and benchmark networks for various domains and tasks [24]. Due to the abundance of annotated publicly available datasets in urban environments, semantic segmentation has already shown potential in urban objects analysis and decision-making assistance, such as urban planning through lane markings [25], building extraction [26], [27], [28], and traffic and pedestrian monitoring [29]. However, annotated publicly available datasets containing aerial images of fluvial scenes that focus on detection of waterborne obstacles are exceptionally rare. Therefore, to unleash broader applications of semantic segmentation to the ASV autonomous navigation and river branch selection problems, this article presents a custom aerial fluvial dataset (shown in Fig. 1) that has eight classes, and is binarized to water and nonwater for all neural network training and testing.

The main contributions of this work are two-fold as follows: 1) A novel public aerial fluvial image dataset (AFID) that incorporates aerial and fluvial scenes with detailed semantic annotation from different camera perspectives. AFID contains 816 multiclass multiperspective semantically labeled images of two inland waterways with annotations for waterborne obstacles. 2) Building the performance benchmark for 12 advanced semantic segmentation models in curriculum training fashion on existing datasets and the subsets of AFID to highlight the capability of the presented dataset and evaluate learning capabilities for various segmentation architectures and feature-extracting encoders. AFID is essential for water channel surveillance, waterfront land cover characterization, and long-duration collaborative autonomous navigation of both UAV and ASV to fulfill unmanned hydrologic data collection, environmental inspection, and disaster warning tasks using deep semantic segmentation models. The binary segmentation benchmarks can serve as a reference for future multiclass segmentation networks to decrease the effort spent on model selection. AFID dataset can also be used to enrich the work on light-weight semantic segmentation model developments, so that real-time long-duration inference on unmanned autonomous vehicles is achievable.

The rest of this article is organized as follows: Pertinent existing datasets and their used semantic segmentation methods are reviewed in Section II. The detailed description of our dataset is in Section III. The experimental design and evaluation including adopted architectures, encoders, metrics, and benchmarks are detailed in Section IV. The discussion of experiment results is in Section V. Finally, Section VI concludes this article.

II. Related Work

For UAV to cooperatively assist ASV navigation through complicated branching fluvial scenes, UAVs must be able to accurately and routinely recognize water and nonwater entities within said scene. For semantic segmentation networks to classify water pixels from RGB images, they have to be trained on enough water-containing images to enable generalization to *in situ* imagery. However, due to possible significant difference in spatialwise and channelwise pixel distributions of water and nonwater pixels in images among various aquatic datasets, it is unreasonable to use many existing ground and surface datasets to train a network for a domain-specific task, such as aerial water segmentation. In Section II-A, RGB image datasets that are pertinent to ASV and UAV operating in and around bodies of water are introduced, as well as their applicability to the aerial fluvial semantic segmentation task. Finally, a brief investigation of semantic segmentation networks that have been used for similiar tasks is given in Section II-B.

A. Aquatic Semantic Segmentation Datasets

Multiple maritime and fluvial RGB image datasets for semantic segmentation of water exist due to the fact that water recognition and localization has many applications. However, many such semantic datasets serve various purposes in varying environments (e.g., littoral or blue water, harbor, rivers, lakes, etc.) and can have very different feature distributions. This makes it difficult for a neural network to learn and generalize well across all aquatic datasets. Thus many datasets are inappropriate for semantic segmentation network training for segmenting water in complex fluvial scenes due to differences in feature values and distributions. These differences narrow the datasets that are applicable to aerial fluvial imagery inference.

To the authors' best knowledge, existing fluvial datasets either contain images taken on ground level from riverbanks [30], are general low-resolution online images that contain water but also watermarks [31], [32], or do contain aerial fluvial imagery but with only single camera perspective and no annotations for waterborne obstacles [33], or do contain aerial images of rivers with waterborne floating ice but not other more common obstacles [34]. All these factors make the multiclass annotated aerial fluvial dataset developed within this work not only meaningful for water semantic segmentation applications, such as

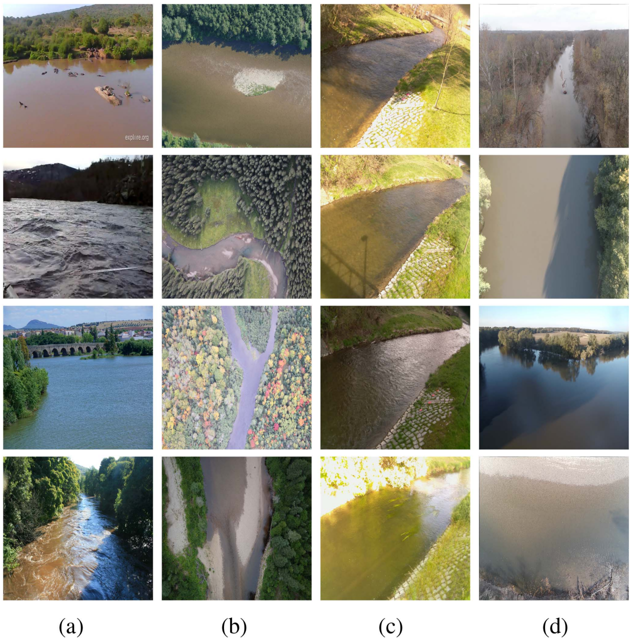


Fig. 2. Qualitative comparison of the three existing fluvial datasets used for pretraining (a)–(c) in this work with our aerial fluvial dataset (d). (a) Surveillance [31]. (b) 11 rivers [33]. (c) Lantern pole [30]. (d) AFID.

river branch investigation and selection but also unique in inland waterway autonomous navigation tasks.

Flood monitoring datasets [32], [35], [36] mainly contain images taken from ground level that have pixel label distributions skewed by urban backdrops and features that are uncommon in aerial fluvial imagery. Similarly, the ATLANTIS [37] dataset has been created for generic water resources management, but contain diverse scenes, such as waterfalls and wet roads, where ASVs cannot be possible to drive on, and thus, are not helpful for training networks to recognize obstacles toward fluvial navigation. Satellite image datasets for river regulation [38], while containing the applicable imagery perspective and contents, have spatial resolutions that are too large for visualization, annotation, and ultimately recognition of waterborne obstacles. Furthermore the distribution of water labeled pixels in satellite imagery is significantly lower than any imagery taken by drones in stable flight at altitude. Maritime [6], [39], [40], [41], [42], river [43], [44], and lakes and canal [45] datasets for ASV navigation and mapping have images collected only from the surface level, which share a common spatial object distribution with all boat or shore-based imagery: water in the lower, potential obstacles in the middle, and sky in the upper portions of images. This spatial pattern of surface images is diametrically opposed to aerial images of fluvial scenes [as shown in Fig. 2(d)], especially nadir views where water labeled pixels can propagate along any image axis in any orientation.

Multiple aerial semantic segmentation image datasets [29], [46], [47], [48], [49] collected by aircraft have been developed to advance deep learning models and their applications to object monitoring and tracking. However, these datasets focus primarily on urban or field scenes that contain negligible or no water pixels and are thus not applicable to semantic training of neural networks to segment water within fluvial imagery. Aerial image datasets of marine environments with labeled bounding boxes of waterborne objects also exist [50], [51], but only serve for object detection in surveys and interrogation tasks. They do not suffice for navigation tasks because water region cannot be represented by bounding boxes quite well. By contrast, semantic segmentation or pixelwise classification methods offer both feature classification as well as location information of all pixels in aerial imagery, and thus, yield both the navigable water regions of images and pixel locations of obstacles. Currently, [33] is one of, if not the only, aerial semantically segmented image dataset that focuses on fluvial scenes. The dataset developed within this work expands on [33] by using not only nadir view images but also forward looking ones to diversify the spatial composition of pixel classes, thus reducing the possibility of model overfitting. Moreover, our dataset contains finely annotated pixel boundaries for all classes, especially waterborne obstacles, and binary segmentation masks for the simplified classification of water and nonwater entities (Section III).

Three currently existing fluvial datasets are applicable to aerial fluvial semantic segmentation that can be beneficial for inland waterway navigation tasks. These three networks were used for pretraining of several deep semantic segmentation networks (Section IV). A qualitative comparison between these datasets and the dataset developed for this work is shown in Fig. 2. The first fluvial scene image dataset used is not publicly available but is available upon request from the authors. Lopez et al. [31], [52] collected fluvial images from Google, surveillance cameras, and other sources. The resulted dataset of 300 images has large variances of water color, turbulence, angle, and illumination, Fig. 2(a). Although the dataset is binary labeled (water and nonwater) and contains watermarks on some of the images, it was still used for pretraining due to its abundance in varying river scenes and elevated perspectives. The second airborne fluvial image dataset [33], [53] contains 1223 images spanning 11 rivers from 5 countries with five land-cover classes: water, dry exposed sediment, green vegetation, senescent vegetation, and roads, Fig. 2(b). After filtering out the mislabeled or coarsely labeled images of this dataset, 535 images from 5 rivers were used as part of our pretraining set. The final dataset used for pretraining [54] proposed by [30] was collected by a camera affixed to a pole along a river bank over a one year time span. The dataset contains 20 309 manually binary-labeled masks for images, Fig. 2(c). Although the dataset has significant temporally related variances in imagery, the spatial variance across dataset images is significantly smaller than other comparable datasets due to the fixed nature of the data recording device. Considering the potential spatial homogeneity of the images in this dataset, 512 images were randomly extracted from the original dataset's validation subset for use as part of our pretraining set.

B. Semantic Segmentation Methods

For semantic segmentation networks, the output mask has the same resolution with the input image, and each pixel is classified according to a category. Classical machine learning methods like support vector machines (SVM) [55] and random forest [56] were largely used to do image segmentation, until deep learning methods prevail by comprehensive improvements in performance. As for deep semantic segmentation networks, instead of using fully connected layers at the end of other classification networks [33], whose dimension depends largely on input image size, fully convolutional networks (FCN) [57] replaces those layers with pure convolutional layers, and uses pooling information from downsampling layers to learn nonlinear upsampling with transposed convolution. The downsampling part of a network is the encoder and the upsampling part is the decoder.

Networks, such as SegNet [58] memorize the pooling indexes in all encoder layers and use them for exact upsampling in decoder layers without learning, and have been used in [30], [32] for water segmentation. Although our objective is to find a memory and computation efficient network, we still care a lot about model accuracy, especially boundaries accuracy of water and waterborne obstacles. Thus, network architectures were tested that use more information from encoding layers when decoding, such as Unet [59], PAN [60], and DeepLabV3+ [61] (Section IV-B2). Moreover, different encoder structures show different feature extraction abilities [32], thus five promising encoding methods were chosen to conduct experiments, including four convolution-based and one vision transformer-based encoders. These encoding methods are discussed in Section IV-B1 in detail.

III. Aerial Fluvial Image Dataset

Aerial fluvial videos were collected by a SplashDrone 4 with waterproof GC3-S camera at a 60-Hz frame rate in both 2720 × 1536 (2.7 K) and 2560 × 1440 (2 K) resolutions during the fall of 2021. Videos were taken on segments of the Wabash River and Wildcat Creek in Indiana, USA. More specifically, the Wabash River data were collected on a river segment to the east of Prophetstown State Park (~3.65 km), as well as at the intersection of the Wabash River and Grant Road. Wildcat Creek data were collected between Wildcat Creek Park to Peters Mill Access (~4.71 km), at Mis-So-Lah Public Access Point, and at the intersection of Wildcat Creek with Schuyler Avenue.

Data collection along river segments was done by a team of two people. While one person controlled the water-proof drone to follow the course of the river under FAA guidelines, the other person controlled a chase boat in pursuit of the drone to maintain visibility. The data collection process at intersections of river/creek and roads was done by a human pilot standing on the river bank and maneuvering the drone to fly over the bridge in each fluvial scene.

The AFID dataset[1] [62] contains 816 images that were extracted manually from videos recorded on the SplashDrone. Of the 816 images 303 images were from the Wabash River and 513 images were from Wildcat Creek with both river sets containing subsets of forward-looking (slanted view) and downward-looking (nadir view) perspectives, Table I shows the number of both the river and perspectives subsets present in the AFID dataset.

TABLE I
Image Number of River Subsets and Drone Camera Perspectives

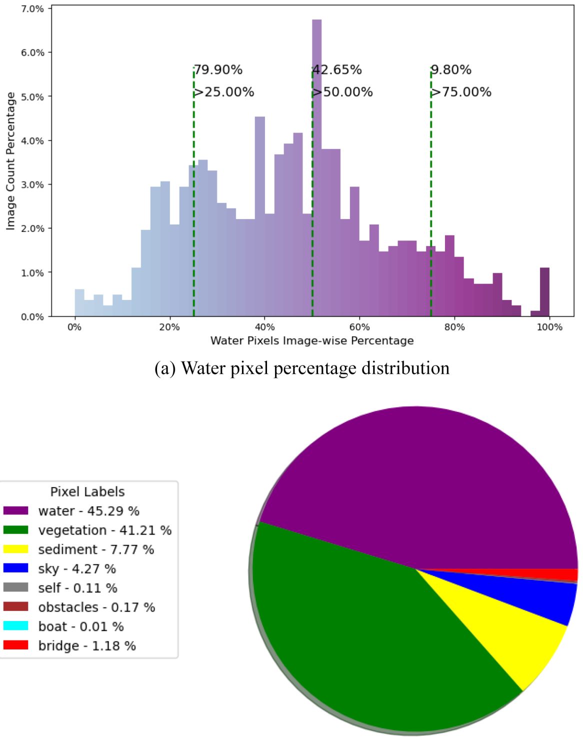
	Wabash River	**Wildcat Creek**	Total
Slanted View	160	299	459
Nadir View	143	214	357
Total	303	513	816

The AFID dataset contains images that vary in water color and texture due to sunlight, water turbulence, and shadows from trees and other objects as shown in Fig. 2(d). There are also a small number of fully and partially blurred images within the dataset from drone motion and lens moisture, respectively. As the SplashDrone and many other aquatic drones can land and take off from water, camera immersion, and extraction from water typically leaves residual moisture on the camera lens surface (i.e., water droplets and steaks). The inclusion of both crisp and blurred images is intended to increase trained network generalizability, which may encounter similar images during deployment in aquatic environments. In addition to image complexities relating to water hue, image crispness, and view differences, there are also significant amounts of waterborne obstacles common to fluvial environments, such as downed tree branches, exposed sand bars and sediments, and rocks in the Wildcat Creek subset. The Wabash River subset stands in contrast to the Wildcat Creek subset as it contains images of wider, deeper, less obstacle ridden, and less turbulent waters. Both subsets have nadir and slanted views of fluvial scenes and share the same set of semantic classes. The combined subsets serve to broaden the limited scope of existing aerial fluvial datasets with more camera perspectives and fluvial environments with specific attention on obstacles that pose hazards to surface-based watercrafts.

All images within the AFID dataset were labeled manually using the public Pixel Annotation Tool [63] with minor modifications of the annotated semantic classes to contain classes for river, boat, bridge, sky, forest vegetation, dry sediment, drone (self), and fluvial waterborne obstacles. Sample annotated images are shown in Fig. 1.

Semantic datasets can suffer from imbalances in class pixel percentages. Being aware of any dataset class imbalances is important as it can have a significant impact on the loss function, and further trained network performance [64]. In binary segmentation (water and nonwater in our case) the percentage of target class pixels usually has a positive correlation with the network's ability to correctly recognize and segment pixels of that class in input images. In our dataset the water class occupies about 45% which means that approximately 55% of pixels are nonwater. In multiclass segmentation the obstacles class takes up 0.17%, as shown in Fig. 3(b). Among the whole dataset, nearly 10% of images have a water pixel ratio over 75%, nearly 42% of images have a water pixel ratio over 50%, and about 80% of images have a water pixel ratio over 25%, as shown as Fig. 3(a).

[1] https://purr.purdue.edu/publications/4105/1



(a) Water pixel percentage distribution

(b) Pixel labels distribution

Fig. 3. (a) Demonstrates the imagewise water pixel percentage distribution, and proportions of images with water pixel percentage larger than three percentiles (25%, 50%, 75%) over the whole dataset. (b) Illustrates the datasetwise pixel distribution of all 8 classes.

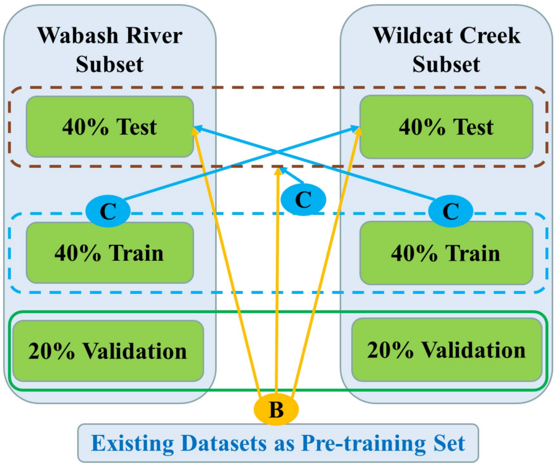


Fig. 4. Pretraining and continue-training diagram of datasets and models. Rectangle boxes denote datasets, where solid-line box around 2 validation sets means they are grouped as a whole for all trainings, and dashed-line boxes mean the subsets can be separate or combined in different trainings. Ellipses sitting on top of different boxes denote models trained on that dataset, where **B** stands for pretrained baseline models, **C** stands for continue-trained models on different training subsets, and arrows point to the subsets that are tested upon.

IV. Experimental Design and Evaluation

While the AFID dataset contains variance in scenes across different fluvial systems, times of day, camera perspectives, and scene compositions, it cannot provide training instances for all potential encountered fluvial images. Therefore the generalizability of neural networks, specifically convolution-based semantic segmentation networks, is leveraged to enable globally applicable learning from locally contextualized datasets (Midwestern US). To evaluate the performance of a semantic network to generalize across scenes in this way the model is trained and tested on different scenes.

Curriculum training was adapted to help improve the network generalization, all of the networks being considered were first trained on the existing datasets mentioned in Section II-A, and then further trained on the novel dataset presented in Section III. This serves to investigate the convergence rate and generalizability of different encoder–architecture combinations (Section IV-B), as well as the similarity of hidden representations among fluvial datasets (Section V).

A. Data Preparation

300, 535, and 512 images were chosen from the three existing aerial fluvial datasets (Section II-A) of [31], [33], and [30], respectively, to form a base training dataset of 1347 images. Sample images from each selected subset of the surveillance [31], 11 rivers [33], lantern pole [30], and our AFID dataset are shown in Fig. 2(a)–(d). AFID is binarized to water and nonwater for all neural network training and testing. This is sufficient for branch planning and obstacle avoidance of ASVs in fluvial navigation since water pixels correspond to all navigable regions in physical scenes. To ensure all of the annotated data were applicable to the binary image segmentation task the classes within the 11 rivers [33] were turned into two classes by combining all nonwater classes into a single class. The surveillance [31] and lantern pole [30] datasets already contain binary water–nonwater masks, and thus, no alterations were required.

Our dataset was first split into Wabash River and Wildcat Creek subsets due to their differences, as discussed in Section III. The geographic subsets were then further split into training, validation, and test sets with a 40%–20%–40% split. Each subset's test set was then additionally combined together to form an equally weighted super-set for testing across both fluvial scenes. After the base training was done with existing datasets, all the models were tested on the the Wabash River, Wildcat Creek, and the combination test sets.

After testing the networks ability to transfer to the new data, all the models were further trained on the training set of each subset and their combination, and then tested on the unseen test subset and the combined test set. All steps of training have the same early-stopping criteria based upon F1 Score of validation set of combined Wabash River and Wildcat Creek subsets (Section IV-C). Model training, validation, and testing scheme on the split dataset is shown in Fig. 4.

Our multiclass dataset was converted to binary labels (water and nonwater) to align with the navigation specific challenges this dataset in part tries to solve and the existing fluvial datasets. However, it still stands as a unique fluvial aerial semantic segmentation dataset due to its attention to waterborne obstacles and inclusion of multiple aerial camera view angles.

B. Segmentation Architectures

State-of-the-art semantic segmentation neural networks usually consist of a encoder that captures contextual information, and a subsequent decoder that recovers the spatial information, with skip connection that bypasses at least one layer from the encoding (downsampling) process to decoding (upsampling) process, so as to merge features in various resolution levels to recover the fine-grained spatial and contextual information. Since the decoder often has geometrically symmetric shape and sequentially reverse order with the encoder, different semantic segmentation architectures mainly vary in the decoding process where various methods of connections between the encoder and decoder parts are used. The details of the encoders we used in this work are provided in Section IV-B1, and that of architectures are given in Section IV-B2.

1) Encoders: Convolutional neural network (CNN) encoders aim to summarize or encode image features using kernels into a spatially dense yet feature verbose latent space that stands for a more abstract representation of the input. On the other hand, inspired by the Transformer [65] design in NLP, many vision-targeted works split an image into a sequence of linearly embedded patches, then feed into a standard Transformer to learn a feature representation with stronger effective receptive field (ERF). Encoder learning involves teaching an encoder how to accurately distill an input into a latent space that is most helpful in correctly classifying an image. Different encoders use different layered architectures or attention mechanism to enable either faster or more accurate latent space realizations for inputs. In this article, to evaluate the performance of different encoders on the binary semantic segmentation task, the following five popular encoders are investigated: 1) ResNet [66], 2) Xception [67], 3) MobileNet-V3 Large 100 [68], 4) EfficientNet-B4 [69], and 5) MiT-b1 [70].

ResNet [66] features residual learning which connects the input and output of each layer (or block). This makes the network easier to optimize and decreases the degradation problem for considerably deeper networks while increasing accuracy. In this work, a moderate-sized ResNet encoder, ResNet50, is used. ResNet50 is a moderate-sized ResNet encoder with 16 bottleneck blocks (each contains three convolutional layers), one initial convolutional layer, and one average pooling layer.

Xception network [67], like Inception-v4 [71], merges the idea of GoogLeNet [72] and ResNet [66], but replaces Inception modules with depthwise separable convolutional layers. It has been shown that separable convolutional layers use fewer parameters, less memory, and fewer computations than regular convolutional layers, and have comparable or even better performance than Inception modules [73].

MobileNets [68] are small, low-latency, low-power models designed to effectively maximize accuracy while being mindful of the restricted resources for embedded or mobile applications. MobileNet V3 [68] harnessed multiple network architecture search algorithms, adapted nonlinearities, and applied squeeze and excite [74] in a quantization friendly and efficient manner. In this article, we used the MobileNet V3 large model with a 100% layer width multiplier (no shrinkage) and no resolution multiplier since the model size is already very small compared to other networks shown in Table II.

TABLE II
ENCODER-DECODER MODEL SPECIFICATIONS IN TERMS OF ON-DISK MODEL SIZE IN MEGABYTES, GFLOPS AND PREDICTION SPEED (NVIDIA RTX2060, BATCH SIZE=1, AVERAGED OVER 300 INFERENCES) IN MILLISECONDS. SPECIFICALLY, RESNET50, MOBILENET V3 LARGE 100, EFFICIENTNET-B4 AND MIT-B1 ENCODERS WERE TESTED, DETAILS IN SECTION IV-B1. BEST PROPERTY VALUES ACROSS ALL MODELS ARE EMPHASIZED IN BOLD.

		Unet	**DeeplapV3+**	**PAN**
ResNet	Size (MB)	32.5	26.7	24.3
	GFLOPs	28.35	24.39	23.10
	Inference Speed (ms)	478.04	445.23	402.12
MobileNet	Size (MB)	6.7	4.7	**3.1**
	GFLOPs	8.20	3.13	**1.42**
	Inference Speed (ms)	221.65	191.19	**139.61**
Xception	Size (MB)	28.8	-	-
	GFLOPs	28.05	-	-
	Inference Speed (ms)	577.45	-	-
EfficientNet	Size (MB)	20.2	18.6	17.7
	GFLOPs	12.39	12.04	10.73
	Inference Speed (ms)	835.10	944.51	923.25
MiT	Size (MB)	16.5	-	13.36
	GFLOPs	13.89	-	6.88
	Inference Speed (ms)	396.66	-	272.79

Specifically, ResNet50, MobileNet V3 Large 100, EfficientNet-B4, and MiT-B1 Encoders were tested, details in Section IV-B1.
Best property values across all models are emphasized in bold.

EfficientNet [69] tries to achieve a better balance between model accuracy and efficiency by a compound scaling method, which scales all dimensions of network width, depth, and resolution. In this article, we use EfficientNet-B4, which applies fourth power to the scaling factors along all three dimensions under some constraints. EfficientNet has a moderate model size when compared with other networks, as shown in Table II.

MiT (mix vision transformer) is proposed in the work of Segformer [70] for transformer-based simple and efficient semantic segmentation. As a feature encoder, MiT has a hierarchical feature representation to generate CNN-like multilevel features with larger effective receptive field. In comparison to other ViT [75]-based architectures like SETR [76], MiT encoder is not only light-weight but also can capture both high-resolution coarse and low-resolution fine features.

2) Architectures: Network decoders, or upsampling layers, within an architecture play a significant role in deep semantic segmentation tasks. Unlike image classification where the spatial representation of the output does not matter, in semantic segmentation the classified output features must have the same spatial resolution as the input. This creates two codependent requirements for decoders, one of spatial extraction and one of accurate pixel classification at spatial density. For this work multiple semantic segmentation architectures, whose major contributions lie in the decoding process, were tested and briefly explained, including Unet [59], DeepLabV3+ [61], and PAN [60].

Unet [59] architecture consists of a convolution path to extract context and a symmetric deconvolution path that enables precise localization with skip connections (cropped and copied along channel dimension) between same level downsampling and upsampling layers. With the benefits brought by skip connections, U-Net can capture fine spatial information and bolster segmentation results whilst keeping computation costs low.

DeepLabV3 [77] architecture proposed atrous spatial pyramid pooling (ASPP), which does convolutions with kernels of different strides (dilation rates) in parallel on the encoder-generated feature maps. On top of DeepLabV3, DeepLabV3+ [61] incorporates a single skip connection (channelwise concatenation) between one encoder layer and the ASPP-processed layer, as well as two bilinear upsamplings before and after the skip connection. In this work, we adopted the decoding part from DeepLabV3+ architecture.

Pyramid Attention Network (PAN) [60] exploits the potential of both attention mechanisms and spatial pyramids to extract precise dense features for pixel labeling by introducing feature pyramid attention (FPA) module and global attention upsample (GAU) module. Acting like a postprocessing layer after the final encoder layer, the FPA module performs spatial pyramid attention combined with global pooling to improve feature representation and receptive fields, whilst the GAU module provides globally aware spatial context to latent space features to aid in localization of classified decoder outputs.

Although DeepLabV3+ [61] investigated the incorporation of depthwise separable convolution from Xception [67] with atrous spatial pyramid pooling (ASPP) [77], it is still controversial to combine them into a single semantic architecture since the DeepLabv3+ atrous depthwise convolution modifies the Xception encoder itself. A similar case happens for the combination of the Xception encoder and PAN [60] network architecture. Since dilation mode of DeepLabV3+ decoder is only for convolution layers, the combination of DeepLabV3+ with MiT encoder is also invalid. Thus finally we tested on 12 encoder-architecture combinations, details shown in Table II.

C. Evaluation Metrics

When evaluating the model performance, two metrics are used: (1) mIoU (mean intersection over union) and (2) F1 Score (harmonic mean of precision and recall), which are effective measures of how well network inferences overlap with groundtruth masks. The four terms, true positive (TP), true negative (TN), false positive (FP), and false negative (FN), can each be thought of as a 2-D tensor along imagewise and classwise directions, respectively. As this article investigates binary (water and nonwater) semantic segmentation, all terms collapse to single dimension array along imagewise direction, and both metrics are calculated at image level and averaged over the dataset.

$$\text{mIoU} = \frac{\text{TP}}{\text{TP + FP + FN}} \tag{1}$$

$$\text{F1 Score} = \frac{\text{2TP}}{\text{2TP + FP + FN}} \tag{2}$$

$$\text{soft Dice Loss} = 1 - \frac{2\,\mathbf{y_{pred}}^{T} \cdot \mathbf{y_{mask}}}{|\mathbf{y_{pred}}|_1 + |\mathbf{y_{mask}}|_1}. \tag{3}$$

The objective function, or loss function, for the training of all tested semantic segmentation networks in this article is soft Dice Loss [78], which is approximately 1 − F1 Score. Suppose $\mathbf{y_{pred}}$ and $\mathbf{y_{mask}}$ are flatten 1-D vectors of sigmoid activated prediction probability and mask values (water: 1, nonwater: 0) of water pixels, respectively, the definition of soft Dice Loss is shown in (3), where $|\mathbf{y_{pred}}|_1$ and $|\mathbf{y_{mask}}|_1$ are 1-D norms of vectors.

Properties, such as model size, giga floating-point operations (GFLOPs), and average inference speed, of each implemented network (Table II) are measured using a Github repository [79]). These three parameters illustrate the model memory consumption, mathematical calculations, and time consumption, respectively, for a network forward inference pass. An intuitive sense of speed, memory, and power consumption of different networks is critical to enable informed decisions regarding network deployment on memory and power limited embedded devices for real-world applications, such as ASV-UAV navigational cooperation.

D. Network Implementation

All training, validation, and testing was done on a NVIDIA RTX2060 GPU with 6 GB of VRAM. PyTorch Lightning [80] was used as a training and testing boilerplate to rapidly assemble, change, train, and test networks within a common framework using PyTorch [81]. A Github repository [82] was used to easily assemble the encoder and architecture frameworks. Finally, Wandb [83] was used to log training and evaluation processes, model hyperparameters, and training checkpoints.

Running mean and standard deviations of all batch normalizations [84] were maintained throughout the training processes of each model. Model (specifically encoder) weights are pretrained on ImageNet [85]. All images and masks are resized to 320 × 544 during dataloader transform, in which resolution the obstacles in river are still visible. Besides, the height and width are each a power of 2 since encoders usually have five stages of downsampling by a factor of 2. Batch size of training data was 6 and 16 b native automatic mixed precision (AMP) was used to account for limited graphic memory allocations. The Adam optimizer [86] was used with learning rate 0.0001 for all networks. Soft Dice Loss [78] [(3)] was used as the single loss function across all training due to its nonconvex nature and better performance on imbalanced data [64]. A maximum of 75 epochs was set for each training session, with early stopping implemented for changes in the networks validation F1 score, with a patience of 10 epochs, and minimum delta of 0.

E. Benchmarks

Based on the experimental setup described in Section IV, performance of pretrained and continue-trained models on two subsets and the combined test set are presented in Table III. Noticeably, all examined models achieve good results after continue-trained on the combined test set. The table shows five main noticeable aspects of the novel dataset and the tested networks. First, the Unet-MiT model achieves the best performance on the unseen combined test set after pretraining, and PAN-MiT model has the second best performance. This reveals the higher generalization ability of Transformer-based encoders over CNN-based ones. Second, among all continue-trained models with ResNet or MobileNet encoder, models performance is less competitive to other encoders with the same decoding

TABLE III
MODEL PERFORMANCE BENCHMARK ON TEST SETS OF OVERALL DATASET AND TWO (WABASH RIVER ONLY AND WILDCAT CREEK ONLY) SUBSETS WITH PRETRAINED BASELINES (DENOTED BY **B**, BEST SCORE IS UNDERLINED BOLD) AND CONTINUE-TRAINED (DENOTED BY **C**, BEST SCORE IS BOLD) MODELS

	Overall Dataset				Wabash Subset				Wildcat Subset			
	mIoU %		F1 %		mIoU %		F1 %		mIoU %		F1 %	
	B	C	B	C	B	C	B	C	B	C	B	C
Unet-R	83.0	95.5	91.2	98.5	86.6	89.1	92.7	94.1	80.8	52.2	90.0	61.9
Unet-X	83.0	95.8	90.7	98.6	80.3	92.1	88.4	95.9	84.6	62.6	92.5	72.2
Unet-E	83.9	95.7	91.2	98.6	83.4	93.6	90.5	96.7	84.2	77.1	91.8	85.2
Unet-M	80.6	95.7	89.7	98.4	87.0	90.4	93.2	95.1	76.9	78.8	86.8	88.0
Unet-T	**88.1**	**96.2**	**94.2**	**98.7**	**88.6**	**93.9**	**93.8**	**96.9**	87.7	73.9	94.5	84.8
DLV3P-R	83.6	95.6	91.7	98.4	86.2	89.7	92.6	94.2	82.1	74.1	90.9	84.6
DLV3P-E	78.5	95.5	87.8	98.5	69.0	91.8	80.3	95.9	84.2	**85.3**	92.9	**93.2**
DLV3P-M	80.9	95.1	90.0	98.3	82.3	91.5	90.3	95.7	80.2	78.5	89.8	88.4
PAN-R	81.6	95.4	90.5	98.4	82.1	91.1	90.2	95.4	81.3	56.7	90.8	66.2
PAN-E	84.6	95.7	92.3	98.5	85.8	90.9	92.1	95.4	83.9	79.5	92.4	89.7
PAN-M	80.3	95.1	89.0	98.3	81.4	91.4	88.8	95.7	79.7	73.3	89.1	83.5
PAN-T	87.9	95.9	93.9	98.5	86.0	92.4	91.9	95.9	**89.0**	81.3	**95.5**	90.9

Results on overall dataset illustrate the baseline models were subsequently trained and tested on the whole dataset. Results on Wabash subset means baseline models continue-trained on Wildcat Creek training subset and tested on Wabash River test subset, results on Wildcat Subset is the other way around. **R**, **X**, **E**, **M**, and **T** Represent **R** esNet50, **X** ception, **E** fficientNet-B4, **M** obilenet V3 Large 100, and Mi **T**-B1, respectively. DLV3P is abbreviation of DeepLabV3+ architecture. MiT is abbreviation of mix transformer encoder.

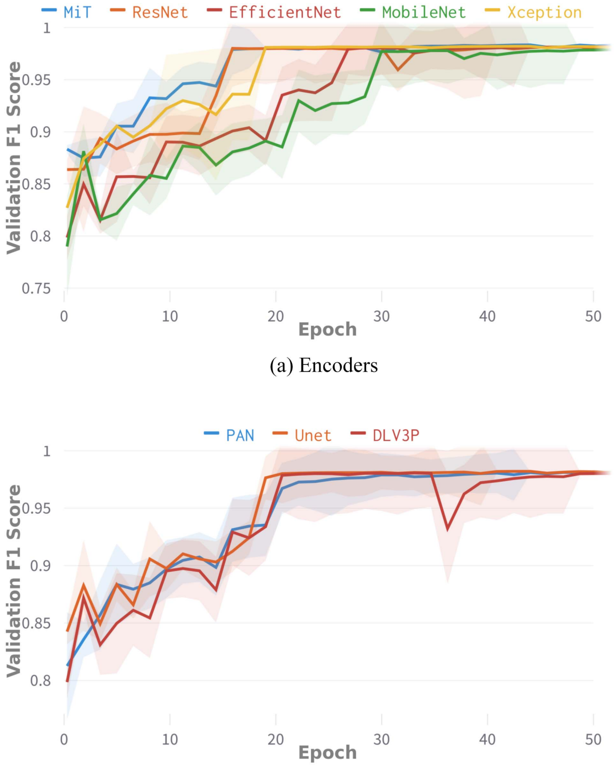
architecture. This further illustrates that semantic segmentation model performance relies more on encoder design than decoder design. Third, nearly all Wabash River continue-trained models exhibit performance drop when tested on Wildcat Creek subset, showing that the Wabash River subset stands in contrast with the Wildcat Creek subset enough to challenge network generalization from the former to the latter.

Fourth, the opposite happens for the Wildcat Creek continue-trained models, showing that the inclusion of obstacle heavy environments is important for network success and does not hinder network performance in less cluttered fluvial environments. Finally, the DeepLabV3+-EfficientNet model has the largest performance boost after continue-training on either subset or overall set, and it is the only model here that improves after subset continue-training, which further validates the power of compound scaling method brought by EfficientNet (Section IV-B1) and ASPP brought by DeepLabV3+ (Section IV-B2).

V. DISCUSSION

Inspired by the concept of curriculum learning [87], where learning the broad concepts on less complex training sets first and then refining the concepts on more complex training sets gives distinct advantages over a network that is trained to grasp all scene complexities at once. The training scheme in this work utilized existing fluvial datasets to provide base training for the networks on context surrounding fluvial scenes before exposing the networks to our novel dataset with more complex and fine-grained waterborne obstacles.

The incorporation of curriculum training is validated by the performance drop (Table III) of models trained beyond the existing base set with Wabash River subset and then tested on the more complex Wildcat Creek subset. The subset is more complex as it involves more water-born obstacles causing it to be significantly different than the pretraining datasets and Wabash River subset, Fig. 2(d) and Fig. 2. Another reason for the continue-training and cross-testing in the experiments is to investigate the generalization ability of different network architectures, which were trained and tested on different fluvial scenes.

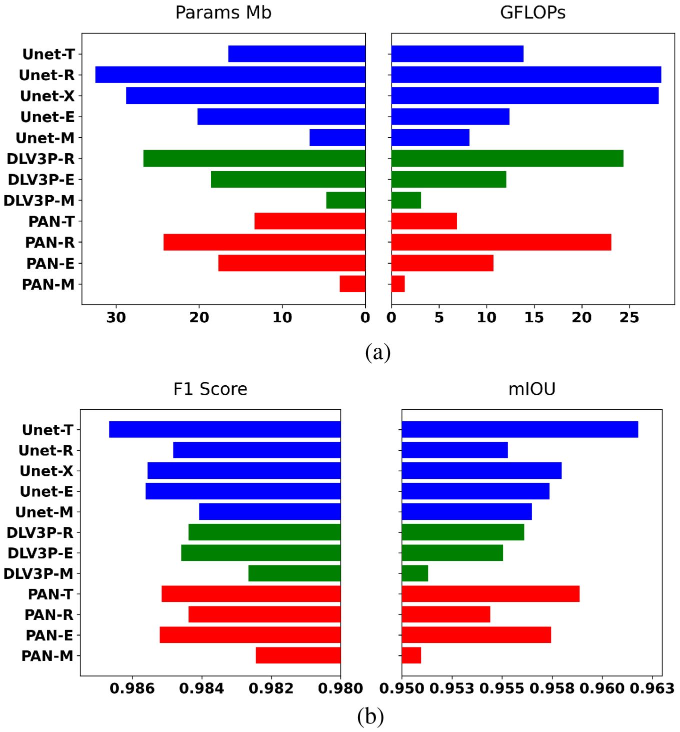


(a) Encoders

(b) Architectures

Fig. 5. (a) Validation set F1 Score of all epochs during pretraining on existing datasets and continue-training on Wabash-Wildcat overall training set in terms of different encoders and (b) different architectures. Solid line represents median, vertical color span represents standard error.

The training process of neural networks exhibits inconspicuous difference in terms of continue-training convergence speed across different decoding architectures. As shown in Fig. 5(b), the sloper of validation F1 Score is basically the same for all



(a)

(b)

Fig. 6. Bar plots of models' overhead and performance comparisons. **R**, **X**, **E**, **M** and **T** represent **R**esNet50, **X**ception, **E**fficientNet-b4, **M**obileNet V3 Large 100 and Mi**T**-b1, respectively. DLV3P is abbreviation of DeepLabV3+ architecture. (a) Model Overhead. (b) Model Performance.

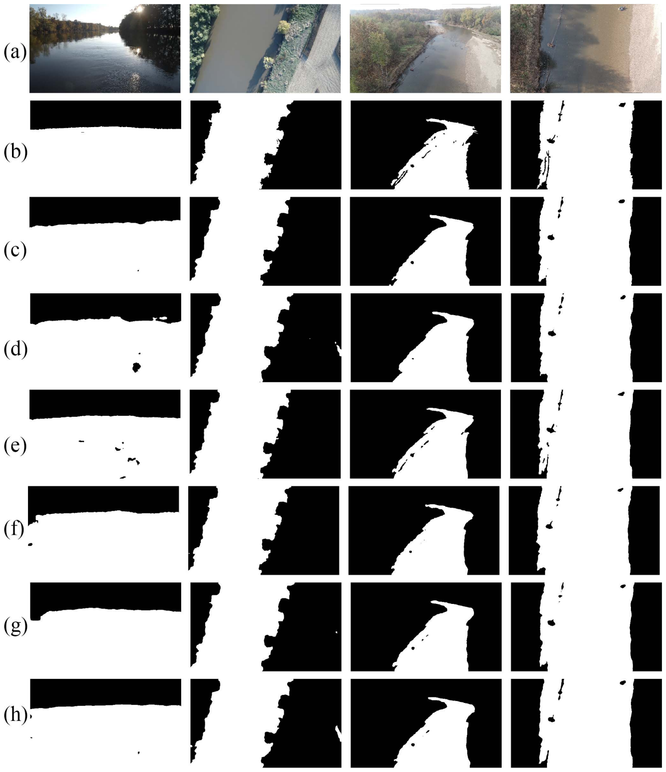


Fig. 7. Qualitative comparison of binary (water and nonwater) predictions of multiple encoder-decoder semantic segmentation networks. Rows a-h denote, respectively. (a) Original RGB images. (b) Ground-truth masks. (c) EfficientNet-Unet predicted masks. (d) MobileNet-Unet predicted masks. (e) MiT-Unet predicted masks. (f) EfficientNet-DeepLabV3+ predicted masks. (g) MiT-PAN predicted masks. (h) EfficientNet-PAN predicted masks.

decoding architectures. One difference is that PAN architecture shows to have more learning stability across all epochs than Unet and DeepLabV3+. However, the training curves of networks show significant difference in validation set F1 Score across different encoders. As shown in Fig. 5(a), the MiT and Xception encoders show a higher learning rate in the pretraining stage, the ResNet encoder shows the largest performance boost when continue-training starts, whereas the EfficientNet and MobileNet encoders have the slowest learning speeds in both pretraining and continue-training stages. Model F1 score starts from about 0.8 since ImageNet [85] pretrained weights were used. Given enough epochs, all models achieve approximately good F1 scores on the validation set.

Fig. 6 provides visual presentation of model overhead and performance. The best-performing model has MiT as encoder and Unet as decoding architecture, as shown in Fig. 6(b). In addition, from Fig. 6(a), this model has nearly half the size and FLOPs compared to those with ResNet50 and Xception encoders. Besides, the Unet-EfficientNet model reaches almost the same F1 Score and mIoU as the Unet-Xception model at only two thirds of its memory consumption and less than half of its FLOPs. The variant of each encoder was chosen to make the encoder-architecture model have approximately the same size (around 25 MB) as the other encoders. The exception to this was the MobileNet encoder, where the biggest variant only occupies around 10 MB.

It can also be verified from Fig. 6(a) that encoder types have more impact on the final model size and matrix operations required for a forward pass than architecture does. It can be observed that models with ResNet50 encoder have the largest model size and FLOPs among all five tested encoders, while MobileNet encoder occupies the least memory. As the most memory-efficient encoder, MobileNet V3 Large 100 consumes almost one third of the memory of EfficientNet-b4, with at most 0.6% mIoU and 0.2% F1 Score performance drop. This makes it also a promising encoder to balance the computational and power requirements of network inferences with the models ability to properly segment fluvial scenes.

Furthermore, it can be observed from Fig. 6(b) that mIOU and F1 Score are positively correlated for all tested models. From (1) and (2), it is known that F1 Score measures the average performance of a network on a dataset, whereas mIOU measures the worst-case performance. Thus the positive correlation of these two metrics reveals the fairness to use either metric as the early stopping criterion together with measuring the model convergence rate during the training process (Fig. 5). Similarly, the same correlation exists between model size and floating-point operations in a forward pass, which can be observed from Fig. 6(a).

The analysis demonstrated that the performance differences among all tested models are inconspicuous [Table III and Fig. 6(b)], however, their overhead differences are quite large

[Table II and Fig. 6(a)]. Fig. 6 shows that models with Unet architecture have an overall higher performance with a higher variance of model overhead. Moreover, models with EfficientNet encoder have both moderate performance and overhead, while models with MobileNet encoder have the worst performance but the least overhead. To better visualize the performance difference among models while considering the accuracy-overhead tradeoff, six promising models were selected for further qualitative comparison: Unet architecture with MiT-b1, Efficient-b4 and MobileNetV3Large100 as encoders, DeepLabV3+ architecture with Efficient-b4 as encoder, PAN architecture with MiT-b1 and EfficientNet-b4 as encoders.

A qualitative comparison of the inferenced binary masks of four input images (two from Wabash River, two from Wildcat Creek; two slanted view, two nadir view) from six chosen models is presented in Fig. 7. Row a and row b are original RGB images and ground-truth masks, respectively. Rows c through e are the predictions of the Unet architecture with three best encoders. Row f is the prediction of DeepLabV3+ architecture with EfficientNet-b4 encoder. Row g and row h are results from the PAN-MiT-b1 network and PAN-EfficientNet-b4 network.

For fine-grained waterborne obstacles, Unet-EfficientNet (row c) and PAN-MiT (row g) suffer the least from false negatives, whereas other models falsely recognize more water pixels as nonwater, as shown in the first column. From column three and column four, it can be seen that the Unet-MiT (row f) model has the highest resolution predictions of waterborne obstacles, but also suffers from false positives shown in column 1. In addition, MobileNet-based model (row d) struggle handling water reflections and shadows and have larger numbers of false negatives (column 1) and false positives (column 4). Conclusively, the Unet-MiT model has the best prediction accuracy according to Table III. Unet-MiT and PAN-MiT models have the best accuracy-efficiency tradeoff according to Figs. 6 and 7.

VI. Conclusion

ASV navigation along rivers and creeks with branch planning is challenged by localized flatten view from the surface level. Currently, it is still difficult for ASVs to make long-range navigation plans with river branch selection in GPS-denied areas, and to swiftly and safely do obstacle avoidance while navigating in narrow, rapidly flowing, and obstacle-intensive inland waterways. This hinders the applications using ASV, like autonomous water quality/level monitoring, land cover change analysis, and disaster alerts. UAVs can provide images from various elevations, camera perspectives, and radial distances, with fluvial context beyond that obtainable by an ASV. This article presents the novel semantic-segmented multiclass AFID with 816 images. This dataset is featured with both slanted and nadir perspectives of drone images taken on top of river and creek, with carefully and finely labeled eight classes, especially waterborne obstacle pixels, for autonomous fluvial navigation of both drones and surface vehicles in inland waterways. Multiple deep semantic segmentation encoder–decoder neural networks were pretrained on the existing fluvial datasets, and continued trained on our dataset to build the binary (water and nonwater) baseline performance of models in terms of F1 Score and mIOU. Comparison among models was conducted, and analyzed both quantitatively and qualitatively, with special attention in accuracy-efficiency tradeoff. The semantic segmentation models Unet-MiT and PAN-MiT stand out from the 12 tested models with the most potential in water segmentation and embedded device deployment.

In the future, this dataset will be used to train the light-weight multiclass semantic segmentation models to be deployed directly on UAV and/or ASV to gain more awareness of water and obstacles regions during autonomous enroute navigation. Besides, this AFID dataset has been made publicly available [62] for researchers both in remote sensing field and robotic automation field to train their deep semantic segmentation neural networks on for various tasks.

Acknowledgment

The authors would like to thank Li-Fan Wu for assistance in data collection, Reeve Lambert for meticulous and rigorous reviewing and editing, Ajinkya Chaudhary and Arijeet Nath for image annotation, and Jalil Chavez Galaviz for reflective discussion.

References

[1] R. Lambert, B. Page, J. Chavez, and N. Mahmoudian, “A low-cost autonomous surface vehicle for multi-vehicle operations,” in *Proc. Glob. Oceans: Singap.–US Gulf Coast*, 2020, pp. 1–5.

[2] R. Lambert, J. Li, L.-F. Wu, and N. Mahmoudian, “Robust ASV navigation through ground to water cross-domain deep reinforcement learning,” *Front. Robot. AI*, 2021, Art. no. 289.

[3] J. Choi, J. Park, J. Jung, Y. Lee, and H.-T. Choi, “Development of an autonomous surface vehicle and performance evaluation of autonomous navigation technologies,” *Int. J. Control, Automat. Syst.*, vol. 18, no. 3, pp. 535–545, 2020.

[4] C.-M. Tsai, Y.-H. Lai, J.-W. Perng, I.-F. Tsui, and Y.-J. Chung, “Design and application of an autonomous surface vehicle with an AI-based sensing capability,” in *Proc. IEEE Underwater Technol.*, 2019, pp. 1–4.

[5] M. Dunbabin, A. Grinham, and J. Udy, “An autonomous surface vehicle for water quality monitoring,” in *Proc. Australas. Conf. Robot. Automat.*, 2009, pp. 2–4.

[6] L. Steccanella, D. D. Bloisi, A. Castellini, and A. Farinelli, “Waterline and obstacle detection in images from low-cost autonomous boats for environmental monitoring,” *Robot. Auton. Syst.*, vol. 124, 2020, Art. no. 103346.

[7] J. Moulton, N. Karapetyan, M. Kalaitzakis, A. Quattrini Li, N. Vitzilaios, and I. Rekleitis, “Dynamic autonomous surface vehicle controls under changing environmental forces,” in *Field and Service Robotics*, G. Ishigami and K. Yoshida, Eds. Singapore: Springer Singapore, 2021, pp. 381–394.

[8] N. Karapetyan, J. Moulton, and I. Rekleitis, “Meander-based river coverage by an autonomous surface vehicle,” in *Field and Service Robotics*, G. Ishigami and K. Yoshida, Eds. Singapore: Springer Singapore, 2021, pp. 353–364.

[9] F. Peralta, M. Arzamendia, D. Gregor, D. G. Reina, and S. Toral, “A comparison of local path planning techniques of autonomous surface vehicles for monitoring applications: The Ypacarai lake case-study,” *Sensors*, vol. 20, no. 5, 2020, Art. no. 1488.

[10] D. H. Stolfi, M. R. Brust, G. Danoy, and P. Bouvry, “UAV-UGV-UMV multi-swarms for cooperative surveillance,” *Front. Robot. AI*, vol. 8, 2021, Art. no. 616950.

[11] H. G. Tanner, “Switched UAV-UGV cooperation scheme for target detection,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2007, pp. 3457–3462.

[12] S. Hood, K. Benson, P. Hamod, D. Madison, J. M. O’Kane, and I. Rekleitis, “Bird’s eye view: Cooperative exploration by UGV and UAV,” in *Proc. Int. Conf. Unmanned Aircr. Syst.*, 2017, pp. 247–255.

[13] Z. Kashino, G. Nejat, and B. Benhabib, "Aerial wilderness search and rescue with ground support," *J. Intell. Robotic Syst.*, vol. 99, no. 1, pp. 147–163, Jul. 2020.

[14] L. Cantelli, P. Laudani, C. Melita, and G. Muscato, "UAV/UGV cooperation to improve navigation capabilities of a mobile robot in unstructured environments," in *Advances in Cooperative Robotics*. Singapore: World Scientific, 2017, pp. 217–224.

[15] L. Cantelli, M. L. Presti, M. Mangiameli, C. Melita, and G. Muscato, "Autonomous cooperation between UAV and UGV to improve navigation and environmental monitoring in rough environments," in *Proc. 10th Int. Symp. Humanitarian Demining Coupled 11th IARP WS HUDEM*, 2013, vol. 23, pp. 109–112.

[16] P. Kim, L. C. Price, J. Park, and Y. K. Cho, "UAV-UGV cooperative 3D environmental mapping," in *Proc. ASCE Int. Conf. Comput. Civil Eng.*, 2019.

[17] F. Guérin et al., "UAV-UGV cooperation for objects transportation in an industrial area," in *Proc. IEEE Int. Conf. Ind. Technol.*, 2015, pp. 547–552.

[18] R. R. Murphy, E. Steimle, C. Griffin, C. Cullins, M. Hall, and K. Pratt, "Cooperative use of unmanned sea surface and micro aerial vehicles at hurricane Wilma," *J. Field Robot.*, vol. 25, no. 3, pp. 164–180, 2008.

[19] E. Vasilopoulos, G. Vosinakis, M. Krommyda, L. Karagiannidis, E. Ouzounoglou, and A. Amditis, "A comparative study of autonomous object detection algorithms in the maritime environment using a UAV platform," *Computation*, vol. 10, no. 3, 2022, Art. no. 42.

[20] X. Xiao, J. Dufek, T. Woodbury, and R. Murphy, "UAV assisted USV visual navigation for marine mass casualty incident response," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 6105–6110.

[21] M. F. Ozkan, L. R. G. Carrillo, and S. A. King, "Rescue boat path planning in flooded urban environments," in *Proc. IEEE Int. Symp. Meas. Control Robot.*, 2019, pp. B2-2-1–B2-2-9.

[22] E. Pinto, P. Santana, F. Marques, R. Mendonça, A. Lourenço, and J. Barata, "On the design of a robotic system composed of an unmanned surface vehicle and a piggybacked VTOL," in *Proc. Doctoral Conf. Comput., Elect. Ind. Syst.*, 2014, pp. 193–200.

[23] N. O'Mahony et al., "Deep learning vs. traditional computer vision," in *Proc. Sci. Inf. Conf.*, 2019, pp. 128–144.

[24] L. P. Osco et al., "A review on deep learning in UAV remote sensing," *Int. J. Appl. Earth Observ. Geoinformation*, vol. 102, 2021, Art. no. 102456.

[25] S. M. Azimi, P. Fischer, M. Körner, and P. Reinartz, "Aerial LaneNet: Lane-marking semantic segmentation in aerial imagery using wavelet-enhanced cost-sensitive symmetric fully convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 5, pp. 2920–2938, May 2019.

[26] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Can semantic labeling methods generalize to any city? The Inria aerial image labeling benchmark," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2017, pp. 3226–3229.

[27] F. Rottensteiner, G. Sohn, M. Gerke, J. D. Wegner, U. Breitkopf, and J. Jung, "Results of the ISPRS benchmark on urban object detection and 3D building reconstruction," *ISPRS J. Photogrammetry Remote Sens.*, vol. 93, pp. 256–271, 2014.

[28] W. Boonpook, Y. Tan, and B. Xu, "Deep learning-based multi-feature semantic segmentation in building extraction from images of UAV photogrammetry," *Int. J. Remote Sens.*, vol. 42, no. 1, pp. 1–19, 2021.

[29] Y. Lyu, G. Vosselman, G.-S. Xia, A. Yilmaz, and M. Y. Yang, "UAVid: A semantic segmentation dataset for UAV imagery," *ISPRS J. Photogrammetry Remote Sens.*, vol. 165, pp. 108–119, 2020.

[30] A. Eltner, P. O. Bressan, T. Akiyama, W. N. Gonçalves, and J. Marcato Junior, "Using deep learning for automatic water stage measurements," *Water Resour. Res.*, vol. 57, no. 3, 2021, Art. no. e2020WR027608.

[31] L. Lopez-Fuentes, C. Rossi, and H. Skinnemoen, "River segmentation for flood monitoring," in *Proc. IEEE Int. Conf. Big Data*, 2017, pp. 3746–3749.

[32] M. Zaffaroni and C. Rossi, "Water segmentation with deep learning models for flood detection and monitoring," in *Proc. Conf. Inf. Syst. Crisis Response Manage.*, 2020, pp. 24–27.

[33] P. E. Carbonneau et al., "Adopting deep learning methods for airborne RGB fluvial scene classification," *Remote Sens. Environ.*, vol. 251, 2020, Art. no. 112107.

[34] X. Zhang et al., "IceNet: A semantic segmentation deep network for river ice by fusing positional and channel-wise attentive features," *Remote Sens.*, vol. 12, no. 2, 2020, Art. no. 221.

[35] C. Sazara, M. Cetin, and K. M. Iftekharuddin, "Detecting floodwater on roadways from image data with handcrafted features and deep transfer learning," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2019, pp. 804–809.

[36] S. Sarp, M. Kuzlu, M. Cetin, C. Sazara, and O. Guler, "Detecting floodwater on roadways from image data using Mask-R-CNN," in *Proc. Int. Conf. Innovations Intell. Syst. Appl.*, 2020, pp. 1–6.

[37] S. M. H. Erfani, Z. Wu, X. Wu, S. Wang, and E. Goharian, "ATLANTIS: A benchmark for semantic segmentation of waterbody images," *Environ. Modelling Softw.*, vol. 149, 2022, Art. no. 105333.

[38] M. Xia, J. Qian, X. Zhang, J. Liu, and Y. Xu, "River segmentation based on separable attention residual network," *J. Appl. Remote Sens.*, vol. 14, no. 3, 2019, Art. no. 032602.

[39] B. Bovcon, J. Muhovič, J. Perš, and M. Kristan, "The mastr1325 dataset for training deep USV obstacle detection models," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 3431–3438.

[40] B. Bovcon et al., "Stereo obstacle detection for unmanned surface vehicles by IMU-assisted semantic segmentation," *Robot. Auton. Syst.*, vol. 104, pp. 1–13, 2018.

[41] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek, "Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 8, pp. 1993–2016, Aug. 2017.

[42] J. Liu, H. Li, J. Luo, S. Xie, and Y. Sun, "Efficient obstacle detection based on prior estimation network and spatially constrained mixture model for unmanned surface vehicles," *J. Field Robot.*, vol. 38, no. 2, pp. 212–228, 2021.

[43] R. Lambert, J. Chavez-Galaviz, J. Li, and N. Mahmoudian, "ROSEBUD: A deep fluvial segmentation dataset for monocular vision-based river navigation and obstacle avoidance," *Sensors*, vol. 22, no. 13, 2022, Art. no. 4681.

[44] S. Achar, B. Sankaran, S. Nuske, S. Scherer, and S. Singh, "Self-supervised segmentation of river scenes," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 6227–6232.

[45] J. Taipalmaa, N. Passalis, H. Zhang, M. Gabbouj, and J. Raitoharju, "High-resolution water segmentation for autonomous unmanned surface vehicles: A novel dataset and evaluation," in *Proc. IEEE 29th Int. Workshop Mach. Learn. Signal Process.*, 2019, pp. 1–6.

[46] C. Mostege, M. Maurer, N. Heran, J. Puerta, and F. Fraundorfer, "Semantic drone dataset," 2019. [Online]. Available: http://dronedataset.icg.tugraz.at

[47] J. R. Ballesteros, G. Sanchez-Torres, and J. W. Branch-Bedoya, "HAG-DAVS: Height-augmented geo-located dataset for detection and semantic segmentation of vehicles in drone aerial orthomosaics," *Data*, vol. 7, no. 4, 2022, Art. no. 50.

[48] A. Marcu, D. Costea, V. Licaret, and M. Leordeanu, "Towards automatic annotation for semantic segmentation in drone videos," 2019. [Online]. Available: https://scholar.google.com/scholar_lookup?arxiv_id=1910.10026

[49] S. Speth et al., "Deep learning with RGB and thermal images onboard a drone for monitoring operations," *J. Field Robot.*, vol. 39, pp. 840–868, 2022.

[50] R. Ribeiro, G. Cruz, J. Matos, and A. Bernardino, "A data set for airborne maritime surveillance environments," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 9, pp. 2720–2732, Sep. 2017.

[51] L. A. Varga, B. Kiefer, M. Messmer, and A. Zell, "Seadronessee: A maritime benchmark for detecting humans in open water," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2022, pp. 2260–2270.

[52] L. Lopez-Fuentes and C. Rossi, "River segmentation dataset," Oct. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1003085

[53] P. E. Carbonneau and J. T. Dietrich, "CNN-supervised-classification," Jul. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3928808

[54] A. Eltner, P. O. Bressan, W. N. Gonçalves, T. Akiyama, and J. M. Junior, "Using deep learning for automatic water level measurements," 2020. [Online]. Available: https://doi.org/10.7910/DVN/ONOZRW

[55] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[56] T. K. Ho, "Random decision forests," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, 1995, pp. 278–282.

[57] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[58] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[59] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervention*, 2015, pp. 234–241.

[60] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," 2018, *arXiv:1805.10180*.

[61] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 801–818.

[62] Z. Wang, L.-F. Wu, and N. Mahmoudian, “Aerial fluvial image dataset (AFID) for semantic segmentation,” Jul. 2022. [Online]. Available: https://purr.purdue.edu/publications/4105/1

[63] A. Bréhéret, “Pixel annotation tool,” 2017. https://github.com/abreheret/PixelAnnotationTool

[64] S. Jadon, “A survey of loss functions for semantic segmentation,” in *Proc. IEEE Conf. Comput. Intell. Bioinf. Comput. Biol.*, 2020, pp. 1–7.

[65] A. Vaswani et al., “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[66] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[67] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1251–1258.

[68] A. Howard et al., “Searching for MobileNetv3,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1314–1324.

[69] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[70] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, “Segformer: Simple and efficient design for semantic segmentation with transformers,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 12077–12090.

[71] C. Szegedy, S. Ioffe, V. Vanhoucke, and A.A Alemi, “Inception-v4, inception-ResNet and the impact of residual connections on learning,” in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4278–4284.

[72] C. Szegedy et al., “Going deeper with convolutions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[73] A. Géron, *Hands-On Machine Learning With Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. Sebastopol, CA, USA: O’Reilly Media, Inc., 2019.

[74] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.

[75] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” 2020, *arXiv:2010.11929*.

[76] S. Zheng et al., “Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6881–6890.

[77] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” 2017, *arXiv:1706.05587*.

[78] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, “Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations,” in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Berlin, Germany: Springer, 2017, pp. 240–248.

[79] “fvcore library,” GitHub repository, GitHub, https://github.com/facebookresearch/fvcore/

[80] F. Williamand the PyTorch Lightning team, “PyTorch Lightning,” 2019. [Online]. Available: https://github.com/PyTorchLightning/pytorch-lightning

[81] A. Paszke et al., “Pytorch: An imperative style, high-performance deep learning library,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.

[82] P. Yakubovskiy, “Segmentation models Pytorch,” 2020. [Online]. Available: https://github.com/qubvel/segmentation_models.pytorch

[83] L. Biewald, “Experiment tracking with weights and biases,” 2020, software available from wandb.com. [Online]. Available: https://www.wandb.com/

[84] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” 2015, *arXiv:1502.03167*.

[85] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[86] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, *arXiv:1412.6980*.

[87] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 41–48.