Deep Learning-Based Spatial Detection of Drainage Structures using Advanced Object Detection Methods

1st Shayan Jalalipour
Computer Science
Portland State University
Portland OR, United States
shayan2@pdx.edu

2nd Sriharshitha Ayyalasomayjula
Computer Science
Portland State University
Portland OR, United States
ayyalas@pdx.edu

3rd Hashem Damrah
Computer Science
Portland Community College
Portland OR, United States
hashem.damrah@pcc.edu

4th Junfan Lin
Computer Science
Portland State University
Portland OR, United States
junfan@pdx.edu

5th Banafsheh Rekabdar
Computer Science
Portland State University
Portland OR, United States
rekabdar@pdx.edu

6th Ruopu Li
Geography and Environmental Resources
Southern Illinois University
Carbondale, Illinois
ruopu.li@siu.edu

***Abstract*—Hydrologic connectivity plays a critical role in understanding and managing environmental processes. The spatial characterization of hydrologic connectivity often relies on hydro-topographic delineation using Geographic Information Systems (GIS) and digital terrain models (DEMs). Recent advancements in LiDAR technology have provided high-resolution DEMs that accurately represent topographic conditions. However, accurately delineating hydrologic connectivity using LiDAR DEMs faces challenges, particularly in the presence of virtual flow barriers such as roads and bridges which impede water flow and act as "digital dams." This paper addresses the need for an efficient and effective approach to detect the locations of drainage structures, such as roads and bridges, which significantly impact hydrologic connectivity. While previous studies have shown that incorporating drainage structures improves the delineation of drainage flows, the availability of consistent and high-quality drainage structure datasets remains limited. Therefore, this study aims to develop a methodology that utilizes deep learning (DL) frameworks to detect drainage structures by leveraging their unique topographic patterns on LiDAR DEMs and supplemental GIS datasets. The paper explores multiple advanced deep learning-based object detection models, including Faster RCNN, DINO, DETR:DINO and YOLOv5, to analyze the distinctive patterns exhibited by drainage structures. These models are trained to spatially detect the locations of drainage structures by recognizing the specific "signatures" present in their topographic patterns. The investigation of these state-of-the-art DL frameworks for drainage structure detection represents a novel approach that extends the current understanding of utilizing DL techniques in the field of hydrologic connectivity analysis. We performed both quantitative and qualitative analyses, and propose a novel evaluation framework to demonstrate that DINO:DETR and Faster-RCNN methods are both capable of correctly identifying culvert locations and outperform DETR and YOLOv5 methods.**

***Index Terms*—Faster RCNN, DINO, DINO:DETR, YOLOv5, LiDAR Elevation Data, DEMs, GIS, Advanced DL methods**

This material is based upon work supported by the National Science Foundation under Grant 1951741.

I. Introduction

Hydrologic connectivity in a landscape is critical to understand a range of environmental management issues, such as tracking the nutrient transport for diffuse pollution runoff [1] [2]. The spatial characterization of hydrologic connectivity often involves hydro-topographic delineation using Geographic Information Systems (GIS) and digital terrain models (DEMs). In recent decades, high-resolution DEMs products, mostly generated from Light Detection and Ranging (LiDAR) technology, have demonstrated their excellent performance in representing topographic conditions in subtle spatial details compared with conventional DEMs [3]. According to the U.S. Geological Survey's 3D Elevation Program, it is expected to have "the first-ever national baseline of consistent high-resolution" LiDAR elevation data by 2023 [4]. However, accurate delineation of hydrologic connectivity using these LiDAR DEMs still exhibits a unique challenge. Many studies have shown that drainage flowlines derived from LiDAR DEMs is susceptible to virtual flow barriers (e.g., roads and bridges), which function as 'digital dams' [5] [6] [7]. This is especially the case in agricultural areas, where the gentle terrain is widely segmented by rural road networks. Previous studies have shown that the incorporation of drainage structures such as culverts and bridges can improve the delineation of drainage flows over the landscape [6] [8] [9]. However, such a drainage structure dataset is largely unavailable, or merely available with inconsistent format and quality. Although the dataset may be developed based on transportation infrastructure records maintained by state transportation agencies, or by on-screen digitization, these methods suffer from the problems of data completeness, accuracy, and/or labor intensity. Therefore, it is imperative to develop an approach that can efficiently and

979-8-3503-5801-8/23/$31.00 ©2023 IEEE
DOI 10.1109/TransAI60598.2023.00007

effectively detect the locations of drainage structures.

The objective of this study is to develop deep learning (DL) models capable of spatially detecting the locations of drainage structures using spatial images that represent topographic and landscape characteristics. In this regard, this paper makes a significant contribution by exploring multiple advanced DL-based object detection models, including Faster RCNN [10], DETR [11], DINO:DETR [12] and YOLOv5 [13]. These models are employed to analyze the unique topographic patterns exhibited by drainage structures on LiDAR DEMs dataset. By leveraging the distinctive 'signatures' present in these patterns, the proposed models aim to efficiently and effectively detect the locations of drainage structures. The investigation of state-of-the-art DL frameworks for drainage structure detection represents a novel approach that expands our current understanding of using DL techniques in hydrologic connectivity analysis. Additionally, it introduces a new framework of metrics to enhance our comprehension of model performance in this specific use case.

The paper is organized as follows: Section II will provide a review of related works. Section III describes the study area and dataset. Section IV provides a background on DL-based object detection methods. In Section V-A and B, our proposed models are presented along with a novel evaluation framework. Section V-C covers the experimental design, evaluation, and results. Finally, in Section VI, we conclude the paper.

II. Related Work

Research has indicated that drainage structures have exhibited unique topographic patterns on the LiDAR DEMs and other supplemental GIS datasets such as aerial orthophotos [14] [15]. These patterns can be leveraged as unique 'signatures' to guide the detection of drainage structures through DL frameworks. DL has shown excellent performance comparable to human performance for solving a wide range of geographic object classification and detection. For example, Ye et al. detected landslides from images using Deep Belief Network (DBN) [16]. For terrain feature mapping, Li and Hsu detected a large collection of terrain features from remotely sensed images using a Faster-RCNN approach [17]. Xu et al. adopted an attention U-Net framework and LiDAR DEMs for developing detailed streamline detection, resulting in an improved performance over traditional machine learning methods [18]. The flow barriers from road embankments were recognized as an important issue to be addressed for future research [19]. Our previous work proved convolutional neural network (CNN) models can achieve high classification accuracy and good model transferability for identifying images containing drainage structure locations [14] [20]. However, it is unclear if a deep learning framework can be extended for detecting the locations of objects with bounding boxes. In this paper, we explore the application of advanced DL-based object detection algorithms, including Faster RCNN, DINO, DETR:DINO, and YOLOv5, for the spatial detection of drainage structures.

TABLE I
Data sources of LIDAR-derived HRDEMs and aerial orthophotos

Data	Locations	Sources	Spatial Resolution	Vertical Accuracy RMSE	Number of Samples
DEM	West Fork Big Blue watershed, Nebraska	Nebraska Department of Natural Resource	1.0m	0.185m	2022
	Vermilion River Watershed, Illinois	Illinois Geospatial Data Clearinghouse	0.30m	0.36m	1011
	Maple River Watershed, North Dakota	North Dakota GIS Hub Data Portal	0.61m	0.15m	613
	Sacramento-Stone Corral Watershed, California	USGS	1.0m	0.196m	2388
Aerial Orthophotos	Four Watersheds	USGS National Agriculture Imagery Program (NAIP)	1.0m	-	6034

III. Dataset

Our data was collected in four study areas - West Fork Big Blue Watershed in Nebraska, Vermilion River Watershed in Illinois, Maple River Watershed in North Dakota, and Sacramento-Stone Corral Watershed in California (Fig. 1). The landscapes of these areas are dominated by relatively flat terrains and dense road networks. We selected these four watersheds because they represent major agricultural systems in the United States. We downloaded LiDAR DEMs and 4-band aerial orthophotos from federal and state agencies. The sources and specifications of these datasets are shown in Table I.

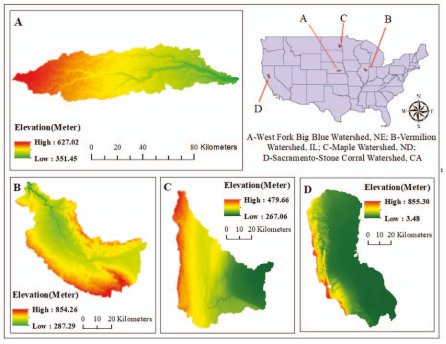


Fig. 1. Topography and locations of four study areas [15]

The data are formatted as 32-bit TIF files, each records terrain elevation data of 800x800 segments of a larger region. The entire dataset is comprised of 6k samples (whose geographical distribution is outlined in Table I) of various topographical segments containing either one or more culverts. Example images can be seen in Fig. 6.

IV. General Approach

Many traditional object detection models face challenges when it comes to accurately detecting drainage structures in LiDAR DEMs. This is primarily because drainage structures have unique topographic patterns that may differ significantly

from other objects or features in the landscape. Traditional object detection models often rely on predefined patterns or templates that may not adequately capture the distinct characteristics of drainage structures. As a result, these models may struggle to accurately identify and delineate drainage structures from LiDAR DEM data. Although prior research has demonstrated the benefits of incorporating drainage structures in drainage flow delineation, there is a scarcity of consistent and high-quality drainage structure datasets. To address this gap, our study aims to develop a methodology that employs DL frameworks to detect drainage structures by leveraging their unique topographic patterns on LiDAR DEMs and supplemental GIS datasets.

In this section, we will assess the effectiveness of three prominent DL based object detection architectures in solving our drainage structures detection (culvert/bridge detection) problem. A key to adapting many object detection models to our use case is framing culvert detection as both object detection (locating an object or bounding box label) as well as classification (identifying the detected object's classification). While our task is not inherently a classification problem, as we know all "objects" of interest are culverts, it is required to utilize existing models whose architectures are framed around both locating a bounding box as well as classifying the object within. To solve this problem, we represent our task as both culvert detection as well as binary classification. Our object classes were defined as "Culvert/Bridge" and "Background". In other words, each object could be classified as either a culvert, or not. With this modification, our task adapts to the detection-and-classification paradigm of many object detection models allowing for easier adaptation for our use case.

A. Faster-RCNN

Faster-RCNN [10] utilizes two model components, a Region Proposal Network (RPN) and an object classifier. The RPN is implemented as a convolutional network whose output is a series of proposed regions (boxes potentially bounding the location of a culvert) as well as 2 scores (for each proposal) representing the probability of being a culvert or not. The proposed regions then feed forward into the second model component, a classifier which shares the convolutional layers' learned features (which are referred to as a feature map). The classifier's output then gives a classification to all regions of interest. Fig. 2 outlines this process, we can see the convolutional network takes input and learns a feature map. These features are given to the RPN, whose output feeds to the classifier along with the feature maps for analysis. Additionally, pre-trained resnet-50 [21] was used as the backend convolutional neural network for this model.

Potential advantages of Faster-RCNN include its reduced parameter size which helps reduce overfitting on our limited dataset size, as well as its competitive run time speed for real time image analysis [10] (which benefits any further users of our model).

B. DETR

Detection Transformer (DETR) is an end to end object detection model consisting of a 'Backbone' convolutional neural network as well as a transformer (encoder-decoder) model. As seen in Fig. 3 the backbone CNN model creates an activation map of features from the input image to be supplied to the transformer for further use. The transformer's encoder learns positional embeddings called "Object (culvert) Queries". The decoder takes object (culvert) queries as input and outputs into a feed forward neural network directly converting transformer output to predicted bounding boxes. Then some matching algorithm (such as bipartite matching) is used to relate the predictions to ground truth labels. We used a Resnet50 [21] as the backbone CNN for the DETR model.

C. DINO: DETR

DINO [12] represents an advanced end-to-end object detection system that surpasses DETR-like models in terms of both efficacy and efficiency. The model incorporates DAB-DETR [22], DN-DETR [23], and Deformable DETR [24] strategies, including a contrastive denoising training methodology, a mixed query selection technique for anchor initialization, and a look forward twice scheme for box prediction.

Similar to DETR [11], the DINO [12] architecture consists of a backbone, a multi-layer transformer encompassing an encoder and decoder, and multiple prediction heads. Fig.4 shows the process of culvert detection using DINO model. Resnet50 serves as the backbone to extract multi-scale features from input images. These extracted features, along with positional embeddings, are then fed into the encoder for enhanced feature representation. To initialize the anchors and positional queries for the decoder, a mixed query selection approach is employed while retaining the learnable aspect of content queries. By utilizing deformable attention, the encoder outputs are combined with initialized anchors and learnable content queries, resulting in feature fusion and iterative query refinement. The final outputs are generated by refining anchor boxes and predicting classification results based on refined content features.

To enhance training efficiency and detection performance, this model formulates positional queries as dynamic anchor boxes and is trained with an additional DN loss. A query selection process generates class embeddings, which are used to predict the object class for each detection. Class embeddings are learned representations of each class (or category) in the dataset.

D. YOLO and YOLOv5

"You Only Look Once" or YOLO for short, is an object detection model which reframes object detection as a single regression problem, simplifying the detection pipeline while maintaining accuracy [25]. By combining all components into a single neural network, YOLO predicts bounding boxes for multiple object classes simultaneously, enabling it to detect a wide range of objects within a single pass.

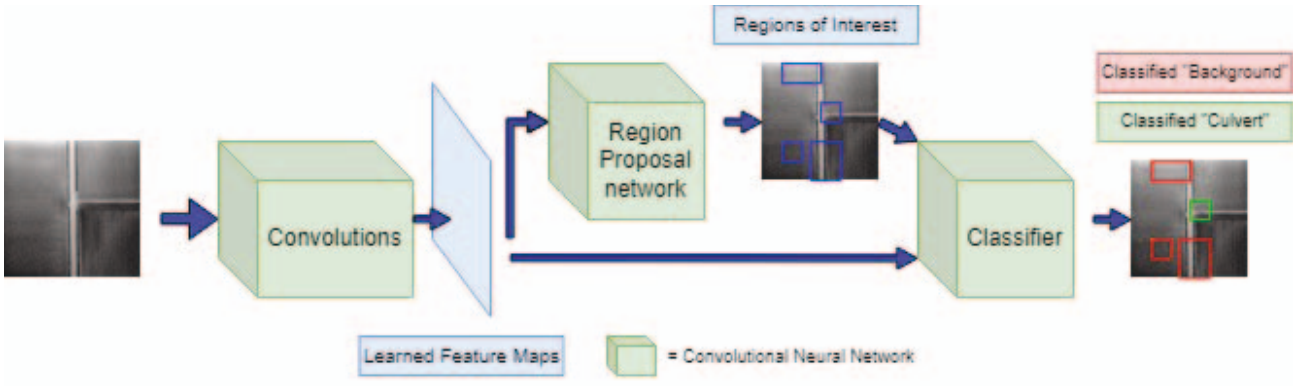


Fig. 2. The Flowchart outline of the Faster-RCNN model applied to our use case of identifying culverts.

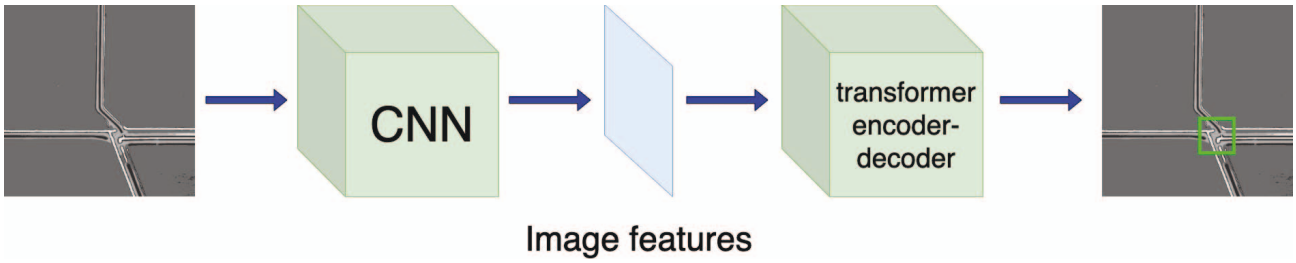


Fig. 3. Example usage of the DETR model and architecture in culvert analysis.

YOLO's primary methodology involves dividing an image into an NxN grid, where each cell in the grid is responsible for determining if the center of an object falls within its bounds, predicting a set number of potential bounding boxes, as well as predicting class probabilities for objects that may or may not fall in the grid cell. An example of this process can be visualized in Fig. 5 This model's greatest advantage is its highly competitive analysis speed.

YOLOv5 is the fifth iteration improvement on the original YOLO algorithm, and the version used in our experiments. While retaining the same fundamental algorithm, it contains several hardware acceleration improvements (for faster training times) as well as architectural improvements such as the addition of a multi-scale learning system, allowing the model to analyze images across various sizes (scales) during a single pass through.

YOLOv5 is also available in various parameter sizes: nano, small, medium, large, and Xlarge. The nano size is used in our experiments because the reduced number of parameters will help mitigate the effects of overfitting on our limited dataset size.

E. Data Preparation

In our approach, the images are first compressed to 8 bits and normalized between 0 and 255 to simplify processing, reduce computational complexity, and enhance visibility.

Each image is labeled with the exact coordinates for all of its present culverts (which we will refer to as "centroids"), as well as a "Culvert" class label for each. To represent the ground truth labels, a variable-width bounding box centered around each culvert centroid is defined. In our experiments, a bounding box with side lengths of 100 pixels yielded optimal results for both model training and qualitative evaluation.

Information of all bounding boxes for a given image is then stored in each tested model's respective file format: Pascal VOC XML for Faster-RCNN, COCO for DETR models, and YOLO DarknetTXT for YOLOv5. Each image-file pair represents a single data sample for model training.

It is also important to note that only true class labels (those marked as "Culvert") are used in the data, as any non-culvert feature on an image would be an arbitrary label and could technically count as a 'False' or 'Background' classification, and thus does not provide any additional value.

To further enhance the training process and promote generalization, we employ Albumentations [26], a popular data augmentation library. This allows us to apply various transformations to the data, such as image flipping, rotation, and shifting, augmenting the dataset for improved training and generalization performance.

In all experiments, to evaluate the advanced object detection models, we divide data into train, validation, and test sets, where final evaluations were performed on a reserved and non-utilized set of data randomly drawn from the total pool of images across all geographical regions.

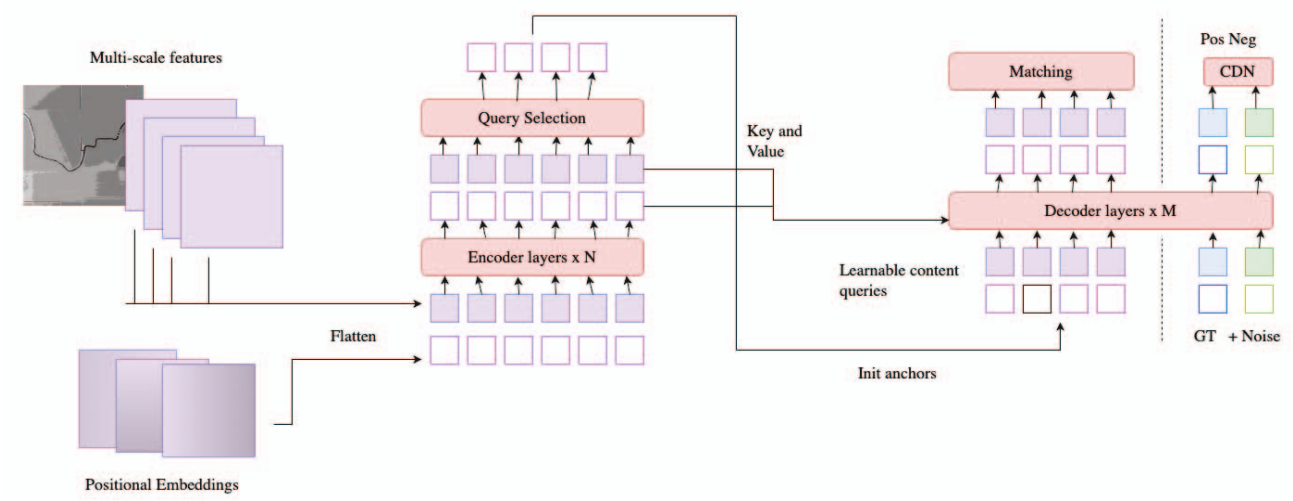


Fig. 4. The overall pipeline of DINO model applied to our use case of identifying culverts.

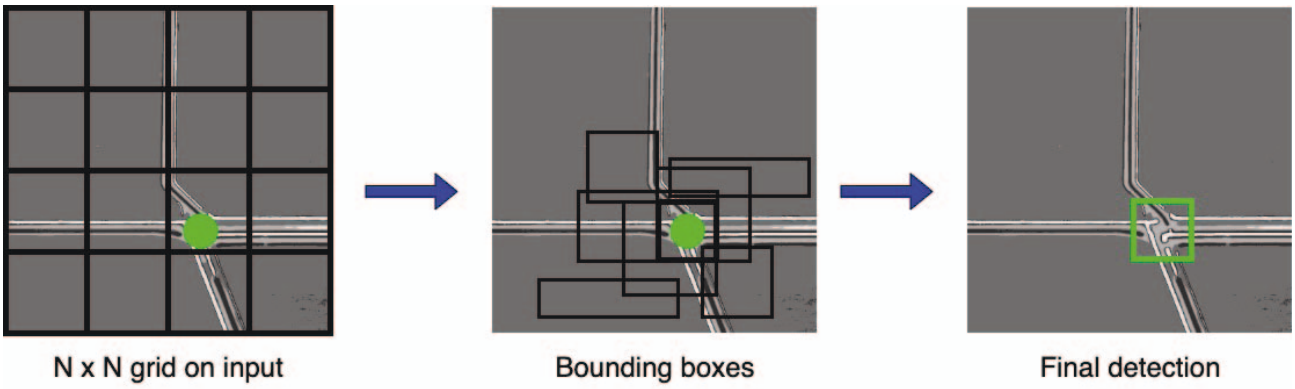


Fig. 5. Example usage of the YOLO model and architecture in culvert analysis.

V. Results and Discussion

A. Experimental Setup

Our tested models utilize publicly available pre-trained weights, which are then fine tuned on our dataset. Fine-tuning allows us to utilize prior knowledge and boosts performance on our relatively limited dataset size.

To consider which culvert predictions to keep in the presence of many potential outputs, we cannot match culverts to their nearest prediction, as this would leave us no way to select outputs in the absence of labels (such as in an applied setting, searching for culvert locations). To combat this problem, we set a ”Confidence Threshold” parameter for each model, allowing a consistent and non-label dependent method to consider which model outputs are worthy of evaluation.

Confidence Threshold is defined as a number between 0 and 1, representing the model's prediction confidence needed to consider a bounding box proposal as an official model prediction. This hyper-parameter allows us to tune the balance between a model's false positive and negative rates. A low confidence threshold results in a high prediction rate and many proposed bounding boxes, a high threshold results in much more limited predictions.

An example of the threshold values effects can be seen in Fig. 12. The model makes a set of predictions that are filtered to meet the minimum confidence threshold. As the threshold increases in value, less confident predictions are filtered out, leaving only predictions which meet the new threshold.

The exception to the threshold rule is the base DETR model, which due to architecture limitations, we select its top-k results. We find that a k of 2 (such that we are taking only the top two predictions) works best, as we do not want to severely skew results by consistently guessing several extra culvert locations when many of the images contain only a single culvert.

Notable parameters such as learning rate and detection threshold are listed in Table II. Abbreviations used in the table are defined as follows: Stochastic Gradient Descent (SGD), Adam Optimization (Adam), an alternate weight decaying version of Adam (AdamW), Learning Rate (LR), Weight

(a) (b) (c) (d) (e)

Fig. 6. Five randomly chosen images from the dataset, after preprocessing. Normalized into visual range for display. An example culvert can be seen clearly in image (d), where the roadway traveling horizontally across the screen intersects the waterway traveling vertically down through the center. It is important to note that while not apparent, all images contain at least a single culvert.

TABLE II
MODEL PARAMETERS

Model	Faster RCNN	DETR	DETR: DINO	YOLOv5
Optimizer	SGD	AdamW	AdamW	SGD
Optimizer Parameters	LR = .001 M = .9 WD = .005	LR = .0001 WD = .0001	LR = .0001 WD = .0005	LR = .01 M = .3 WD = .001
Confidence Threshold	.7	Top Two Choices	.3	.1

Decay (WD), and Momentum (M).

The Yolov5-nano model was used in our tests. The "nano" version of the model is a reduced size architecture that has significantly less parameters. This benefits our limited dataset size, allowing for more impactful fine tuning. Additionally, a much smaller detection threshold of 0.1 was used to consider this model's predictions.

B. Metrics

The unique nature of our drainage crossing (culvert) detection analysis presents challenges for conventional object detection methods. Unlike typical scenarios where multiple classes need to be distinguished, our objective is to simply identify the presence of a single class label within an image. Consequently, traditional object detection approaches may not be directly applicable. To achieve a comprehensive and accurate assessment of model performance, we propose a framework for quantitatively analyzing the models' ability to identify culverts. In evaluating the performance of a model, we consider its overall effectiveness across the following metrics:

- **Accuracy**: The accuracy of which the model correctly identifies all culverts (Accuracy).
- **IOU**: Intersection Over Union of predicted bounding boxes.
- **Extra**: Rate of predicting extra labels and bounding boxes.
- **Miss**: Rate the model misses the presence of a culvert.

Accuracy Defined as the percent of culverts correctly identified out of the total being present. Because there can be multiple culvert locations with multiple predicted locations, we must first map predictions to their most likely culvert label. First, we find the centroid for each prediction and culvert (the ground truth label centroid). Then each prediction centroid is mapped to its closest available ground truth centroid. Accuracy is then computed as a percentage of ground truth labels which have been assigned a culvert prediction. This process is visualized in Fig. 7

This value does not account for any extra predictions (e.g. if there is a single culvert present in an image and two predictions are made, only the correctly identified culvert is counted). We compensate for this discrepancy with the "Extra" value.

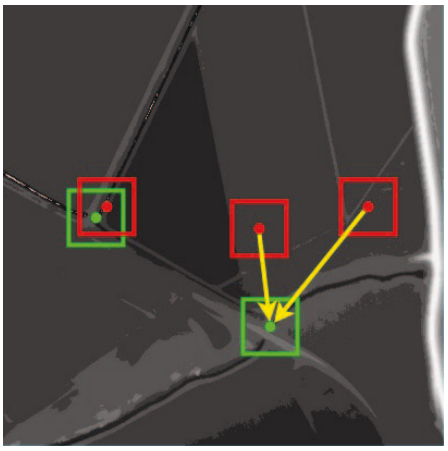


Fig. 7. Example assignment of predictions (red) to their closest possible label (green) before computing performance metrics. The two furthest right predictions would count as both a miss and a single "Extra" label.

IOU Intersection over union is used to account for the accuracy of the predicted bounding boxes which have been mapped to each label. While accuracy reveals when the model correctly identifies the presence of a culvert, we must analyze whether the predicted bounding box correctly and precisely locates the culvert in the image.

IOU is the value used to describe the percentage overlap of two shapes. It takes the intersection of two shapes (area of overlap) and divides by the union (the area of both shapes together) and is defined in (1). A visualization of the equation is also available in Fig. 8.

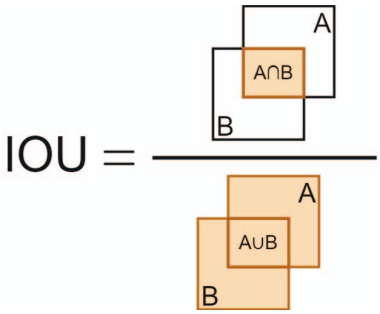


Fig. 8. A visualization of Intersection Over Union, which computes the percentage overlap of two shapes.

$$IOU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{1}$$

Extra Accounts for the rate at which a given model provides *too many* bounding boxes. As defined in (2) the Extra value is computed as the number of extra labels predicted (that are not mapped to a potential ground truth label) over the total number of ground truth labels. For example, if there are 3 culverts present in an image and the model predicts 4, the number of extra labels would be 1. This would result in an Extra rate of $1/3$ or, 1 extra prediction in 3 labels.

$$Extra = \frac{\text{Extra predicted labels}}{\text{Total True Labels}} \tag{2}$$

Miss accounts for the rate at which a given model provides *too little* bounding boxes. We define Miss, shown in (3), as the number of unpredicted labels present in a given image, out of the total. For example, if there are 3 culverts present in an image and the model predicts bounding boxes for 2, this would result in a Miss rate of $1/3$, or 1 miss in 3 labels.

$$Miss = \frac{\text{Unpredicted Labels}}{\text{Total True Labels}} \tag{3}$$

Quantitative model performance is based on the average values for each of the above metrics averaged across all validation data samples.

C. Results and Discussion

State of the art object detection techniques each offer unique advantages and disadvantages. As will be seen in the following results, architecture changes can drastically change both model performance and behavior. These differences are further exacerbated by our limited data set size of six thousand samples, as real world data is not always available in the quantities commonly used for newer state of the art models. Decreased data quantity poses an increased risk of both overfitting to limited samples, as well as underfitting when models have too many parameters to train and not enough samples to do so.

Culvert detection provides a unique challenge to traditional object detection algorithms. Culverts do not have distinct object boundaries the way a physical object, such as a ball or a car would. A culvert, for the purpose of model training, is defined not only by its visible features but also its bounding box label which may or may not include non-relavent background. In essence, the ”signature” (features that indicate an object) are much more loosely defined and rely partly on a bounding box label rather than just visible features.

There is also the additional challenge of terrain resolution. Because images are normalized for model use, minute high-resolution details that may have once been present in LiDAR imaging can be lost. This in turn results in instances of very difficult to identify culverts. One such example can be seen in Fig. 13 (a) and (c), where the model classifies the roadway rather than the culvert because its loss in resolution renders the culvert nearly invisible.

Thresholding model predictions according to confidence played a large role in understanding model performance. Fig. 12 outlines the same model (DETR: DINO) generating predictions across three different threshold values. Model architecture has a large impact on how sensitive model confidence is on potentially identified culverts. This means tuning the threshold value for each model is key in optimizing its performance.

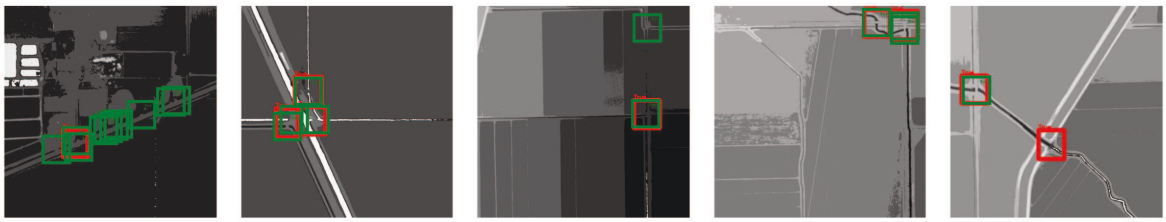
Because model architectures have varying degrees of sensitivity when generating confidence for their predictions, thresholding adjustment allows us to tune how a model is interpreted for evaluation. If a model is highly sensitive, high threshold values can be used to reign in overly confident but poorly placed predictions. Conversely, if a model's architecture results in overall low confidence scores, we can adjust the threshold value to be lower, fine tuning what model output will be considered as a prediction. We find that each architecture must strike its own balance. Threshold values for our experiments can be found in Table II.

Table III outlines the overall results of all models. The highest accuracy model was DETR: DINO, while the model with the most accurate bounds was Faster-RCNN. While it may seem that DETR:DINO lacks in performance due to its notably high Extra rate, we will see in the following sections that upon further qualitative analysis, it provides accurate and competitive predictions even when considering its large number of extra predictions.

TABLE III
PERFORMANCE BY MODEL.

Model	Accuracy	IOU	Miss	Extra
Faster RCNN	.723	**.557**	.277	.194
DETR	.510	.004	.364	.100
DETR: DINO	**.778**	.024	**.221**	.507
YOLOv5	.335	.358	.665	**.016**

*1) **Faster-RCNN**:* The Faster RCNN model shows clear promise in its performance. Fig. 9 shows various examples generated from the model to display its different behaviors. While there are some images such as in Fig. 9 (a) in which the model massively over-predicts, it has a high IOU rate and low overall Extra rate (Table III) , indicating over predictions do not happen frequently, and that there is a high rate of overlap



(a) (b) (c) (d) (e)

Fig. 9. A variety of validation samples generated by the Faster-RCNN model. Model predictions are denoted in red. (a) shows an instance where the model predicts only a single culvert when there are many, (b) shows a handful of quite precise predictions, (c) shows a clear miss, (d) is another accurate prediction, (e) is an instance where the model was capable of finding a culvert that was missed in the data labeling process.

between true and predicted bounding boxes. In addition to precise bounding boxes in the remaining images it is particularly notable to see its ability to classify and locate the unlabeled culvert in image (e) that was initially overlooked by those creating the dataset.

*2) **DETR**:* DETR performance was somewhat hindered by its top-k selection of bounding boxes. This is consequently the reason behind its low extra rate and should thus be disregarded for this model (Table III). Additionally its low IOU indicates that the bounds it did detect, were rarely accurate. The primary issue with its performance, as can be seen in Fig. 10, is its tendency to generate large enveloping predictions, resulting in a low IOU.

*3) **DETR: DINO**:* The choice of the detection threshold value has a significant impact on the DINO model's performance, with particular implications observed in the context of culvert detection. Fig.12 illustrates the diverse behaviors of the DINO model across various examples, with distinct outcomes observed for three different threshold values, demonstrating its particular sensitivity in confidence values for predictions. Adjusting the threshold from 0.3 to 0.1 increases the number of detected culverts and also increases false positives. For a threshold of 0.1, the model produces 270% more bounding boxes than there are objects, resulting in an inflated accuracy and very high extra rate. For a threshold of 0.2, the model results in 100% more bounding boxes, with an 88% accuracy. However, when the threshold is 0.3, the model generates only 50% more boxes, resulting in an accuracy drop of 77%. Because of this, a model threshold of 0.2 was used in its final results.

DETR:DINO achieves the highest accuracy of the tested models, however its sensitivity in prediction confidence results in many extra bounding predictions (Table III). This consequently decreases the model's IOU rating drastically, as many extra predictions that do not fall on a correct bounding box have no intersection.

Despite its low IOU, a qualitative analysis of sample outputs such as in Figure 11 show that its predictions can be quite accurate, particularly when its extra predictions are correctly limited.

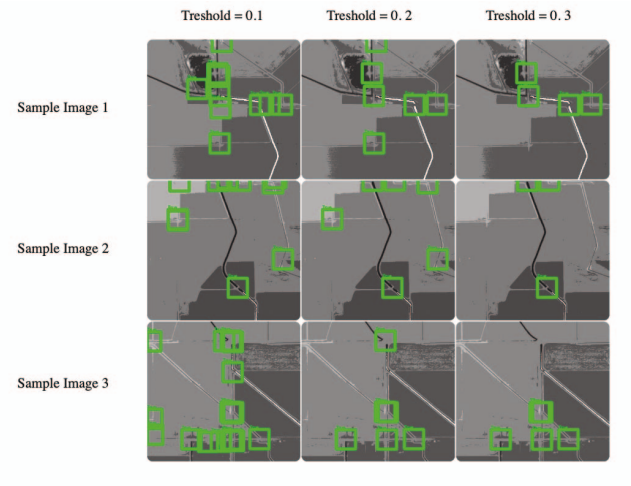
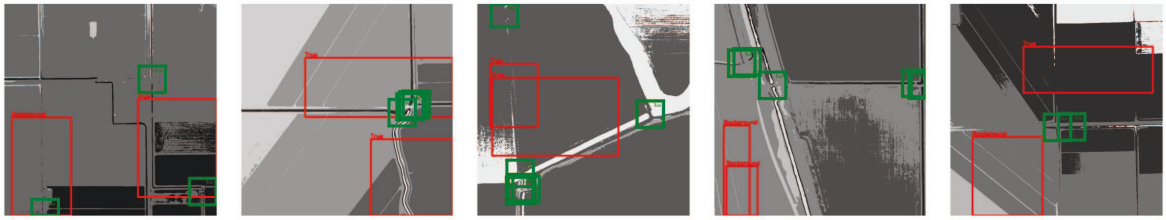


Fig. 12. Validation samples generated by DINO model over various threshold values. Model predictions are outlined in green. Notice how threshold increase filters out lower confidence predictions.

*4) **YOLOv5**:* Our YOLOv5 model had lower than expected overall accuracy (Table III). A notable result is the stunningly low "Extra" rate, as the model was incredibly reserved in its confidence values. The model was evaluated on a confidence threshold of 0.1 (meaning only a confidence of 10% was needed to count a prediction) and yet it still has an extra rate of approximately 1.6%. This means the drop in accuracy is largely due to a lack of predictions, with a notably small confidence across many predictions. Low confidence values are likely a consequence of the architecture, as its output contains a very large number of predictions, resulting in small probability differences between each prediction. Examples can be seen in Fig. 13 in which single predictions (as indicated by the low extra rate) are either relatively accurate, or completely off.

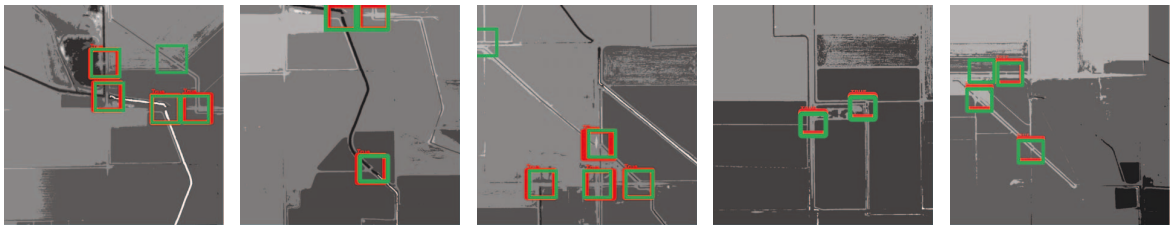
VI. Conclusions

In this study, we developed an efficient methodology for detecting drainage structures, specifically culverts and bridges, using DL based object detection frameworks. Through quantitative and qualitative analyses, we found that DINO:DETR and Faster-RCNN methods outperformed DETR and YOLOv5 in



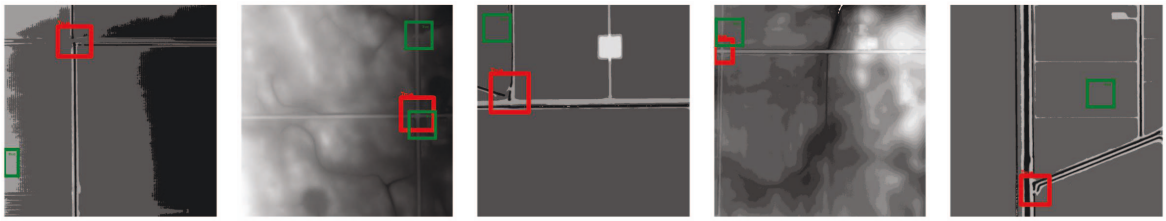
(a) (b) (c) (d) (e)

Fig. 10. A series of DETR output examples showcasing model tendency to create bounds that are far too large, resulting in low IOU. As well as examples of misplaced bounds (model output is red, culvert labels are green).



(a) (b) (c) (d) (e)

Fig. 11. A series of DETR:DINO output examples (model output is red, culvert label is green).



(a) (b) (c) (d) (e)

Fig. 13. A series of YOLOv5 output examples (model output is red, culvert label is green).

accurately identifying culvert locations. Our research emphasizes the potential of these advanced DL models for object detection in improving the delineation of hydro-topographic features and enhancing our understanding of hydrologic connectivity. Applying these DL frameworks for detecting drainage structures represents a new approach in the field of hydrologic connectivity analysis. Our findings underscore the ability of these models to significantly enhance the accuracy and efficiency of hydro-topographic delineation, especially in areas where flow barriers create challenges for water movement.

VII. Future Work

It still remains to be seen how well models trained on a single geographical region can transfer to different regions, particularly those outside the US. Additionally, it may be fruitful to explore how different architectures perform on objects with more or less distinct signatures (for example, a culvert bounding box is more loosely defined than the presence of a person, or visibly apparent object). There is also a potential that different geographical archetypes and terrain types (such as mountains, flatlands, desert, etc.) may be more or less difficult to analyze.

VIII. Acknowledgements

This work is supported by a grant awarded by the National Science Foundation (1951741). We would also like to thank Google Cloud for awarding us credits for GCP computing resources.

References

[1] M. Stieglitz, J. Shaman, J. McNamara, V. Engel, J. Shanley, and G. W. Kling, "An approach to understanding hydrologic connectivity on the hillslope and the implications for nutrient transport," *Global biogeochemical cycles*, vol. 17, no. 4, 2003.

[2] I. Thomas, P. Jordan, P.-E. Mellander, O. Fenton, O. Shine, D. Ó hUallacháin, R. Creamer, N. T. McDonald, P. Dunlop, and P. N. Murphy, "Improving the identification of hydrologically sensitive areas using lidar dems for the delineation and mitigation of critical source areas of diffuse pollution," *Science of the Total Environment*, vol. 556, pp. 276–290, 2016.

[3] J. N. Callow, K. P. Van Niel, and G. S. Boggs, "How does modifying a dem to reflect known hydrology affect subsequent terrain analysis?" *Journal of hydrology*, vol. 332, no. 1-2, pp. 30–39, 2007.

[4] V. Lukas and V. Baez, "3d elevation program—federal best practices," US Geological Survey, Tech. Rep., 2021.

[5] G. D. Duke, S. W. Kienzle, D. L. Johnson, and J. M. Byrne, "Improving overland flow routing by incorporating ancillary road data into digital elevation models," *Journal of Spatial Hydrology*, vol. 3, no. 2, 2003.

[6] R. Li, Z. Tang, X. Li, and J. Winter, "Drainage structure datasets and effects on lidar-derived surface flow modeling," *ISPRS International Journal of Geo-Information*, vol. 2, no. 4, pp. 1136–1152, 2013.

[7] G. Sofia, G. D. Fontana, and P. Tarolli, "High-resolution topography and anthropogenic feature extraction: Testing geomorphometric parameters in floodplains," *Hydrological Processes*, vol. 28, no. 4, pp. 2046–2061, 2014.

[8] J. B. Lindsay and K. Dhun, "Modelling surface drainage patterns in altered landscapes using lidar," *International Journal of Geographical Information Science*, vol. 29, no. 3, pp. 397–411, 2015.

[9] S. Bhadra, R. Li, D. Wu, G. Wang, and B. Rekabdar, "Assessing the impacts of anthropogenic drainage structures on hydrologic connectivity using high-resolution digital elevation models," *Transactions in GIS*, vol. 25, no. 5, pp. 2596–2611, 2021.

[10] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497

[11] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," *CoRR*, vol. abs/2005.12872, 2020. [Online]. Available: https://arxiv.org/abs/2005.12872

[12] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "Dino: Detr with improved denoising anchor boxes for end-to-end object detection," 2022.

[13] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios," 2021.

[14] S. Talafha, D. Wu, B. Rekabdar, R. Li, and G. Wang, "Classification and feature extraction for hydraulic structures data using advanced cnn architectures," in *2021 Third International Conference on Transdisciplinary AI (TransAI)*. IEEE, 2021, pp. 137–146.

[15] D. Wu, R. Li, B. Rekabdar, C. Talbert, M. Edidem, and G. Wang, "Classification of drainage crossings on high-resolution digital elevation models: A deep learning approach," *GIScience & Remote Sensing*, 2023.

[16] C. Ye, Y. Li, P. Cui, L. Liang, S. Pirasteh, J. Marcato, W. N. Goncalves, and J. Li, "Landslide detection of hyperspectral remote sensing data based on deep learning with constrains," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 12, pp. 5047–5060, 2019.

[17] W. Li and C.-Y. Hsu, "Automated terrain feature identification from remote sensing imagery: a deep learning approach," *International Journal of Geographical Information Science*, vol. 34, no. 4, pp. 637–660, 2020.

[18] Z. Xu, S. Wang, L. V. Stanislawski, Z. Jiang, N. Jaroenchai, A. M. Sainju, E. Shavers, E. L. Usery, L. Chen, Z. Li *et al.*, "An attention u-net model for detection of fine-scale hydrologic streamlines," *Environmental Modelling & Software*, vol. 140, p. 104992, 2021.

[19] N. Van Nieuwenhuizen, "An analysis of preprocessing techniques for the removal of transportation embankments and surface roughness in fine-resolution lidar dems," Ph.D. dissertation, University of Guelph, 2021.

[20] R. Li, D. Wu, B. Rekabdar, and G. Wang, "Evaluating lidar-based elevation-derived hydrography in low-lying agricultural landscapes," in *AGU Fall Meeting Abstracts*, vol. 2022, 2022, pp. H55K–0708.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[22] S. Liu, F. Li, H. Zhang, X. Yang, X. Qi, H. Su, J. Zhu, and L. Zhang, "Dab-detr: Dynamic anchor boxes are better queries for detr," 2022.

[23] F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, and L. Zhang, "Dn-detr: Accelerate detr training by introducing query denoising," 2022.

[24] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," 2021.

[25] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015. [Online]. Available: http://arxiv.org/abs/1506.02640

[26] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: fast and flexible image augmentations," *Information*, vol. 11, no. 2, p. 125, 2020.