

# Journal of Applied Hydrography

HYDROGRAPHISCHE NACHRICHTEN

10/2026

HN 135

Hydrographie  
im  
Nahbereich



# Automatic detection and multiclass segmentation of underwater pipelines

## A comparative study on different approaches using computational models

An article by SETHMIYA HERATH MUDIYANSELAGE

Underwater pipeline inspection is important in several aspects. Computer vision-based detection and segmentation have played a crucial role in underwater pipeline inspections in the last few decades. This study focuses on implementing a deep learning-based workflow for underwater pipeline detection and segmentation using YOLOv8 small and large, and Mask R-CNN ResNet 50 and 101 models. All four models could fulfil both detection and segmentation approaches in a single model. The dataset used consists of pipes, leakages, concrete mats, concrete weights and pipe couplings. The performance of the models was evaluated based on the loss function, precision, inference time and visualisation results. Experimental results indicate that YOLOv8 outperforms Mask R-CNN in accuracy and real-time detection. YOLOv8 segmentation small model achieved the highest Mean Average Precision (50–95) as 77 % in object detection and 74 % in segmentation. The YOLO segmentation large model achieved the highest class-wise Mean Average Precision (50–95) as 97 % in the pipe class. In future work, the YOLOv8 model will be modified to derive the real-time location of autonomous underwater vehicles, and Mask R-CNN models will be modified to improve the precision.

pipeline detection | Mask R-CNN | YOLOv8  
Erkennung von Rohrleitungen | Mask R-CNN | YOLOv8

Die Inspektion von Unterwasserrohrleitungen ist in mehrfacher Hinsicht von Bedeutung. Die auf Computer-Vision basierende Erkennung und Segmentierung hat in den letzten Jahrzehnten eine entscheidende Rolle bei der Inspektion von Unterwasserrohrleitungen gespielt. Diese Studie konzentriert sich auf die Implementierung eines auf Deep Learning basierenden Arbeitsablaufs für die Erkennung und Segmentierung von Unterwasserrohrleitungen unter Verwendung der Modelle YOLOv8 (klein und groß) sowie Mask R-CNN ResNet 50 und 101. Alle vier Modelle konnten sowohl Erkennungs- als auch Segmentierungsaufgaben in einem einzigen Modell erfüllen. Der verwendete Datensatz besteht aus Rohren, Leckagen, Betonmatten, Betongewichten und Rohrkupplungen. Die Leistung der Modelle wurde anhand der Verlustfunktion, der Präzision, der Inferenzzeit und der Visualisierungsergebnisse bewertet. Die Versuchsergebnisse zeigen, dass YOLOv8 Mask R-CNN in Bezug auf Genauigkeit und Echtzeit-Erkennung übertrifft. Das kleine YOLOv8-Segmentierungsmodell erzielte die höchste mittlere durchschnittliche Präzision (50–95) mit 77 % bei der Objekterkennung und 74 % bei der Segmentierung. Das große YOLO-Segmentierungsmodell erzielte die höchste klassenweise mittlere durchschnittliche Präzision (50–95) von 97 % in der Rohrklasse. In zukünftigen Arbeiten wird das YOLOv8-Modell modifiziert, um die Echtzeitposition autonomer Unterwasserfahrzeuge zu ermitteln, und die Mask-R-CNN-Modelle werden modifiziert, um die Präzision zu verbessern.

### Author

Sethmiya Herath  
Mudiyansele studied  
Hydrography at HafenCity  
University Hamburg, where  
he is currently working as a  
research assistant.

sethmiya.mudiyansele@hcu-hamburg.de

### 1 Introduction and motivation

Underwater pipelines play a crucial role in the oil and gas industry. Regular inspections are essential to ensure their integrity and determine necessary maintenance work. Leaks or spills from damaged pipelines can have severe consequences for marine ecosystems and the economy. Early identification and routine inspections are essential to mitigate severe damage (Li et al. 2022; Martin-Abadal et al. 2021).

Vision-based approaches play an important role in pipeline detection compared to traditional echo-based methods. Detection and segmentation are two approaches in digital image processing. YOLOv8 and Mask R-CNN are advanced deep learning models that combine object detection and segmentation in a single framework. YOLOv8 is optimised for real-time performance, offering high speed and efficiency while maintaining

strong accuracy. In contrast, Mask R-CNN focuses more on precise segmentation and feature extraction, often achieving high accuracy but at slower speeds (Hussain 2024).

This research will examine the capabilities of underwater pipeline detection and segmentation using YOLOv8 and Mask R-CNN models, and compare their applicability qualitatively using the testing dataset and quantitatively using the evaluation matrix. Also, the research evaluates the real-time detection capabilities using inference time. Data augmentation methods are also applied to address the class imbalance problem in this research.

## 2 Methodology

A Slovenian research team collected experimental data using a remotely operated vehicle (ROV) to record underwater pipelines from multiple camera angles. After capturing the videos, they have extracted two frames per second to create the dataset. Fig. 1 shows some random images of the dataset (Gašparović et al. 2022).

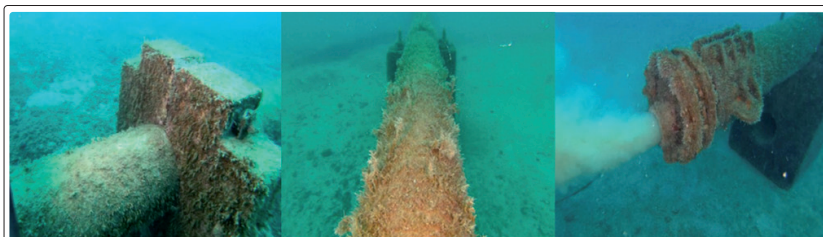
Fig. 2 depicts the research methodological pipeline. Images were manually labelled using the Roboflow open-source labelling platform, which allows multi-class labelling. Manual labelling was used to create a high-quality training dataset. The initial dataset consists of 1,008 images. The dataset is divided into 70 %, 20 % and 10 % ratios for training, validation and testing randomly after the labelling process. Training and validation datasets were used in training of the models, and the testing unseen dataset was used for qualitative analysis. To address the class imbalance problem, 126 random images were generated using geometric data augmentation. The dataset consists of images of pipes, pipe couplings, concrete weights, concrete mats and leakages.

Class information and label waypoints were stored in the Mask R-CNN-understandable COCO.json format and the YOLOv8-understandable .txt format. Google Colab paid version was used as the training platform, which provides powerful GPUs. Results were evaluated quantitatively using the evaluation matrix and qualitatively using the loss function and visualisation results.

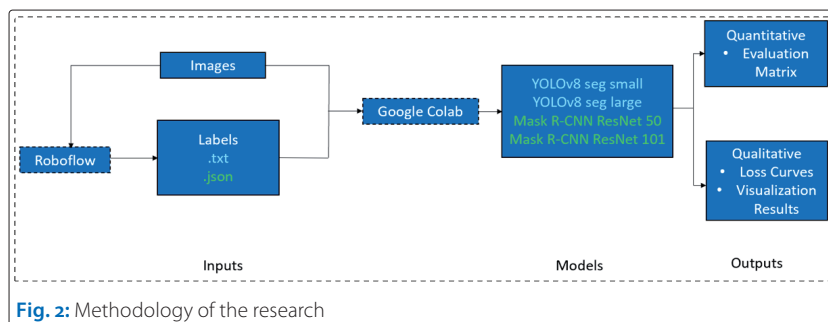
The leakage class is specially designed considering the leakage pattern rather than the leakage point to increase the number of pixels to address the class misbalance problem. Table 1 shows the sizes of the training, validation and testing samples of the initial dataset (version 4) and the dataset with the initial and augmented images (version 5).

| Version | Training | Validation | Testing |
|---------|----------|------------|---------|
| V4      | 739      | 212        | 105     |
| V5      | 791      | 227        | 113     |

**Table 1:** Dataset separation for training, validation and testing



**Fig. 1:** Random images from the dataset



**Fig. 2:** Methodology of the research

| Class name      | Instances (V4) | Instances (V5) |
|-----------------|----------------|----------------|
| Concrete Mat    | 364            | 379            |
| Concrete Weight | 763            | 803            |
| Leakage         | 117            | 148            |
| Pipe            | 1085           | 1281           |
| Pipe Coupling   | 88             | 124            |

**Table 2:** Number of instances of each class

Table 2 represents the number of instances in each class of the version 4 and 5 datasets. Generating images using data augmentation mostly focused on underrepresented pipe coupling and leakage classes.

The same training parameters are used for evaluating model performance during training. Batch size refers to the number of training samples processed before updating the model weights. Epoch size characterises one complete pass through the entire dataset during training. Learning size refers to the step size to reach local minima of the optimiser. A batch size of 8, a learning rate of 0.001 and a number of epochs of 30 were used in the training process.

### Model architecture

Fig. 3 and Fig. 4 depict the model architecture of YOLOv8 and Mask R-CNN. The YOLOv8 architecture follows a streamlined pipeline where the input image is first processed through a backbone to extract features at multiple scales, then passed into a feature aggregation stage (FPN) to enhance semantic information and finally into a detection head that directly predicts bounding boxes, classes and segmentation masks in a single forward pass. This design emphasises speed and efficiency for real-time tasks. In contrast, Mask R-CNN uses a two-stage approach: a backbone

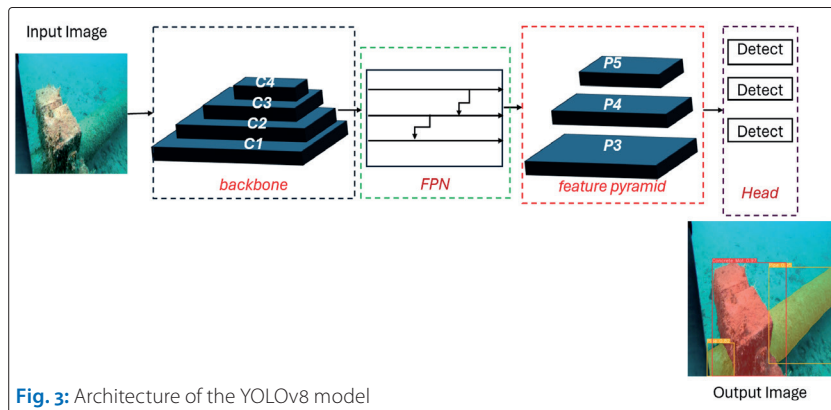


Fig. 3: Architecture of the YOLOv8 model

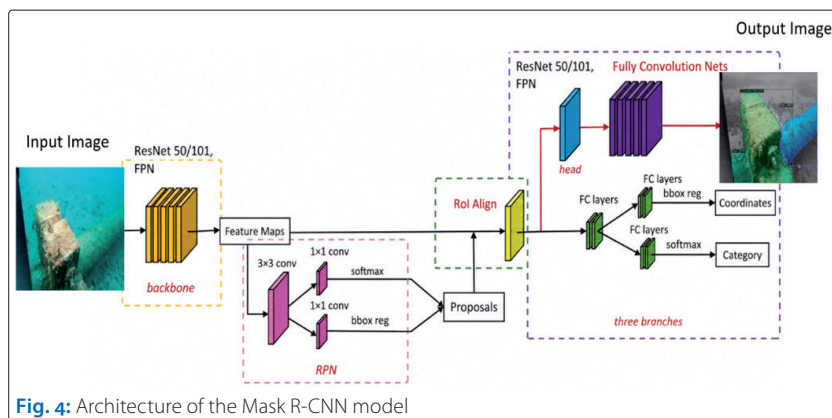


Fig. 4: Architecture of the Mask R-CNN model

with FPN extracts features, followed by a Region Proposal Network (RPN) that generates candidate object regions. These regions are refined using RoI Align and passed through separate branches for classification, bounding box regression and pixel-level mask prediction, making it more precise but computationally heavier. Mask R-CNN ResNet 50 and 101 consist of ResNet 50 and 101 backbones, respectively. YOLOv8 small model has fewer parameters compared to the large model.

### 3 Results, analysis and discussion

Models were evaluated using quantitative and qualitative analysis.

#### 3.1 Loss function

Analysis of the loss function is important to avoid overfitting of the model. YOLO models, particularly in the early stages of training, show more fluctuations in validation loss. This volatility indicates instability, which is common in object detection and segmentation tasks due to the complex nature. YOLO models typically prioritise speed and real-time inference, and such fluctuations might reflect trade-offs in their design, especially when optimising for both accuracy and efficiency. On the other hand, Mask R-CNN models demonstrate a smoother convergence, with validation loss decreasing more stably due to their network architecture. However, training and validation loss curves showed smooth distribution after 15 epochs, and training was stopped after 30 epochs, as shown in Fig. 5.

It is difficult to compare the loss values of models numerically because YOLOv8 uses a combination of binary cross-entropy loss and dice loss, while Mask RCNN uses binary cross-entropy loss for segmentation. This is the reason to give a higher loss value in YOLOv8 models compared to Mask RCNN models. YOLOv8 uses the AdamW optimiser, and Mask RCNN uses the Stochastic Gradient Descent (SGD) optimiser. AdamW optimisers prioritise speed and reach the local minima faster compared to SGD optimisers, which is the reason for the inconsistency of the loss function in early epochs of YOLO models.

#### 3.2 Quantitative analysis

Results were analysed quantitatively using evaluation matrices as described below.

##### Precision-recall (P-R) curve

The precision-recall curve plots the recall on the X-axis and precision on the Y-axis, based on the intersection over union (IoU) of threshold values. The P-R curve of the YOLO segmentation small model version 5 was illustrated in Fig. 6 for demonstration purposes. This is an example for Average Precision @ IoU 50. Average precision can be calculated by integrating the area under the curve, as Equation 1. The blue line represents the P-R curve for model

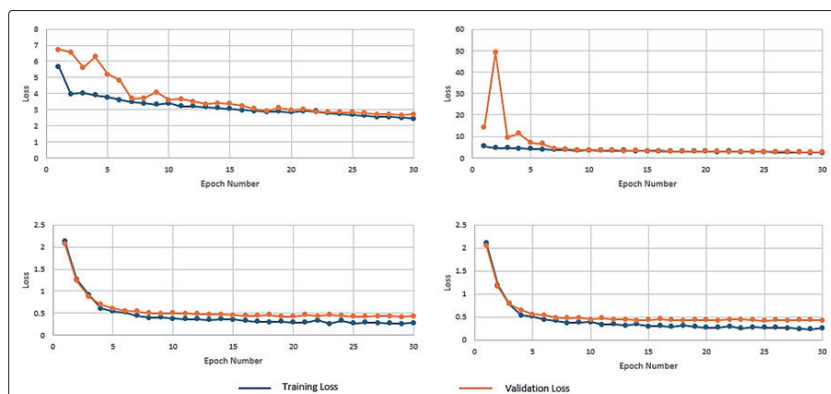


Fig. 5: Loss curve of four models (top left and right corners – YOLOv8 segmentation small and large models, bottom left and right corners – Mask R-CNN ResNet 50 and 101 models, respectively)

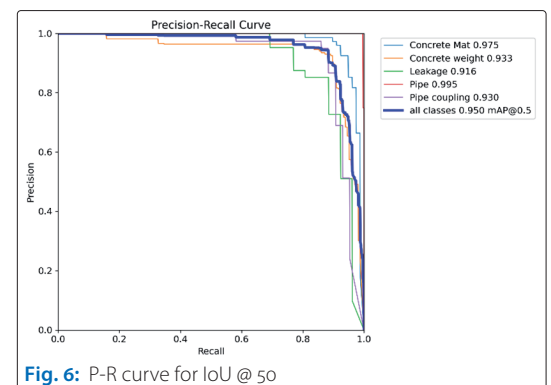


Fig. 6: P-R curve for IoU @ 50

performance @ IoU 50. Other lines represent class-wise P-R curves.

$$AP = \int_0^{\text{Threshold}} P(R) dR$$

**Equation 1:** Average Precision (P stands for Precision, R stands for Recall, e.g., threshold value is 50 for Average Precision @ IoU 50)

$$mAP_{50-95} = 1/10 \sum_{t=0.50}^{0.95} AP(t)$$

**Equation 2:** Mean Average Precision (draw the P-R curves for IoU 50 to 95 in 10 thresholds with 5 % step size, calculate the area under the curves of all 10 graphs, and compute the average)

YOLOv8 segmentation models improve notably with training compared to their base model performance, according to Table 3. Both YOLO models achieve above 72 % detection and segmentation mAP (50–95). All four models achieve more than 90 % average precision AP @ 50, showing remarkable performance of the models for the underwater environment. mAP (50–95) matrix should be used to identify small objects by the models rather than AP @ 50. The Mask R-CNN ResNet 101 model achieves higher detection and segmentation accuracy than the ResNet 50 model. YOLOv8 outperforms Mask R-CNN, suggesting its applicability for underwater scenarios. Results also suggest that Mask R-CNN models need more training samples. bbox stands for detection, and segm stands for segmentation. AP stands for average precision @ IoU 50, and mAP (50–95) stands for Mean Average Precision @ IoU 50 to 95 %.

The mAP (50–95) matrix was used for the class-wise results analysis, as shown in Table 4. The pipe class shows the highest mean average precision in all four models. Leakage detection posed a significant challenge for all evaluated models, showing lower mAP (50–95) scores. This is attributed to the irregular and diffuse nature of leakage patterns, which complicates feature discrimination.

YOLOv8 showed particularly strong performance in detecting structurally well-defined objects such as concrete mats, concrete weights and pipes, indicating robust generalisation across varying object scales and geometries. Notably, the smaller YOLOv8 variant marginally outperformed the larger model in certain categories, suggesting that increased model complexity does not necessarily yield proportional performance gains. In contrast, Mask R-CNN demonstrated reduced accuracy in several classes, particularly for under-represented categories such as leakage and pipe

| Model                 | mAP (50–95) |       | Base model   |              | mAP(50–95)   |              | AP(50) |      |
|-----------------------|-------------|-------|--------------|--------------|--------------|--------------|--------|------|
|                       | bbox        | segm  | bbox         | segm         | bbox         | segm         | bbox   | segm |
| YOLOv8 seg small      | 0.446       | 0.368 | <b>0.777</b> | <b>0.747</b> | <b>0.962</b> | <b>0.965</b> |        |      |
| YOLOv8 seg large      | 0.523       | 0.426 | <b>0.774</b> | <b>0.724</b> | <b>0.961</b> | <b>0.959</b> |        |      |
| Mask R-CNN ResNet 50  | 0.340       | 0.336 | 0.677        | 0.668        | 0.904        | 0.922        |        |      |
| Mask R-CNN ResNet 101 | 0.382       | 0.354 | 0.691        | 0.665        | 0.937        | 0.934        |        |      |

**Table 3:** Overall model performance on the validation dataset

| Model                 | Concrete mat |       | Concrete weight |       | Leakage      |              | Pipe         |              | Pipe coupling |              |
|-----------------------|--------------|-------|-----------------|-------|--------------|--------------|--------------|--------------|---------------|--------------|
|                       | bbox         | segm  | bbox            | segm  | bbox         | segm         | bbox         | segm         | bbox          | segm         |
| YOLOv8 seg small      | 0.806        | 0.797 | 0.698           | 0.627 | 0.677        | 0.627        | <b>0.966</b> | <b>0.969</b> | <b>0.736</b>  | <b>0.715</b> |
| YOLOv8 seg large      | 0.810        | 0.792 | 0.686           | 0.609 | 0.632        | 0.544        | <b>0.976</b> | <b>0.973</b> | <b>0.737</b>  | <b>0.747</b> |
| Mask R-CNN ResNet 50  | 0.758        | 0.741 | 0.619           | 0.615 | <b>0.509</b> | <b>0.489</b> | 0.899        | 0.909        | 0.602         | 0.587        |
| Mask R-CNN ResNet 101 | 0.728        | 0.736 | 0.602           | 0.612 | 0.575        | 0.507        | 0.925        | 0.918        | 0.623         | 0.552        |

**Table 4:** Class-wise Mean Average Precision results

couplings, highlighting its sensitivity to the size of the training dataset.

### Data augmentation

The comparison between version 4 and version 5 indicates minimal overall performance change. A modest improvement in segmentation mAP (50–95) was observed for ResNet 50 and ResNet 101 models from 66 % to 69 %, as shown in Table 5.

Class-wise analysis revealed notable improvement for previously underrepresented classes,

| Model                 | Version | mAP(50–95) |              |
|-----------------------|---------|------------|--------------|
|                       |         | bbox       | segm         |
| YOLOv8 seg small      | v4      | 0.777      | 0.747        |
|                       | v5      | 0.767      | 0.742        |
| YOLOv8 seg large      | v4      | 0.774      | 0.724        |
|                       | v5      | 0.772      | 0.736        |
| Mask R-CNN ResNet 50  | v4      | 0.677      | 0.668        |
|                       | v5      | 0.690      | <b>0.692</b> |
| Mask R-CNN ResNet 101 | v4      | 0.691      | 0.665        |
|                       | v5      | 0.689      | <b>0.693</b> |

**Table 5:** Comparison of Mean Average Precision @ IoU 50–95 of version 4 and 5 datasets

| Model                 | Version | Concrete mat |       | Concrete weight |       | Leakage |              | Pipe  |       | Pipe coupling |              |
|-----------------------|---------|--------------|-------|-----------------|-------|---------|--------------|-------|-------|---------------|--------------|
|                       |         | bbox         | segm  | bbox            | segm  | bbox    | segm         | bbox  | segm  | bbox          | segm         |
| YOLOv8 seg small      | v4      | 0.806        | 0.797 | 0.698           | 0.627 | 0.677   | 0.627        | 0.966 | 0.969 | 0.736         | 0.715        |
|                       | v5      | 0.806        | 0.793 | 0.696           | 0.609 | 0.646   | 0.605        | 0.951 | 0.958 | 0.737         | 0.747        |
| YOLOv8 seg large      | v4      | 0.810        | 0.792 | 0.686           | 0.609 | 0.632   | 0.544        | 0.976 | 0.973 | 0.737         | 0.747        |
|                       | v5      | 0.815        | 0.798 | 0.688           | 0.613 | 0.609   | 0.590        | 0.965 | 0.957 | 0.783         | 0.723        |
| Mask R-CNN ResNet 50  | v4      | 0.758        | 0.741 | 0.619           | 0.615 | 0.509   | 0.489        | 0.899 | 0.909 | 0.602         | 0.587        |
|                       | v5      | 0.738        | 0.741 | 0.621           | 0.598 | 0.542   | <b>0.563</b> | 0.896 | 0.884 | 0.654         | <b>0.683</b> |
| Mask R-CNN ResNet 101 | v4      | 0.728        | 0.736 | 0.602           | 0.612 | 0.575   | 0.507        | 0.925 | 0.918 | 0.623         | 0.552        |
|                       | v5      | 0.705        | 0.704 | 0.619           | 0.600 | 0.555   | <b>0.577</b> | 0.907 | 0.902 | 0.653         | <b>0.681</b> |

**Table 6:** Comparison of class-wise mAP @ IoU 50–95 of version 4 and 5 datasets

particularly pipe coupling and leakage, highlighting the effectiveness of data augmentation in the segmentation domain. Segmentation results of Mask R-CNN models show a significant improvement in mAP (50–95), proving that Mask R-CNN models highly depend on the data size, as shown in Table 6.

3.3 Qualitative analysis

All models successfully segmented leakages, achieving nearly 100 % detection and segmentation confidence scores on the unseen testing dataset, as shown in Fig. 7. This reflects the strong performance of the training models in leakage detection and segmentation. Furthermore, the leakage class appears to exhibit sufficiently distinctive visual features, such as colour contrast and boundary definition, which enable accurate segmenta-

tion even under low-visibility conditions. Fig. 7 and Fig. 8 depict the input image in the first column, the ground truth mask in the second column and the prediction results of each model in the third and fourth columns.

Fig. 8 shows the detection and segmentation of a concrete mat.

3.4 Real-time detection and segmentation capabilities

Inference time is an indicator of the model’s applicability for real-time detection and segmentation. YOLOv8 segmentation models have significantly lower inference times than Mask R-CNN models, as shown in Table 7. Specifically, the YOLOv8 small and large models have 1.2 and 3.1 milliseconds of inference speeds on an NVIDIA A100 GPU, which proves their significant capability of real-time de-

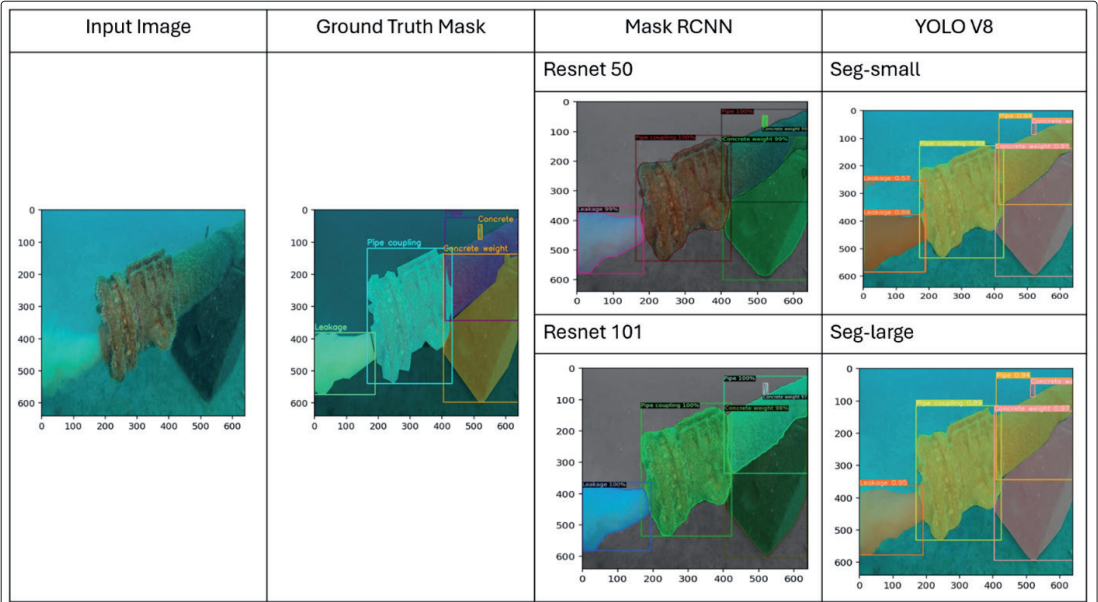


Fig. 7: Visualisation of leakage, pipe, pipe coupling and concrete weight detection and segmentation

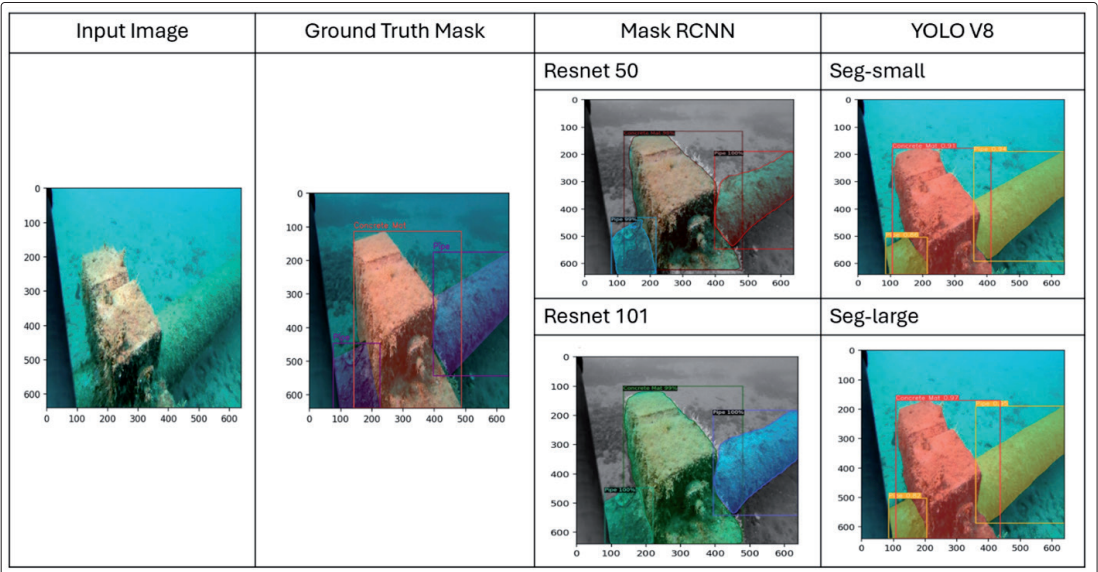


Fig. 8: Pipe and concrete mat detection and segmentation

tection and segmentation compared to the Mask R-CNN models.

4 Conclusion and outlook

The automatic detection and multi-class segmentation of underwater pipelines using computational models present a transformative approach for subsea pipeline monitoring. This study analysed the feasibility of using YOLOv8 and Mask R-CNN models for the detection and segmentation of pipelines and their features, even in challenging underwater environments.

The study demonstrated that the YOLOv8 segmentation small model is a highly efficient and practical model for underwater pipeline detection and segmentation, outperforming YOLOv8 segmentation large and Mask R-CNN (ResNet 50 and ResNet 101) models in both accuracy and inference speed when trained on a limited dataset. These findings emphasise the importance of selecting

| Model                 | Inference time on GPU A100 (ms) |
|-----------------------|---------------------------------|
| YOLOv8 seg small      | 1.2                             |
| YOLOv8 seg large      | 3.1                             |
| Mask R-CNN ResNet 50  | 43                              |
| Mask R-CNN ResNet 101 | 65.5                            |

Table 7: Inference time of models

the right model based on data availability and real-world deployment constraints. Experimental results indicate that data augmentation strategies improve the precision of underrepresented classes. Pipe couplings are installed at fixed intervals along the pipeline. Future research will focus on optimising the proposed YOLO models and evaluating their adaptability for accurately determining the positions of autonomous underwater vehicles (AUVs) relative to known pipe coupling locations. Mask R-CNN models will also be modified to improve the precision. //

Acknowledgement

The author would like to express his sincere gratitude to Boris Gašparović for providing the publicly available dataset and Prof. Harald Sternberg, Dr. M.I.E Wickramasinghe and Markus Kraft for valuable guidance and support throughout this study.

References

Gašparović, Boris; Jonatan Lerga; Goran Mauša; Marina Ivašić-Kos (2022): Deep Learning Approach For Objects Detection in Underwater Pipeline Images. Applied Artificial Intelligence, DOI: 10.1080/08839514.2022.2146853

Hussain, Muhammad (2024): YOLOv1 to v8: Unveiling Each Variant – A Comprehensive Review of YOLO. IEEE Access, DOI: 10.1109/ACCESS.2024.3378568

Li, Shaobo; Jianhu Zhao; Hongmei Zhang; YouPeng Zhang (2022): Automatic Detection of Pipelines from Sub-bottom

Profiler Sonar Images. IEEE Journal of Oceanic Engineering, DOI: 10.1109/JOE.2021.3107609

Martin-Abadal, Miguel; Manuel Piñar-Molina; Antoni Martorell-Torres et al. (2021): Underwater pipe and valve 3d recognition using deep learning segmentation. Journal of Marine Science and Engineering, DOI: 10.3390/jmse9010005

Preise für die ersten drei Plätze!

Jetzt bewerben für 2027



seit 2014

Student Excellence Award

Alle Infos unter:

[www.dhyg.de/index.php/de/beruf/student-excellence-award](http://www.dhyg.de/index.php/de/beruf/student-excellence-award)

