

# VIP: A Visual Insulated Pipeline Dataset for Computer Vision Tasks

**D. R. Papadam, A. Zamioudis, P. Mentesidis, E. Charalampakis  
I. Valsamara, E. Spatharis, V. Mygdalis, I. Pitas**

Department of Informatics  
Aristotle University of Thessaloniki  
Thessaloniki, Greece, 54124  
pitas@csd.auth.gr

---

## Abstract

Although computer vision is being developed across several fields, no well-established dataset currently exists for visual insulated pipeline inspection. The lack of such a dataset negatively affects research progress, but most importantly limits numerous real-world applications, especially industrial ones. To address this, we introduce, to the best of our knowledge, the first publicly available comprehensive Visual Insulated Pipeline (VIP) dataset, comprising 3,400 images across seven European locations and annotated for three computer vision tasks: (i) Pipeline region semantic segmentation, (ii) Pipeline damage semantic segmentation, and (iii) Pipeline damage object detection. Moreover, we implement state-of-the-art algorithms for these tasks on VIP and empirically demonstrate, both quantitatively and qualitatively, that there is ample room for improvement. Therefore, the proposed dataset constitutes a valuable research benchmark for the development of novel computer vision algorithms, particularly in the domain of visual insulated pipeline inspection. In addition, knowledge gained from this domain may be transferable to other domains (e.g., industrial ones), although this was not tested in this paper.

*Keywords:* Pipeline Inspection, Pipeline Dataset, Insulated Pipelines, Object Detection, Semantic Segmentation, Industrial Automation

---



## 1. Introduction

The development of automated and trustworthy pipeline inspection systems is a crucial task in today’s world for the following reasons: *(i)* Ensuring human safety, *(ii)* Avoiding environmental hazards by preventing leakages (e.g., in oil and gas pipelines), *(iii)* Reducing cost by enabling inspection without the need to shut down pipelines, and *(iv)* Allowing access to areas that are hard-to-reach by humans. In fact, according to official data from the U.S. Pipeline and Hazardous Materials Safety Administration, more than 12,508 pipeline incidents occurred from 2001 to 2020, resulting in 285 fatalities and 1181 injuries Liu and Bao (2022). These are strong motivations to accelerate the development of safe autonomous systems that can effectively inspect pipeline infrastructure. Additionally, since this task is currently performed by well-trained personnel, it constitutes a challenging problem for an autonomous system to tackle. While many efforts are being made to this direction Liu and Bao (2022); Khan et al. (2021); Huang et al. (2025); Tan et al. (2021); Yang et al. (2025); Chen (2025); Xie et al. (2025), no comprehensive dataset currently exists to cover the domain of visual insulated pipeline inspection.

From a computer vision perspective, several general-purpose datasets have enabled the advancement of object detection Tripathi et al. (2022), semantic segmentation Guo et al. (2018), and other related algorithms. For example, the PASCAL VOC dataset Everingham et al. (2010) which consists of roughly 11000 everyday images across 20 object categories (e.g., person, dog, cat, bottle), and the MS COCO dataset Lin et al. (2014) which contains approximately 328000 everyday images across 80 object categories (e.g., person, bicycle, bird, traffic light). Moreover, domain-specific datasets have been developed to address specialized computer vision tasks. A noteworthy example is the Cityscapes dataset Cordts et al. (2016), which comprises 25000

urban scene images—5000 of which are finely annotated—across 19 classes (e.g., road, sidewalk, sky, car). All these datasets can be used for training Deep Neural Network (DNN) architectures for various computer vision tasks (e.g., image classification, image semantic segmentation, object detection). However, none of them is specific for pipeline image analysis.

Motivated by the importance of autonomous pipeline inspection systems to real-world applications, as well as the value of insulated pipeline image data to the computer vision research community, we publish the VIP dataset<sup>1</sup>, which comprises 3,400 insulated pipeline images across seven European locations. The VIP dataset is annotated for three important computer vision tasks: (i) Pipeline region semantic segmentation, (ii) Pipeline damage semantic segmentation, and (iii) Pipeline damage object detection. Moreover, VIP can be used for Anomaly Detection and Localization (similarly to IPD Papadam et al. (2025)) with the existing pipeline damage semantic segmentation binary masks, as well as for image classification by converting the existing pipeline damage object detection masks to image-level damage labels (damaged/non-damaged).

The rest of the paper is structured as follows: In Section 2, we make a brief literature review, both in terms of pipe and pipeline inspection methodologies and in terms of datasets related to the one we propose. Moreover, in Section 3, we thoroughly present the VIP dataset, describing the data collection process, sharing a statistical overview of VIP, showcasing the preprocessing and data annotation procedures, as well as the final dataset splits. Furthermore, in Section 4, we perform extensive baseline evaluations for the three, previously mentioned, computer vision tasks that are addressed by VIP. Finally, in Section 5, we summarize the work done in this paper, and highlight the key contributions.

## 2. Related Work

**Production-line Pipe Inspection:** The first inspection in a pipe’s lifecycle takes place on the production line, immediately after manufacturing. In Yang et al. (2021), the authors utilize a powerful X-ray device which is able to penetrate the steel pipes. They create a custom dataset of 3,408 images, which they manually annotate, accounting for 8 types of defects: Blowhole,

---

<sup>1</sup>The VIP dataset is available at: <https://aiia.csd.auth.gr/vip-dataset/>

Undercut, Broken arc, Crack, Overlap, Slag inclusion, Lack of fusion, and Hollow bead. Moreover, the images are increased to 30,672 after 9 types of data augmentation are performed. Finally, YOLOv5 Jocher et al. (2020a) is effectively used to perform object detection. In industrial plants, pipes are assembled to form pipeline networks that are typically used to carry fluids or gases and are usually insulated (e.g., for hot liquid transportation). In the following paragraph, we discuss the task of In-Line Inspection (ILI), which refers to the internal inspection of installed pipelines, typically after a temporary shutdown to allow for a more precise inspection.

**In-line Pipeline Inspection:** In most pipeline inspection systems employed today, the interior of the pipelines is captured with the use of ILI technology, though external inspection is also viable. A recent survey on ILI Xie and Tian (2018), investigates the following well-established non-destructive testing (NDT) methods: Magnetic Flux Leakage (MFL) Chen et al. (2024), Ultrasonic Testing (UT) Siqueira et al. (2004), Electromagnetic Acoustic Transducer (EMAT) Zhang et al. (2024), and Eddy current Testing (ET) Mohamad et al. (2023). Moreover, it points out that the MFL and UT technologies are typically used for metal loss detection, while UT, EMAT, and ET are typically used to detect cracks. Notably, the authors point out that communication between the research community and the industry should be enhanced, in order to develop ILI technologies effectively and efficiently. Another very extensive survey on pipeline robots in the oil and gas industry Xu et al. (2025), also emphasizes that each NDT method has its strengths and weaknesses. Moreover, it suggests that integrating different types of data-collection sensors (e.g. MFL, UT, visual) in a single ILI system, will significantly increase automated inspection performance.

An additional approach to pipeline inspection—although it is not considered ILI in the strict sense—is internal pipeline inspection with the use of Closed-Circuit Television (CCTV) surveillance systems to capture visual data (e.g., RGB Tan et al. (2021) or infrared Yang et al. (2025)). In this case, real-time state-of-the-art (SOTA) object detection Tripathi et al. (2022); Li et al. (2023) and semantic segmentation Mo et al. (2022) algorithms are used to localize defects and measure their severity. Last but not least, in Chen et al. (2024), the authors construct and deploy an ILI robot which uses MFL. Nevertheless, they process the MFL signals visually by combining YOLOv5 Jocher et al. (2020a) with a Vision Transformer Dosovitskiy et al. (2020).

**Pipeline Inspection Datasets:** A fairly comprehensive dataset for defect detection in the production phase of steel pipelines, is publicly shared



in Yang et al. (2021). As mentioned above, the dataset is captured using a powerful X-ray device and it comprises 3408 images, which are augmented to 30672 images accounting for 8 types of defects. However, this is not a dataset for visual insulation inspection like VIP but it covers a different domain. A noteworthy dataset for visual pipeline inspection in sewerage infrastructure is Sewer-ML Haurum and Moeslund (2021). It consists of roughly 1.3 million images, while the labels account for 18 specific classes. Moreover, it can only be used for image classification, unlike VIP which can be used for object detection and semantic segmentation as well. Finally, the visual data are captured from within the pipelines, meaning that Sewer-ML also covers a different domain from that of the proposed VIP dataset.

A dataset that focuses on the same domain as VIP is the IPD dataset Papadam et al. (2025), which contains 357 images of insulated pipelines annotated with binary mask ground truths. Unlike the dataset we propose in this paper, IPD was only used for the task of anomaly detection and localization. Moreover, the PDI dataset Montesidis et al. (2024) which comprises 2190 images of insulated pipelines, also covers the same domain as VIP, but it is only annotated for the task of pipeline damage object detection. In this work, we incorporated parts of the IPD and PDI datasets into the VIP dataset, while we also added additional images in order to create a comprehensive, multi-location insulated pipeline dataset.

### 3. The VIP Dataset

In this section, the proposed VIP dataset is thoroughly presented. The main focus of this dataset is the identification of pipeline insulation damage, particularly in the metal jacketing of the pipelines. In Table 1, we present a high-level comparison of VIP with two other pipeline datasets. In each column, except for the first two, we mention a computer vision task, and the datasets that can be used for this tasks are given a checkmark (✓). This does not mean that they have been used, but that the existing annotations allow for it. For example, the VIP dataset presented in this paper, has not been used for image classification or anomaly detection and localization, but given the pipeline damage object detection masks we can very easily produce image classification labels, and we can also use the existing pipeline damage semantic segmentation masks to perform anomaly detection and localization.

The subsections that follow are structured as follows: In Section 3.1, we discuss the data collection process and also present a visualization and

overview of the VIP dataset. Moreover, in Section 3.2, we present the data preprocessing techniques we used and the distribution of VIP image dimensions. Furthermore, in Section 3.3, we thoroughly describe the annotation process and also share visualizations regarding bounding box annotations. Finally, in Section 3.4, we describe the dataset splits.

Table 1: High-level comparison of VIP with other pipeline datasets.

Datasets	Images	Image Classification	Pipeline Semantic Segmentation	Damage Semantic Segmentation	Damage Object Detection	Anomaly Detection and Localization
IPD	357	✓		✓		✓
PDI	2190	✓			✓	
VIP	3400	✓	✓	✓	✓	✓

### 3.1. VIP Data Collection and Overview

The VIP dataset was mainly collected from industrial and public facilities in Europe with cluttered environments, while pipelines from public facilities were also included. It focuses on inspection of insulated pipelines that are jacketed with metal, and it has been collected from seven distinct locations in total, which are depicted in Figure 1. To capture the data, three types of devices were used: High-end cinema camera, UAV camera, and smartphone cameras. During the data collection process, we reached out to industry experts to better understand the factors involved in the identification of insulation damages. After consultation, we decided to account for two types of visible insulation damages: (i) Hole, and (ii) Open Insulation (see Figure 5). The VIP dataset which we propose in this paper, consists of 3400 images depicting insulated pipelines. As shown in Table 2, 2135 of these images contain at least one damaged pipeline, while the remaining 1265 depict non-damaged pipelines.

### 3.2. Data Preprocessing

In order to ensure confidentiality, areas that could potentially reveal the identity of specific facilities were blurred using Gaussian noise. Moreover, to increase the scenarios where holes are present, the GIMP software The GIMP Development Team (2024) was used to add synthetic holes to 44 images in total. An example from location 1 is depicted in Figure 2. Finally, since some of the original images were very large—even above 8K resolution—all images were resized so that their large spatial dimension is 1920 and the small one

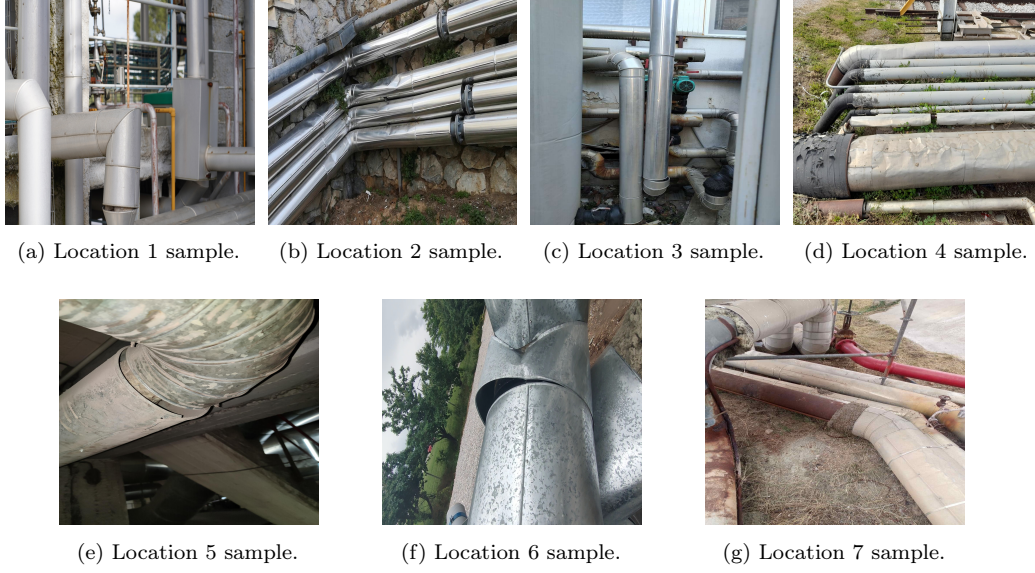


Figure 1: Image samples from the seven pipeline locations used in VIP dataset creation.

Table 2: Statistical overview of the VIP dataset.

Location	Facility	Setting	No. of Images	No. of Defective Images	No. of Holes	No. of Open Insulations
Location 1	Industrial	Outdoor	1775	852	397	1523
Location 2	Campus	Outdoor	122	116	506	243
Location 3	Healthcare	Outdoor	346	159	124	281
Location 4	Industrial	Outdoor	685	541	94	1512
Location 5	Industrial	Indoor	270	268	95	415
Location 6	Disposal	Outdoor	181	180	53	844
Location 7	Industrial	Outdoor	21	19	5	43
<b>Total:</b>			3400	2135	1274	4861

is calculated based on the aspect ratio. The image dimension distribution before and after preprocessing is depicted in Figure 3.



Figure 2: VIP synthetic holes sample from location 1.

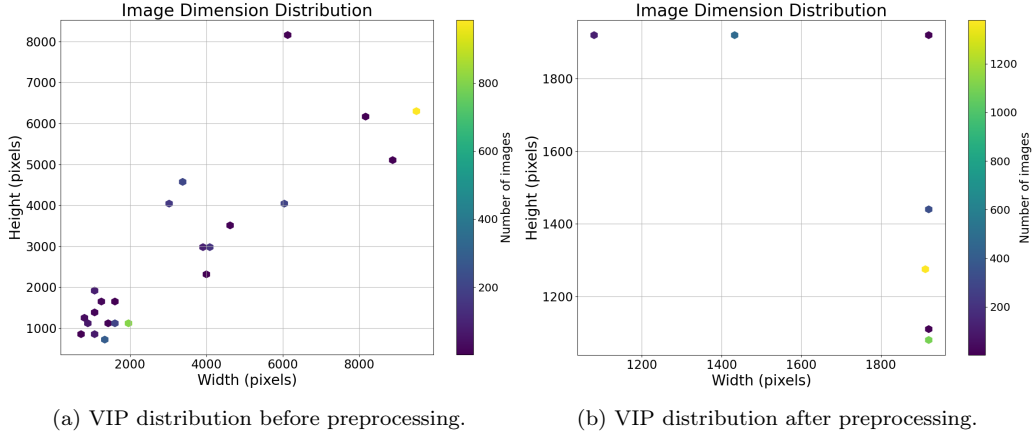


Figure 3: VIP image dimension distribution.

### 3.3. Data Annotation Procedure

To perform data annotation we made use of the Label Studio Tkachenko et al. (2019), and Labelme Wada (2021) annotation tools. We performed 3 types of annotation: 1) Pipeline region pixel-level annotation, 2) Pipeline damage pixel-level annotation, and 3) Pipeline damage bounding box annotation (hereafter “type-1”, “type-2”, and “type-3” annotations, respectively). For

all types of annotations, we only considered pipelines insulated with metal jacketing. In Subsection 3.3.1, we present the quality assurance process we followed and define the damage classes for type-3 annotations. Finally, in Section 3.3.2, we explore technical details regarding the annotation process and present visualizations of representative VIP samples.

### 3.3.1. *Quality Assurance*

The image annotation process has been performed by four of the authors. To ensure that the annotation quality is good, and that the annotations are consistent throughout the whole dataset, we conducted annotator meetings. In the first meeting, as previously mentioned, it was agreed to only consider pipelines jacketed with metal for all annotation types. Moreover, regarding type-1 annotations, it was agreed to consider any metal-jacketed pipeline that is clearly visible without the need to zoom in. Furthermore, regarding type-2 and type-3 annotations, it was agreed that small damages with respect to the image size are considered only when they are clearly visible without the need to zoom in. Particularly, for type-3 annotations, the following rules were agreed upon:

- Class definition:
  - We define a **hole** as the lack of insulation material at some spot on the insulation.
  - We define an **open insulation** as the misalignment at insulation junctions.
- When the same damage is visible in multiple spots (e.g., foreground object visually separates the damaged area into two or more parts), then one bounding box is used to cover all spots.

In subsequent annotator meetings, we discussed about specific image annotations, and came to a mutual decision of how to proceed; in other words, we made iterations of giving feedback to each other and refined the annotations several times, before achieving the final VIP ground-truth bounding boxes and masks.

### 3.3.2. *Data Annotation Details*

Damage annotations of type-2 and type-3 were performed sequentially using the Label Studio Tkachenko et al. (2019) tool, while pipeline annotations

of type-1 were performed in parallel with the other two, using both annotation tools that were previously mentioned. After type-3 bounding box damage annotation was completed for all 3400 VIP images, the Segment Anything Model (SAM) Kirillov et al. (2023) was used in combination with these bounding boxes Skalski (2024), to generate preliminary type-2 pixel-level damage masks. Moreover, due to time restrictions, 979 of those masks were manually refined using the Label Studio tool Tkachenko et al. (2019), to produce the final pixel-level damage annotations of type-2. Regarding type-1 pipeline pixel-level annotations, the Labelme tool Wada (2021) with the built-in SAM model Kirillov et al. (2023) was initially used to generate 776 preliminary pipeline masks efficiently. These masks were also refined using the Label Studio tool Tkachenko et al. (2019), to yield our final type-1 pixel-level pipeline annotations. In summary, we produced 776 type-1 annotations, 979 type-2 annotations, and 3400 type-3 annotations. Type-1 and type-2 annotations comprise binary masks (i.e., 8-bit grayscale “.png” files with zeros for background and ones for pipeline and damage, respectively). Moreover, type-3 annotations consist of “.txt” files in YOLO format (i.e., each bounding box is defined by: `<class> <x> <y> <width> <height>`).

Figure 4 illustrates a statistical overview of type-3 bounding box annotations in the VIP dataset. In particular, Figure 4a shows the distribution of the bounding box centers (normalized), indicating that the annotations are spread across the entire image space, though most of them are concentrated near the image center. Moreover, Figure 4b presents the normalized widths and heights of the bounding boxes relative to the image dimensions. It can be observed that the majority of the bounding boxes occupy less than 10% of the total image area, highlighting the small size of the target objects (i.e., holes, and open insulations) that poses a significant challenge for object detection DNN models. Type-3 annotation samples are visualized in Figure 5a (open insulation damage from location 1), and the Figure 5b (hole damage from location 5). In addition, type-1 and type-2 annotation samples are visualized in Figure 6a (type-3 pipeline annotation from location 2), and Figure 6b (type-2 damage annotation from location 4).

### 3.4. VIP Dataset Splits

With the assumption (which was later empirically validated) that the VIP dataset is a challenging benchmark for current SOTA computer vision algorithms, we simply performed a uniform random split for each annotation type. Specifically, 70%, 15%, and 15% of the VIP dataset were allocated

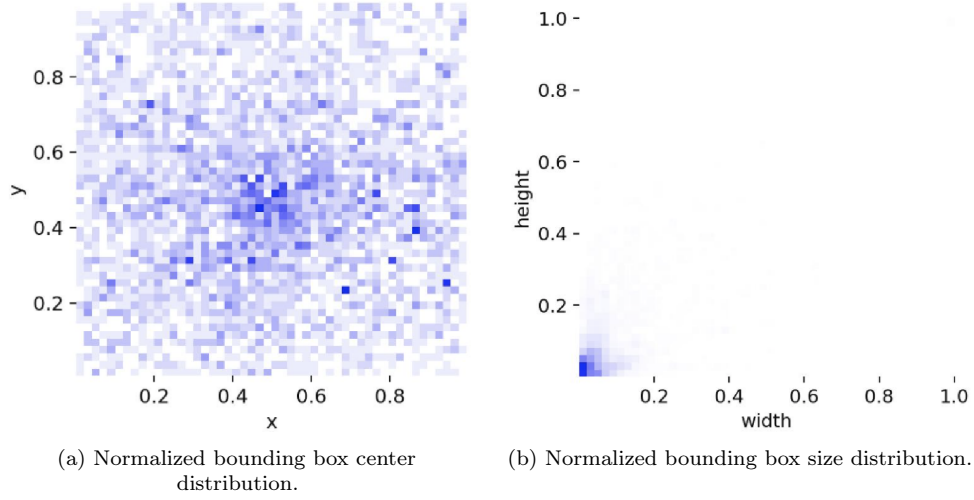


Figure 4: Statistical overview of VIP bounding box annotations relative to image dimensions.

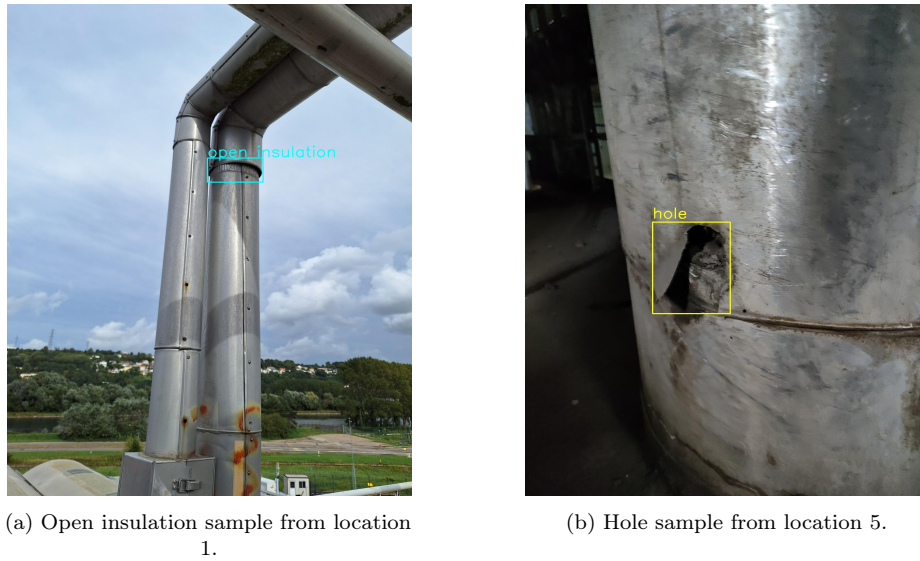


Figure 5: VIP damage bounding box annotation samples.

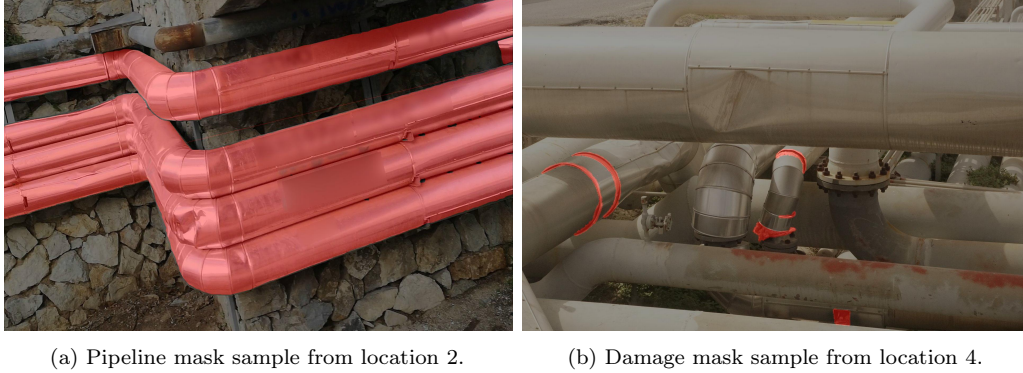


Figure 6: VIP pixel-level annotation samples.

for the training, validation, and test sets, respectively. Each annotation type corresponds to a task in Section 4; in particular, annotation type-1 corresponds to Section 4.1, annotation type-2 corresponds to Section 4.2, and annotation type-3 corresponds to Section 4.3.

#### 4. Computer vision algorithm benchmarking on the VIP dataset

In this section we use the VIP dataset as a testbed for benchmarking SOTA computer vision algorithms. Their moderate performance showcases that VIP is a valuable benchmark for further development of novel computer vision algorithms, as well as for adapting existing ones to work better on cluttered images of insulated pipelines. In Section 4.1, we address the task of insulated (with metal jacketing) pipeline region semantic segmentation both in a supervised (Section 4.1.1) and in an unsupervised (Section 4.1.2) manner. Moreover, in Section 4.2, we consider the task of pipeline damage semantic segmentation. Finally, in Section 4.3, we tackle the problem of insulated pipeline damage object detection (i.e., holes and open insulation areas), which is similar to the previous task in the sense that we detect damages, but in this case the DNN model outputs bounding boxes and not binary masks.

##### 4.1. Pipeline Region Semantic Segmentation

**Task Description.** Image semantic segmentation algorithms Guo et al. (2018) segment an image into different regions (e.g., sky, street, building, person, vehicle). Mathematically, given an RGB image  $\mathbf{I} \in \mathbb{R}^{3 \times H \times W}$  and a set of  $k$  image region classes  $\mathcal{C} = \{c_1, c_2, \dots, c_k\}$ , image semantic segmentation



can be described as a function  $f(\mathbf{I}; \boldsymbol{\theta})$ , where  $f : \mathbf{I}_{h,w} \rightarrow \mathcal{C}$  maps each pixel  $\mathbf{I}_{h,w} \in \mathbb{R}^3$  to an image region class  $c_i \in \mathcal{C}$ ,  $i \in \{1, 2, \dots, k\}$ . In this section, we perform two-class pipeline region semantic segmentation, segmenting pipeline regions (class 1) from all other image regions (class 2). Specifically, in Section 4.1.1, we perform a baseline evaluation for our task in a supervised manner, while in Section 4.1.2, we do the same in an unsupervised manner.

#### 4.1.1. Supervised Pipeline Region Semantic Segmentation

**Implementation.** For supervised pipeline region semantic segmentation, we used the “Segmentation Models PyTorch” library Iakubovskii (2019), which performs segmentation in two steps, as shown in Figure 7. Initially, the input image  $\mathbf{I} \in \mathbb{R}^{3 \times H \times W}$  goes through a DNN backbone (hereafter backbone) for feature extraction. In the next step, the extracted features go through a DNN image segmentor (hereafter segmentor), to produce the final mask  $\mathbf{C} \in \{1, 2\}^{H \times W}$ . During supervised pipeline region semantic segmentation training, the DNN model is given pairs of images and ground-truth masks, and learns appropriate weights  $\boldsymbol{\theta}$  to map image pixels  $\mathbf{I}_{h,w}$  to the proper class  $c_i \in \mathcal{C}$ ,  $i \in \{1, 2, \dots, k\}$ .

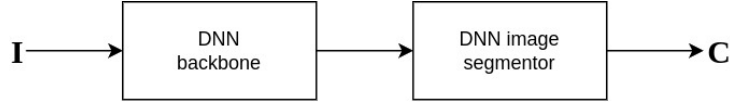


Figure 7: Image semantic segmentation DNN model.

For this task, we evaluated six segmentors: UNet++ Zhou et al. (2018), MANet Fan et al. (2020), LinkNet Chaurasia and Culurciello (2017), PSPNet Zhao et al. (2017), UPerNet Xiao et al. (2018), and Segformer Xie et al. (2021a). Furthermore, we combined each segmentor with 15 backbones: resnet152 He et al. (2016), resnext101\_32x8d Xie et al. (2017), dpn131 Chen et al. (2017), vgg19 Simonyan and Zisserman (2014), se\_resnext101\_32x4d Hu et al. (2018), densenet201 Huang et al. (2017), inceptionresnetv2 Szegedy et al. (2017), inceptionv4 Szegedy et al. (2017), efficientnet-b7 Tan and Le (2019), mobilenet\_v2 Sandler et al. (2018), xception Chollet (2017), timm-efficientnet-b7 Tan and Le (2019), timm-efficientnet-b8 Tan and Le (2019), mit\_b5 Xie et al. (2021b), and mobileone\_s4 Vasu et al. (2023). This led to a total of 88 experiments (i.e.,  $6 \times 15 - 2$  configurations that were not supported by the framework Iakubovskii (2019)). In particular, the mit\_b5 backbone Xie et al. (2021b) was incompatible with the UNet++ Zhou et al.

(2018) and LinkNet Chaurasia and Culurciello (2017) segmentors. In our setup, each image semantic segmentation DNN model—consisting of a DNN backbone and a DNN image segmentor—was trained for 70 *epochs*, with *batch\_size*  $\in \{4, 8\}$ , and initial *learning\_rate* of 0.0002 which was gradually decreased to 0.00001 by the end of the training process using a cosine learning rate annealing scheduler. In terms of hardware, for each experiment we used either an “*NVIDIA GeForce RTX 2080 Ti*” GPU or an “*NVIDIA GeForce RTX 4080 SUPER*” GPU. To evaluate model performance numerically, we utilized the most frequently used image segmentation metric: the mean Intersection over Union (mIoU).

**Numerical Benchmarking Results.** Experimental results based on the mIoU test set performance are presented in Table 3. We use **bold font** to denote the best results row-wise (i.e., the best segmentor for a given backbone), while underlining highlights the best results column-wise (i.e., the best backbone for a given segmentor). Furthermore, cells marked with “-” denote a combination of backbone and segmentor that is not supported by the used “Segmentation Models PyTorch” library Iakubovskii (2019). We observe that among the tested segmentors, UNet++ Zhou et al. (2018), UPerNet Xiao et al. (2018), and LinkNet Chaurasia and Culurciello (2017), achieve the best performance across 12, 2, and 1 out of 15 backbones, respectively. Moreover, among the tested backbones, mit\_b5 achieves the best performance across 3 out of 6 segmentors even though it is incompatible with 2 of them (i.e., UNet++ Zhou et al. (2018), and LinkNet Chaurasia and Culurciello (2017)). Each of the resnext101\_32x8d Xie et al. (2017), vgg19 Simonyan and Zisserman (2014), se\_resnext101\_32x4d Hu et al. (2018), and mobileone\_s4 Vasu et al. (2023) achieves the best performance in 1 out of 6 segmentors. Furthermore, it is noteworthy that the resnext101\_32x8d Xie et al. (2017) which introduces the concept of cardinality, does not consistently outperform resnet152 He et al. (2016) like in does in Section 4.2. Additionally, se\_resnext101\_32x4d Hu et al. (2018) which introduces the “Squeeze-and-Excitation” (SE) block to calibrate features channel-wise, does not consistently outperform resnext101\_32x8d Xie et al. (2017) like in Section 4.2. However, it consistently matches or outperforms the standard resnet152 He et al. (2016), even though it uses significantly fewer parameters. In addition, Table 4 presents the 10 best models based on the test set mIoU performance metric. We observe that although MANet Fan et al. (2020) is not in the previously mentioned 3 best segmentors, it does make it to the top 10 when combined with the mit\_b5 Xie et al. (2021b) backbone. Conversely, even though efficientnet-b7 Tan and Le (2019), timm-

efficientnet-b7 Tan and Le (2019), and resnet152 He et al. (2016) are not in the previously mentioned best backbones, they make it to the top 10 best performing models when combined with the UNet++ Zhou et al. (2018) segmentor. Finally, it is noteworthy that even though mobileone\_s4 Vasu et al. (2023) is a lightweight network with a relatively small number of parameters that was designed for mobile devices and efficient deployment, it achieves the best overall performance when combined with the UNet++ Zhou et al. (2018) segmentor.

Table 3: mIoU performance of pipeline region semantic segmentation models on VIP.

Segmentor	UNet++	MANet	LinkNet	PSPNet	UPerNet	Segformer
Backbone (parameters)						
resnet152 (58M)	<b>88.3%</b>	80.1%	86.1%	79.9%	85.7%	85.1%
resnext101_32x8d (86M)	<b>89.3%</b>	75.5%	87.9%	81.1%	86.3%	86.1%
dpn131 (76M)	87.6%	71.6%	85.3%	82.9%	<b>87.9%</b>	87.0%
vgg19 (20M)	<b>88.1%</b>	87.1%	87.1%	86.0%	87.8%	87.6%
se_resnext101_32x4d (46M)	88.3%	87.4%	<b>88.7%</b>	81.5%	87.3%	87.7%
densenet201 (18M)	<b>87.8%</b>	76.3%	86.7%	80.7%	86.3%	85.8%
inceptionresnetv2 (54M)	<b>86.4%</b>	81.9%	85.7%	71.9%	83.6%	82.6%
inceptionv4 (41M)	<b>88.0%</b>	86.5%	86.5%	81.2%	84.9%	86.0%
efficientnet-b7 (63M)	<b>89.1%</b>	88.2%	87.7%	80.8%	86.8%	86.4%
mobilenet_v2 (2M)	<b>85.4%</b>	84.3%	81.3%	68.0%	82.0%	82.4%
xception (20M)	<b>88.0%</b>	86.6%	85.2%	69.6%	82.6%	83.5%
timm-efficientnet-b7 (63M)	<b>88.5%</b>	87.1%	87.2%	79.5%	85.9%	86.6%
timm-efficientnet-b8 (84M)	<b>87.0%</b>	86.5%	85.6%	79.7%	85.0%	83.8%
mit_b5 (81M)	-	88.5%	-	81.0%	<b>88.7%</b>	<u>87.8%</u>
mobileone_s4 (12M)	<b>89.3%</b>	85.8%	87.2%	79.4%	85.4%	86.2%

Detailed results are logged in: [https://wandb.ai/auth\\_cvml/VIP-Pipeline\\_Semantic\\_Segmentation](https://wandb.ai/auth_cvml/VIP-Pipeline_Semantic_Segmentation).

Table 4: Top 10 pipeline region semantic segmentation models on VIP.

Model (Segmentor / Backbone)	mIoU
UNet++ / mobileone_s4	89.3%
UNet++ / resnext101_32x8d	89.3%
UNet++ / efficientnet-b7	89.1%
UPerNet / mit_b5	88.7%
LinkNet / se_resnext101_32x4d	88.7%
MANet / mit_b5	88.5%
UNet++ / timm-efficientnet-b7	88.5%
UNet++ / resnet152	88.3%
UNet++ / se_resnext101_32x4d	88.3%

**Visual Benchmarking Results.** Visualizations of inference results using the best model (i.e., UNet++ / mobileone\_s4) are depicted in Figure 8.

One sample image was chosen from each of the seven locations. It can be seen that the predictions are near-perfect in the samples of locations 2 and 6 (Figures 8b, 8f), while there is more room for improvement on the other location samples (Figures 8a, 8c, 8d, 8e, 8g). Nevertheless, the visual results verify that the task of supervised pipeline region semantic segmentation is easier than the task of supervised pipeline damage semantic segmentation (see Figure 10), which is explored in Section 4.2.

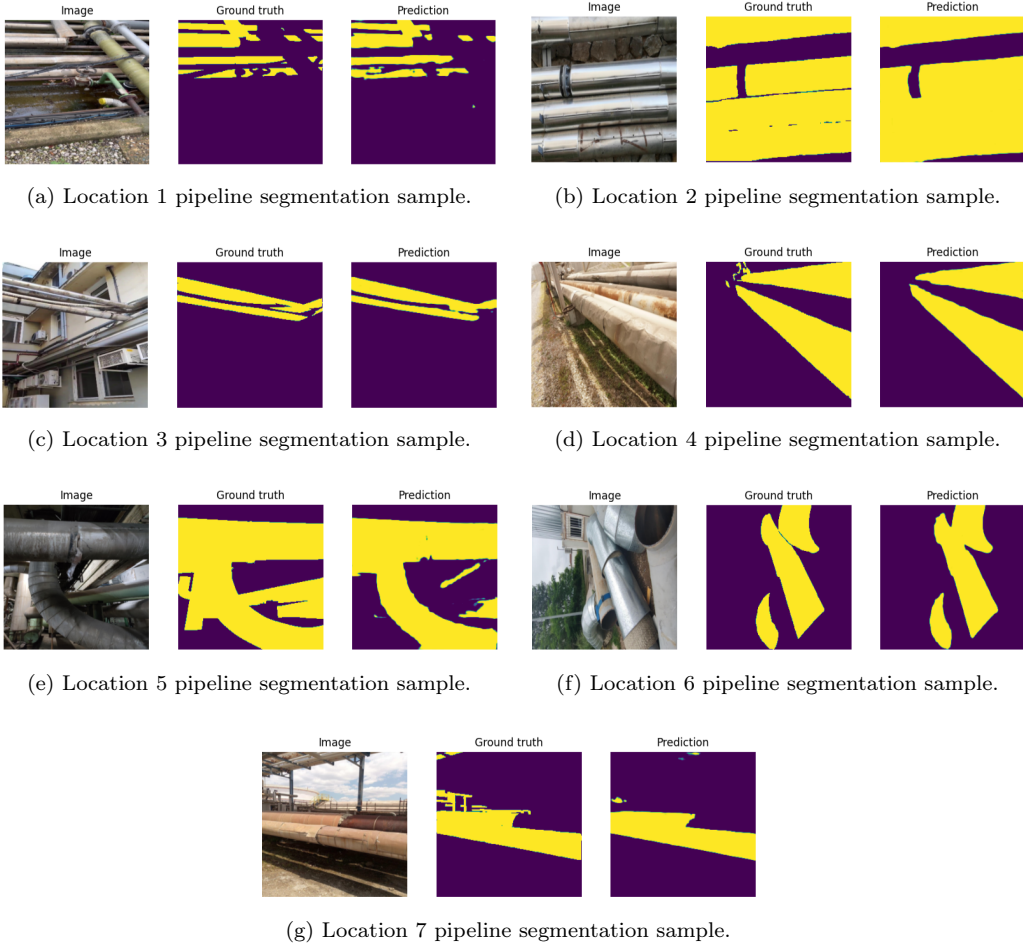


Figure 8: Supervised pipeline region semantic segmentation visual results on VIP samples.  
The best-performing model was used: UNet++ / mobileone\_s4.

#### 4.1.2. Unsupervised Pipeline Region Semantic Segmentation

**Task Description.** Manual annotation for image semantic segmentation is a costly and time-intensive process, especially in industrial inspection scenarios where expert knowledge is required. Moreover, scaling to large datasets often raises privacy concerns when data must be shared with external annotators. Unsupervised Semantic Segmentation (USS) has emerged as a promising direction to alleviate these limitations. Recent USS methods leverage representation learning techniques on pretrained DNN backbones Dosovitskiy et al. (2021); Caron et al. (2021), followed by clustering techniques to produce semantically coherent pixel clusters in the feature space, without relying on human-provided labels.

**Implementation.** In this work, we evaluate two SOTA USS methods, namely STEGO Hamilton et al. (2022) and EAGLE Kim et al. (2024), on our proposed VIP dataset. Both approaches leverage a pre-trained DINO backbone Caron et al. (2021), which processes input images patches to produce per-patch feature representations. These methods subsequently project the high-dimensional DINO feature space into a lower-dimensional embedding, promoting similarity among features corresponding to semantically coherent image regions. EAGLE improves upon STEGO by incorporating an eigen-analysis module, which facilitates more robust clustering of semantically consistent object-level information. This additional component enables EAGLE to mitigate the incorrect cluster assignment of pixels belonging to the same object class, even when these pixels exhibit local appearance variations—for instance, ensuring that features corresponding to parts such as a car door or wheel are correctly grouped with those of the full car object. Since a binary semantic segmentation task (foreground metal-jacketed pipeline vs. background) is addressed in this paper, we adapted both USS methods to segment each image into two semantic clusters. Furthermore, we adapted the training procedure of both methods to our VIP dataset by removing the use of negative samples in their contrastive learning loss functions. While contrastive learning with negatives is a widely adopted technique in self-supervised representation learning Ericsson et al. (2022), it is not applicable in industrial pipeline datasets like VIP, where the selection of negative images is inherently ambiguous (all images depict pipelines). Finally, to measure the performance of our implemented approaches we used two well-established metrics in the domain of image semantic segmentation: mean Intersection over Union (mIoU), and accuracy.

**Numerical Benchmarking Results.** The USS results presented in Table 5. We use **bold font** to denote the best result per metric, while we highlight the second-best result per metric with underlining. The performance metrics indicate that employing the base Visual Transformer (ViT) Dosovitskiy et al. (2020) variant of DINO, as opposed to the small ViT variant, leads to a substantial improvement in segmentation accuracy for both USS methods on the VIP dataset. Moreover, it is observed that the EAGLE eigen-analysis module consistently improves the accuracy but not the mIoU, while it introduces additional computational overhead during both training and inference. In contrast, STEGO achieves comparable segmentation performance with a considerably simpler implementation. By comparing the entries of Tables 4 and 5, it can be argued that supervised pipeline region semantic segmentation methods perform significantly better than unsupervised ones, at the expense of a much more labor-intensive annotation process.

**Visual Benchmarking Results.** Figure 9 visualizes VIP image segmentation results using EAGLE / vit-base8, which is the best-performing model. Our qualitative analysis indicates that USS segmentation performance highly depends on the visual distinction between the metal-jacketed pipeline and background regions. In images where the foreground and background exhibit clear visual separation (Figures 9e, 9f, 9g), the USS models produce accurate segmentation masks without any human supervision. Conversely, in cases where the background contains clutter or textures that are visually similar to the metal-jacketed pipeline structure (Figures 9a, 9c, 9d), the model yields increased false positive. The same applies in cases where the texture of the metal jacketing varies inside a single image (Figure 9b). Consequently, VIP constitutes a strong testbed for future USS research in pipeline inspection, supporting the direct application of SOTA USS methods.

Table 5: Performance of unsupervised pipeline region semantic segmentation on VIP.

Model (Segmentor / Backbone)	mIoU	Accuracy
STEGO / vit-small16	62.50%	77.34%
STEGO / vit-small8	58.80%	74.56%
STEGO / vit-base16	<u>70.04%</u>	82.60%
STEGO / vit-base8	65.08%	79.20%
EAGLE / vit-small16	67.72%	80.91%
EAGLE / vit-small8	60.88%	76.04%
EAGLE / vit-base16	55.05%	<b>83.97%</b>
EAGLE / vit-base8	<b>70.06%</b>	<u>82.61%</u>

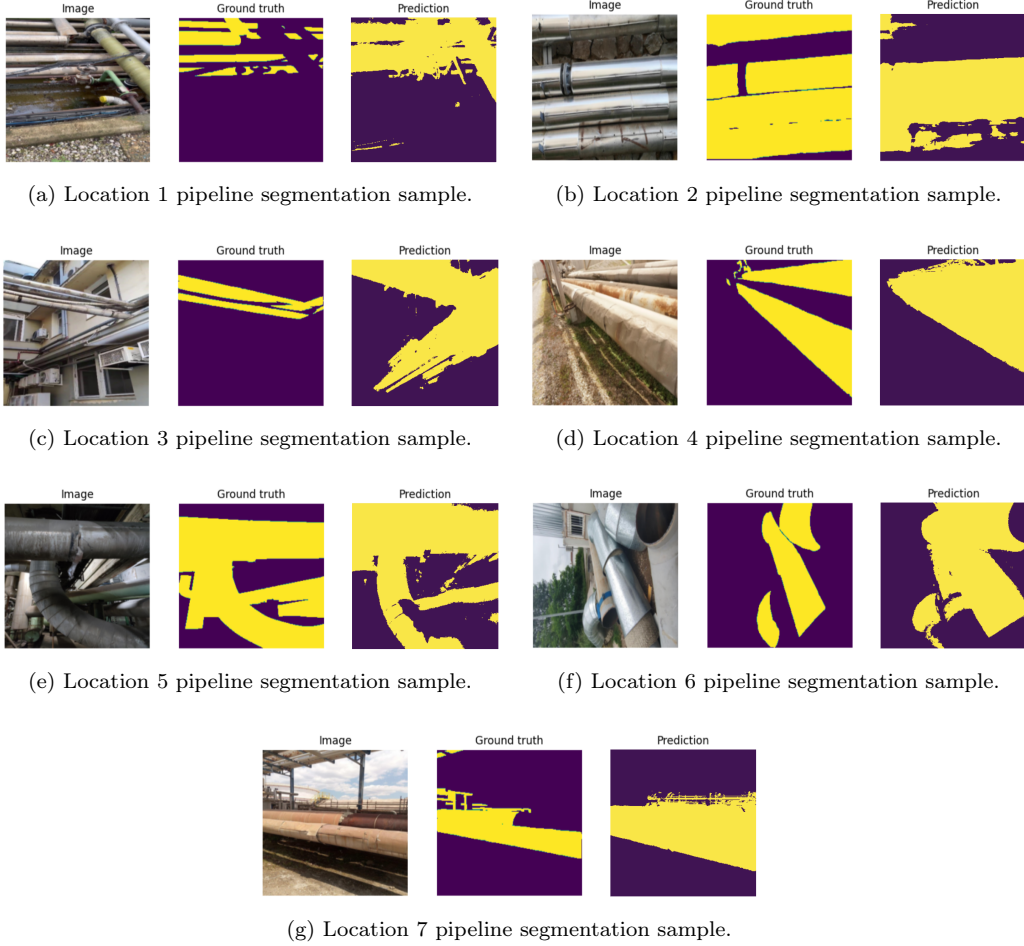


Figure 9: Unsupervised pipeline region semantic segmentation visual results on VIP samples.  
The best-performing model was used: EAGLE / vit-base8.

#### 4.2. Pipeline Damage Semantic Segmentation

**Task Description.** The task of image semantic segmentation Guo et al. (2018) was outlined in Section 4.1. In this section, we perform two-class image semantic segmentation, aiming to separate damages in pipeline insulation (class 1) from all other objects (class 2).

**Implementation.** For (supervised) pipeline damage semantic segmentation, we used the “Segmentation Models PyTorch” library Iakubovskii (2019), as in Section 4.1.1, using the same training parameters and hardware. Moreover, to evaluate the model performance numerically, we used the mean Intersection over Union (mIoU) metric, which is also consistent with Section 4.1.1.

**Numerical Benchmarking Results.** In Table 6, we report the mIoU score of each created model, calculated on the test set. With **bold font**, we highlight the row-wise highest mIoU scores (i.e., the best segmentors for each backbone). Similarly, with underlining, we denote the column-wise highest mIoU scores (i.e., the best backbones for each segmentor). Moreover, cells marked with “-” correspond to a combination of backbone and segmentor not supported by the used framework Iakubovskii (2019). We notice that the UNet++Zhou et al. (2018) segmentor performs the best across 12 out of 15 backbones, while the UPerNet Xiao et al. (2018) segmentor performs the best across the rest 3 out of 15 backbones. Moreover, the efficientnet-b7 Tan and Le (2019) backbone performs the best across 3 out of 6 segmentors; se\_resnext1-1\_32x4d Xie et al. (2017) performs the best across 2 out of 6 segmentors; vgg19 Simonyan and Zisserman (2014) performs the best across 1 out of 6 segmentors. Furthermore, it is noteworthy that resnext101\_32x8d Xie et al. (2017) which introduces the concept of cardinality, consistently outperforms resnet152 He et al. (2016). Additionally, se\_resnext101\_32x4d Hu et al. (2018) which introduces the “Squeeze-and-Excitation” (SE) block to calibrate features channel-wise, consistently outperforms resnext101\_32x8d Xie et al. (2017), even though it uses nearly half of the parameters. Finally, Table 7 depicts the 10 best-performing models based on the mIoU metric, calculated on the test set. It can be seen that the best segmentors previously identified (i.e., UNet++Zhou et al. (2018) and UPerNet Xiao et al. (2018)) achieve a spot in the top 10 even when they are combined with backbones that are not the best ones for this task (e.g., inceptionv4 Szegedy et al. (2017), mobileone\_s4 Vasu et al. (2023)). Conversely, the best backbones previously identified (i.e., efficientnet-b7 Tan and Le (2019), se\_resnext1-1\_32x4d Xie et al. (2017), and vgg19 Simonyan and Zisserman (2014)) make it to the



top 10 when combined with model architectures that are not the best ones for this task (e.g., LinkNet Chaurasia and Culurciello (2017), MANet Fan et al. (2020)). By comparing entries of Tables 4 and 7, it can be argued that pipeline region semantic segmentation, on the VIP dataset, is a much easier task than pipeline damage semantic segmentation. This is natural, as pipeline images contain damages that occupy a relatively small area on the image plane. In addition, it is easier to visually recognize an insulated pipeline than to localize a spot in which the pipeline insulation is damaged.

Table 6: mIoU performance of pipeline damage semantic segmentation models on VIP.

Segmentor	UNet++	MANet	LinkNet	PSPNet	UPerNet	Segformer
Backbone (parameters)						
resnet152 (58M)	<b>50.9%</b>	47.2%	43.0%	42.3%	50.5%	46.9%
resnext101_32x8d (86M)	52.5%	48.2%	50.9%	42.7%	<b>52.7%</b>	51.0%
dpn131 (76M)	48.8%	39.4%	49.6%	46.0%	<b>51.4%</b>	48.8%
vgg19 (20M)	<b>54.7%</b>	53.9%	54.6%	49.3%	51.6%	54.1%
se_resnext101_32x4d (46M)	<b>58.5%</b>	53.4%	56.9%	45.8%	55.2%	55.5%
densenet201 (18M)	<b>54.6%</b>	46.4%	51.2%	46.1%	53.0%	52.8%
inceptionresnetv2 (54M)	<b>54.2%</b>	52.8%	52.4%	43.3%	49.3%	47.9%
inceptionv4 (41M)	<b>57.3%</b>	53.6%	53.4%	45.1%	53.3%	51.9%
efficientnet-b7 (63M)	<b>57.6%</b>	56.3%	55.7%	46.1%	56.3%	56.9%
mobilenet_v2 (2M)	<b>53.1%</b>	50.7%	47.8%	36.0%	51.6%	49.7%
xception (20M)	<b>56.1%</b>	53.0%	51.8%	34.5%	53.7%	51.3%
timm-efficientnet-b7 (63M)	<b>57.7%</b>	53.9%	54.8%	45.0%	55.5%	54.8%
timm-efficientnet-b8 (84M)	<b>55.9%</b>	54.6%	54.1%	45.6%	53.5%	54.1%
mit_b5 (81M)	-	50.4%	-	38.9%	<b>53.9%</b>	53.5%
mobileone_s4 (12M)	<b>57.5%</b>	51.5%	52.3%	45.8%	52.0%	53.4%

Detailed results are logged in: [https://wandb.ai/auth\\_cvml/VIP-Damage\\_Semantic\\_Segmentation](https://wandb.ai/auth_cvml/VIP-Damage_Semantic_Segmentation).

Table 7: Top 10 pipeline damage semantic segmentation models on VIP.

Model (Segmentor / Backbone)	mIoU
UNet++ / se_resnext101_32x4d	58.5%
UNet++ / timm-efficientnet-b7	57.7%
UNet++ / efficientnet-b7	57.6%
UNet++ / mobileone_s4	57.5%
UNet++ / inceptionv4	57.3%
Segformer / efficientnet-b7	56.9%
LinkNet / se_resnext101_32x4d	56.9%
MANet / efficientnet-b7	56.3%
UPerNet / efficientnet-b7	56.3%
UNet++ / xception	56.1%

**Visual Benchmarking Results.** In Figure 10, the best-performing model (i.e., UNet++ / se\_resnext101\_32x4d) is used for inference on sam-

ples of the test set, and the produced visualizations are presented. Since there are no damage mask annotations for location 2, a sample is illustrated from six out of seven locations (i.e., locations 1, 3, 4, 5, 6, 7). In Figures 10a, 10c, 10d, 10e, and 10f, damaged samples are depicted from locations 1, 4, 5, 6, and 7, respectively, while in Figure 10b, a non-damaged sample is presented from location 3. We show samples where the best baseline model performs remarkably well (Figures 10a, 10b), samples where the model achieves solid performance (Figures 10c, 10e), as well as samples where there is ample room for improvement (Figures 10d, 10f).

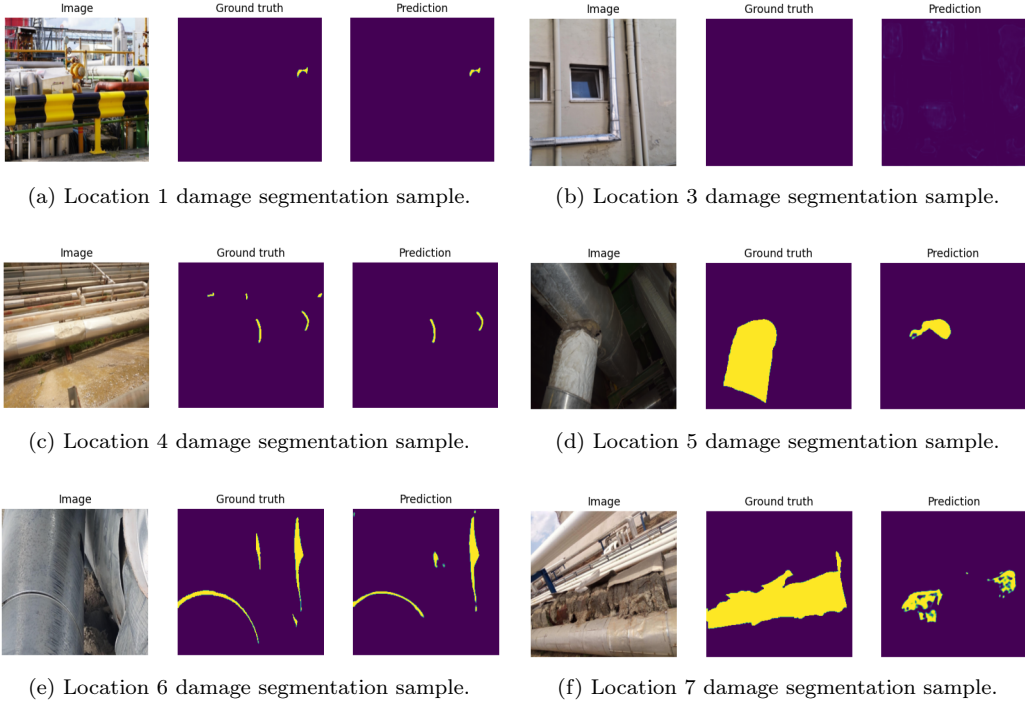


Figure 10: Supervised pipeline damage semantic segmentation visual results on VIP samples.  
The best-performing model was used: UNet++ / se\_resnext101\_32x4d.

#### 4.3. Pipeline Damage Object Detection

**Task Description.** Object detection is a fundamental computer vision task that simultaneously performs classification and localization by identifying multiple objects and regressing their corresponding bounding boxes within an image. In mathematical terms, we are given an RGB image  $\mathbf{I} \in \mathbb{R}^{3 \times H \times W}$ ,

and a target of  $K$  objects to detect for that image:  $\mathbf{Y}_{\mathbf{I}} \in \mathbb{R}^{K \times 5}$ . Each object instance is defined by:  $\mathbf{Y}_{\mathbf{I},k} = [x_k, y_k, w_k, h_k, c_k]$ , where:  $k \in \{1, 2, \dots, K\}$ ;  $x_k, y_k$  (center coordinates of the bounding box);  $w_k, h_k$  (width and height of bounding box);  $c_k \in \{0, 1\}$  is the object class. Specifically,  $c_k = 0, 1$  corresponds to a hole and open insulation, respectively.

The object detection model learns a function  $f(\mathbf{I}; \boldsymbol{\theta})$ , where  $f : \mathbf{I} \rightarrow \hat{\mathbf{Y}}$ . The goal is to approximate the ground-truth  $\mathbf{Y}$ , i.e.,  $\hat{\mathbf{Y}} \approx \mathbf{Y}$ . Typically, as in the first version of You Only Look Once (YOLO) algorithm Redmon et al. (2016), the DNN model is trained in a supervised manner, by optimizing a composite loss function:

$$\mathcal{L} = \lambda_{\text{cls}} \mathcal{L}_{\text{cls}} + \lambda_{\text{loc}} \mathcal{L}_{\text{loc}}, \quad (1)$$

where  $\mathcal{L}_{\text{cls}}$  is the classification loss and  $\mathcal{L}_{\text{loc}}$  is object localization regression loss. The hyperparameters  $\lambda_{\text{cls}}$  and  $\lambda_{\text{loc}}$  balance the two objectives. Nevertheless, in later versions of YOLO (e.g., YOLOv4 Bochkovskiy et al. (2020)), the loss function was extended to include more terms, further improving the performance of the YOLO family of algorithms. The YOLO-based object detectors Tripathi et al. (2022) have surpassed two-stage detectors Girshick et al. (2014); Girshick (2015); Ren et al. (2015), due to their much higher inference speed without significant accuracy loss. This makes them highly suitable for various industrial applications.

**Implementation.** In this section, we evaluate SOTA YOLO object detection models for the task of pipeline damage object detection on our proposed VIP dataset. All models have been pretrained on the COCO dataset and subsequently fine-tuned and tested on VIP. Training was performed for 200 *epochs* with *batch\_size* = 16, using two “*NVIDIA GeForce RTX 4080 SUPER*” GPUs. We used the Ultralytics framework Jocher et al. (2023) for implementation, keeping all hyperparameters at their default values to establish baseline performance. We used the standard evaluation metrics for object detection Padilla et al. (2020), namely mean Average Precision (mAP), Precision, and Recall.

**Numerical Benchmarking Results.** Table 8 presents a comprehensive evaluation of various YOLO architectures on the VIP dataset, reporting precision, recall, mean Average Precision at IoU threshold 0.5 (mAP50), and the averaged mAP across multiple IoU thresholds (mAP50-95) for both the validation and test sets. The reported values regarding the validation set correspond to the maximum performance (for each metric) over the 200 *epochs*

of training, while the reported values regarding the test set are calculated by using the weights that yielded the maximum mAP50-95 performance on the validation set over the 200 *epochs* of training. We use **bold font** to denote the highest value per metric, while with underlining we denote the second-highest value per metric. Across all models, we observe a consistent trade-off between model size (i.e., number of parameters) and inference time. Moreover, the performance usually improves as YOLO models increase in size, but this is not always the case. The YOLOv8 Jocher et al. (2023) series achieves the highest overall mAP scores, with the yolov8s Jocher et al. (2023) model attaining the best balance between precision (0.832 on test) and mAP50-95 (0.418 on test), while maintaining a fast inference time of 0.874 ms. Larger models such as yolov8l Jocher et al. (2023) demonstrate improved recall (0.637) and competitive mAP but incur increased inference latency (3.011 ms). The yolov9c Wang et al. (2024) model also performs competitively, particularly excelling in recall (0.622) and mAP50 (0.671) on the test set, highlighting its efficacy, despite a moderate parameter count. Notably, smaller models like yolov5n Jocher et al. (2020b) and yolov10n Jocher et al. (2023) offer lower computational costs with inference times below 1 ms but at the expense of reduced accuracy metrics. This analysis confirms that mid-sized models from the YOLOv8 Jocher et al. (2023) and YOLOv9 Wang et al. (2024) families are optimal for real-time pipeline damage object detection in industrial pipeline metal-jacketed insulation, balancing accuracy and computational efficiency effectively. As a final remark, compared to other object detection applications (e.g., pedestrian and vehicle detection Liu et al. (2025)), Table 8 illustrates lower YOLO performance on the VIP dataset, indicating that pipeline damage object detection may be a more challenging task. Therefore, the proposed dataset is a valuable resource for designing better pipeline damage object detectors.

**Visual Benchmarking Results.** Figure 11, illustrates yolov8l’s object detection results on two VIP test images, demonstrating its ability to localize and classify damage instances under various conditions. Figure 11a depicts a misclassification of a gauge as an open insulation, illustrating the challenge that the model faces when background cluster is present in the scene. In contrast, Figure 11b depicts a pipeline section with minimal background clutter, where the yolov8l model performs better. These observations highlight the main VIP dataset feature: the complex and cluttered nature of outdoor industrial environments.

Table 8: Performance of pipeline damage object detection algorithms.

Architecture (parameters)	Precision (val   test)		Recall (val   test)		mAP50 (val   test)		mAP50-95 (val   test)		Inference Time (ms)
yolov5n (2.5M)	0.806	0.798	0.560	0.569	0.604	0.636	0.331	0.370	1.011
yolov5s (9.1M)	<u>0.846</u>	0.829	0.587	0.576	0.630	0.662	0.361	0.388	1.478
yolov5m (25.1M)	0.822	0.752	0.607	0.621	0.638	0.654	0.366	0.406	1.963
yolov5l (53.2M)	0.845	0.806	0.613	0.599	0.633	0.659	0.364	0.401	2.711
yolov8n (3M)	0.786	0.735	0.552	0.579	0.603	0.612	0.331	0.359	1.207
yolov8s (11.1M)	<b>0.857</b>	<u>0.832</u>	0.609	0.605	0.651	<b>0.676</b>	0.374	<b>0.418</b>	0.874
yolov8m (25.9M)	0.809	0.811	<u>0.640</u>	0.596	0.655	0.665	0.380	<u>0.408</u>	1.722
yolov8l (43.6M)	0.844	0.765	<b>0.641</b>	<b>0.637</b>	<b>0.658</b>	0.668	0.384	<b>0.418</b>	3.011
yolov9t (2M)	0.811	0.795	0.606	0.561	0.628	0.628	0.365	0.368	1.506
yolov9s (7.3M)	0.820	0.823	0.583	0.601	0.629	0.665	0.356	0.407	1.578
yolov9m (20.2M)	0.824	0.805	0.596	0.589	0.631	0.653	0.364	0.391	2.365
yolov9c (25.5M)	0.836	0.818	0.623	<u>0.622</u>	0.652	<u>0.671</u>	0.377	<b>0.418</b>	2.613
yolov10n (2.7M)	0.779	0.805	0.546	0.535	0.573	0.601	0.312	0.342	<u>0.823</u>
yolov10s (8.1M)	0.793	0.794	0.585	0.566	0.615	0.626	0.350	0.377	1.062
yolov10m (16.5M)	0.819	<b>0.836</b>	0.590	0.576	0.629	0.638	0.363	0.384	2.244
yolov10l (25.8M)	0.830	0.775	0.589	0.601	0.637	0.645	0.375	0.405	5.012
yolo11n (2.6M)	0.821	0.790	0.559	0.562	0.600	0.609	0.333	0.367	<b>0.775</b>
yolo11s (9.4M)	0.841	0.790	0.620	0.592	0.645	0.639	0.388	0.390	1.615
yolo11m (20.1M)	0.830	0.790	0.621	0.600	<u>0.656</u>	0.655	<b>0.399</b>	0.399	1.971
yolo11l (25.3M)	0.843	0.813	0.589	0.563	0.639	0.633	<u>0.389</u>	0.390	2.564

Detailed results are logged in: [https://wandb.ai/auth\\_cvml/VIP-Damage\\_Detection](https://wandb.ai/auth_cvml/VIP-Damage_Detection).

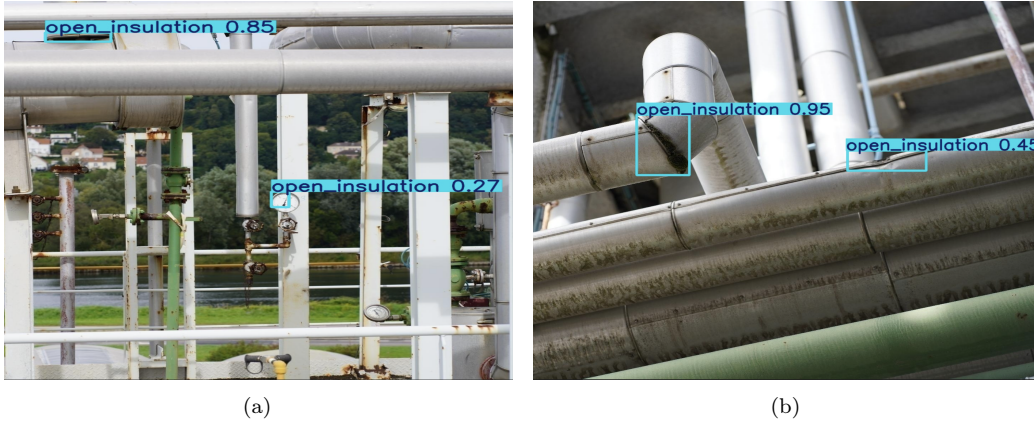


Figure 11: yolov8l pipeline damage object detection on VIP test set samples from location 1.

## 5. Conclusions

In this paper, we introduced the VIP dataset for visual insulated pipeline inspection in cluttered environments. It consists of 3,400 images, while it is currently annotated for three computer vision tasks: (i) Pipeline region semantic segmentation (with binary pipeline masks), (ii) Pipeline damage semantic segmentation (with binary damage masks), and (iii) Pipeline damage object detection (with bounding boxes). Moreover, in the first task, given that the supervised SOTA approaches perform fairly well, achieving nearly 90% mIoU, we also performed a baseline evaluation using unsupervised approaches (i.e., STEGO Hamilton et al. (2022) and EAGLE Kim et al. (2024)). In the second and third tasks, as well as the unsupervised evaluation in the first task, several SOTA algorithms perform quite poorly—as shown both numerically and visually. This demonstrates that VIP is of great significance not only for real-world industrial pipeline applications, but also for the advancement of novel computer vision algorithms in general. Finally, we emphasize the importance of developing more datasets like VIP.

## Acknowledgments

This work has received funding from the European Union’s Horizon Europe research and innovation programme under grant agreement number 101070604 (SIMAR). This publication reflects only the authors’ views. The European Commission is not responsible for any use that may be made of the information it contains.

## References

- Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M., 2020. YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 .
- Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A., 2021. Emerging properties in self-supervised vision transformers, in: Proceedings of the IEEE/CVF international conference on computer vision, pp. 9650–9660.
- Chaurasia, A., Culurciello, E., 2017. Linknet: Exploiting encoder representations for efficient semantic segmentation, in: 2017 IEEE Visual Communications and Image Processing (VCIP), pp. 1–4. doi:10.1109/VCIP.2017.8305148.

- Chen, P., 2025. Advancements and future outlook of safety monitoring, inspection and assessment technologies for oil and gas pipeline networks. *Journal of Pipeline Science and Engineering* , 100267doi:<https://doi.org/10.1016/j.jpse.2025.100267>.
- Chen, P., Li, R., Fu, K., Zhong, Z., Xie, J., Wang, J., Zhu, J., 2024. A cascaded deep learning approach for detecting pipeline defects via pretrained yolov5 and vit models based on mfl data. *Mechanical Systems and Signal Processing* 206, 110919. doi:<https://doi.org/10.1016/j.ymssp.2023.110919>.
- Chen, Y., Li, J., Xiao, H., Jin, X., Yan, S., Feng, J., 2017. Dual path networks, in: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), *Advances in Neural Information Processing Systems*, Curran Associates, Inc.
- Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* .
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2021. An image is worth 16x16 words: Transformers for image recognition at scale, in: *International Conference on Learning Representations*.
- Ericsson, L., Gouk, H., Loy, C.C., Hospedales, T.M., 2022. Self-supervised representation learning: Introduction, advances, and challenges. *IEEE Signal Processing Magazine* 39, 42–62.
- Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A., 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision* 88, 303–338.

- Fan, T., Wang, G., Li, Y., Wang, H., 2020. Ma-net: A multi-scale attention network for liver and tumor segmentation. *IEEE Access* 8, 179656–179665. doi:10.1109/ACCESS.2020.3025372.
- Girshick, R., 2015. Fast r-cnn, in: *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448.
- Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587.
- Guo, Y., Liu, Y., Georgiou, T., Lew, M.S., 2018. A review of semantic segmentation using deep neural networks. *International journal of multimedia information retrieval* 7, 87–93.
- Hamilton, M., Zhang, Z., Hariharan, B., Snavely, N., Freeman, W.T., 2022. Unsupervised semantic segmentation by distilling feature correspondences, in: *International Conference on Learning Representations*.
- Haurum, J.B., Moeslund, T.B., 2021. Sewer-ml: A multi-label sewer defect classification dataset and benchmark, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13456–13467.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Hu, J., Shen, L., Sun, G., 2018. Squeeze-and-excitation networks, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Huang, J., Li, R., Li, Y., Fu, K., Chen, P., 2025. Development and simulation analysis of a novel flexible deformation inspection units (fdiu) for oil and gas pipelines. *Journal of Pipeline Science and Engineering* , 100290doi:<https://doi.org/10.1016/j.jpse.2025.100290>.



- Iakubovskii, P., 2019. Segmentation models pytorch. URL: [https://github.com/qubvel/segmentation\\_models.pytorch](https://github.com/qubvel/segmentation_models.pytorch).
- Jocher, G., Qiu, J., Chaurasia, A., 2023. Ultralytics yolo. URL: <https://ultralytics.com>.
- Jocher, G., Stoken, A., Borovec, J., Changyu, L., Hogan, A., Diaconu, L., Poznanski, J., Yu, L., Rai, P., Ferriday, R., et al., 2020a. ultralytics/yolov5: v3. 0. Zenodo .
- Jocher, G., Stoken, A., Borovec, J., Changyu, L., Hogan, A., Diaconu, L., Poznanski, J., Yu, L., Rai, P., Ferriday, R., et al., 2020b. ultralytics/yolov5: v3. 0. Zenodo .
- Khan, F., Yarveisy, R., Abbassi, R., 2021. Cross-country pipeline inspection data analysis and testing of probabilistic degradation models. *Journal of Pipeline Science and Engineering* 1, 308–320. doi:<https://doi.org/10.1016/j.jpse.2021.09.004>. special Issue on Risk and Reliability Assessment of Pipelines.
- Kim, C., Han, W., Ju, D., Hwang, S.J., 2024. Eagle: Eigen aggregation learning for object-centric unsupervised semantic segmentation, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3523–3533.
- Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollar, P., Girshick, R., 2023. Segment anything, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4015–4026.
- Li, Y., Miao, N., Ma, L., Shuang, F., Huang, X., 2023. Transformer for object detection: Review and benchmark. *Engineering Applications of Artificial Intelligence* 126, 107021. doi:<https://doi.org/10.1016/j.engappai.2023.107021>.
- Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context, in: *Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (Eds.), Computer Vision – ECCV 2014*, Springer International Publishing, Cham. pp. 740–755.

- Liu, W., Qiao, X., Zhao, C., Deng, T., Yan, F., 2025. Vp-yolo: A human visual perception-inspired robust vehicle-pedestrian detection model for complex traffic scenarios. *Expert Systems with Applications* 274, 126837. doi:<https://doi.org/10.1016/j.eswa.2025.126837>.
- Liu, Y., Bao, Y., 2022. Review on automated condition assessment of pipelines with machine learning. *Advanced Engineering Informatics* 53, 101687. doi:<https://doi.org/10.1016/j.aei.2022.101687>.
- Mentesidis, P., Papaioannidis, C., Pitas, I., 2024. Advancing industrial inspection: A dataset for automated damage detection in insulated pipes, in: 2024 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW), pp. 720–724. doi:10.1109/ICASSPW62465.2024.10627321.
- Mo, Y., Wu, Y., Yang, X., Liu, F., Liao, Y., 2022. Review the state-of-the-art technologies of semantic segmentation based on deep learning. *Neurocomputing* 493, 626–646. doi:<https://doi.org/10.1016/j.neucom.2022.01.005>.
- Mohamad, A.J., Ali, K., Rifai, D., Salleh, Z., Othman, A.A.Z., 2023. Eddy current testing methods and design for pipeline inspection system: a review, in: *Journal of Physics: Conference Series*, IOP Publishing. p. 012030.
- Padilla, R., Netto, S.L., Da Silva, E.A., 2020. A survey on performance metrics for object-detection algorithms, in: 2020 international conference on systems, signals and image processing (IWSSIP), IEEE. pp. 237–242.
- Papadam, D.R., Mentesidis, P., Zamioudis, A., Mygdalis, V., Psarras, D., Pitas, I., 2025. Ipd: An industrial pipeline dataset for anomaly detection and localization. 33rd European Signal Processing Conference (EUSIPCO 2025) .
- Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* 28.

- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 .
- Siqueira, M., Gatts, C., da Silva, R., Rebello, J., 2004. The use of ultrasonic guided waves and wavelets analysis in pipe inspection. *Ultrasonics* 41, 785–797. doi:<https://doi.org/10.1016/j.ultras.2004.02.013>.
- Skalski, P., 2024. How to use the segment anything model (sam). URL: <https://blog.roboflow.com/how-to-use-segment-anything-model-sam/>.
- Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A., 2017. Inception-v4, inception-resnet and the impact of residual connections on learning, in: Proceedings of the AAAI conference on artificial intelligence.
- Tan, M., Le, Q., 2019. EfficientNet: Rethinking model scaling for convolutional neural networks, in: Chaudhuri, K., Salakhutdinov, R. (Eds.), Proceedings of the 36th International Conference on Machine Learning, PMLR. pp. 6105–6114.
- Tan, Y., Cai, R., Li, J., Chen, P., Wang, M., 2021. Automatic detection of sewer defects based on improved you only look once algorithm. *Automation in Construction* 131, 103912. doi:<https://doi.org/10.1016/j.autcon.2021.103912>.
- The GIMP Development Team, 2024. Gnu image manipulation program (gimp), version 3.0.4. community, free software (license gplv3). URL: <https://gimp.org/>.
- Tkachenko, M., Schevchenko, N., Liubimov, N., Malyuk, M., 2019. Label Studio. URL: <https://labelstud.io>.
- Tripathi, A., Gupta, M.K., Srivastava, C., Dixit, P., Pandey, S.K., 2022. Object detection using yolo: A survey, in: 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), IEEE. pp. 747–752.
- Vasu, P.K.A., Gabriel, J., Zhu, J., Tuzel, O., Ranjan, A., 2023. Mobileone: An improved one millisecond mobile backbone, in: Proceedings of the

- IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7907–7917.
- Wada, K., 2021. Labelme: Image polygonal annotation with python. URL: <https://github.com/wkentaro/labelme>, doi:10.5281/zenodo.5711226.
- Wang, C.Y., Yeh, I.H., Mark Liao, H.Y., 2024. Yolov9: Learning what you want to learn using programmable gradient information, in: European conference on computer vision, Springer. pp. 1–21.
- Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J., 2018. Unified perceptual parsing for scene understanding, in: Proceedings of the European Conference on Computer Vision (ECCV).
- Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P., 2021a. Segformer: Simple and efficient design for semantic segmentation with transformers, in: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc.. pp. 12077–12090.
- Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P., 2021b. Segformer: Simple and efficient design for semantic segmentation with transformers, in: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc.. pp. 12077–12090.
- Xie, J., Yang, J., Fu, K., Tai, L., Wang, X., Zhu, J., Wang, J., 2025. Quantitative assessment of pipeline defects utilizing a dual-stage deep learning framework: Integration of pretrained yolo network and multi-input parallel convolution architectures on magnetic flux leakage data. *Journal of Pipeline Science and Engineering* , 100282doi:<https://doi.org/10.1016/j.jpse.2025.100282>.
- Xie, M., Tian, Z., 2018. A review on pipeline integrity management utilizing in-line inspection data. *Engineering Failure Analysis* 92, 222–239. doi:<https://doi.org/10.1016/j.engfailanal.2018.05.010>.
- Xie, S., Girshick, R., Dollar, P., Tu, Z., He, K., 2017. Aggregated residual transformations for deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

- Xu, C., Jiang, W., Ma, H., Liu, Y., Men, H., Xu, X., Li, Z., 2025. A review: Research and application of pipeline robots in the oil and gas industry. *Journal of Pipeline Science and Engineering* , 100356doi:<https://doi.org/10.1016/j.jpse.2025.100356>.
- Yang, D., Cui, Y., Yu, Z., Yuan, H., 2021. Deep learning based steel pipe weld defect detection. *Applied Artificial Intelligence* 35, 1237–1249.
- Yang, D., Ma, C., Yu, G., Chen, Y., 2025. Automatic defect detection of pipelines based on improved ofg-yolo algorithm. *Measurement* 242, 115847. doi:<https://doi.org/10.1016/j.measurement.2024.115847>.
- Zhang, X., Zhang, X., Li, J., Niu, X., Wu, Q., Tu, J., Song, X., 2024. Pipeline thickness measurement for in-line inspection using wholly stepped electromagnetic acoustic transducers. *Nondestructive Testing and Evaluation* , 1–19.
- Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J., 2017. Pyramid scene parsing network, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Zhou, Z., Rahman Siddiquee, M.M., Tajbakhsh, N., Liang, J., 2018. Unet++: A nested u-net architecture for medical image segmentation, in: *Deep learning in medical image analysis and multimodal learning for clinical decision support: 4th international workshop, DLMIA 2018, and 8th international workshop, ML-CDS 2018, held in conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, proceedings 4*, Springer. pp. 3–11.