

Comportamento a fatica ad alta temperatura di una ghisa sferoidale Si-Mo

*Original*

Comportamento a fatica ad alta temperatura di una ghisa sferoidale Si-Mo / Matteis, P., Scavino, G., A., C., Firrao, D.. -  
In: LA METALLURGIA ITALIANA. - ISSN 0026-0843. - 107:5(2015), pp. 17-22.

*Availability:*

This version is available at: 11583/2609369 since:

*Publisher:*

Associazione Italiana di Metallurgia

*Published*

DOI:

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)



## IMAGE-BASED MULTI-DAMAGE DETECTION IN TUNNELS: A DEEP LEARNING DATASET FOR STRUCTURAL HEALTH MONITORING

M.Hamed Mozafarian<sup>1</sup>, Giuseppe Desiderio<sup>1</sup>, Zehao Ye<sup>2</sup>, Paola Alice Rosa Cavallaro<sup>1</sup>,  
Valentina Villa<sup>1</sup>, and Jelena Ninic<sup>2</sup>

<sup>1</sup>Politecnico di Torino, Turin, Italy

<sup>2</sup>Birmingham University, Birmingham, United Kingdom

### Abstract

This paper introduces a novel dataset of high-resolution panoramic images from two Italian tunnels, specifically designed for structural health monitoring (SHM). Its innovation lies in extensive annotations of key structural damages—cracks, corrosion, spalling, seepage, and damaged joints—created using the Segment Anything Model (SAM) for pixel-level segmentation and bounding box annotations, formatted in COCO-style. This comprehensive dataset supports various computer vision tasks, including classification, instance segmentation, and object detection. By enabling the benchmarking of advanced deep learning models, our work provides an essential resource for automated damage detection, significantly advancing research and practical infrastructure maintenance.

### Introduction

In 2022, the Italian Ministry of Infrastructure and Sustainable Mobility introduced new guidelines for the classification and management of risk, safety assessment, and monitoring of existing tunnels through Ministerial Decree 247/22 (Ministero delle Infrastrutture e della Mobilità Sostenibili, 2022). The primary objective of these guidelines is to establish a quantifiable level of safety to guide maintenance decision-making, aiming to minimize the risk of hazardous situations and avoid the need for urgent interventions. The document proposes a framework structured into three main steps: collecting existing data (e.g., tunnel length, location, previous maintenance interventions, etc.), risk classification, and safety assessment. The guidelines divide tunnels into segments called "Conci", each 20 meters long. Every segment is analyzed and assigned a specific risk level called the "Class of Attention". These classes serve as the foundation for decision-making in planning maintenance interventions.

As described in the guidelines, the Classes of Attention are determined through a simplified assessment of the risk factors associated with tunnels based on available knowledge and inspections. The recordings and labeling process of the defects to acquire the class of attention still remain primarily manual and done by experienced technicians. Although advanced reality capture technologies have improved the efficiency of automated tunnel data collection (Wang et al., 2024a), the manual recording and

labeling process remains highly time-consuming, labor-intensive, and prone to inconsistency (Huang et al., 2021). Automating and accelerating this phase would help to get updated Classes of Attention and better maintenance planning aligned with the actual condition of the infrastructure. Recent advancements in Computer Vision methodologies have received considerable attention for surface damage detection, attributed to their capacity to provide clear visual evidence of damage in images (Cha et al., 2024). Many studies have been utilizing computer vision techniques for surface damage detection, specifically for tunnels and underground structures (Dong et al., 2019; Zhao et al., 2021; Jiang et al., 2023). Among these techniques, computer vision-based damage detection using Deep Learning (DL) algorithms exhibits a superior performance (Ye et al., 2024). Generally, these methods can be categorized into three stages: the first one is image classification for damage identification (Hassan et al., 2019), the second one is bounding box level object detection for damage identification (Li et al., 2021), and the third is pixel-wise segmentation (Xu et al., 2021). The pixel-wise segmentation for damage detection is mainly divided into semantic segmentation (Wang et al., 2024b) and instance segmentation (Zhao et al., 2020). Each type gradually enhances the amount of information extracted from a single image, ranging from classifying damage types to detecting damage locations with bounding boxes, refining boundary definitions more accurately, and ultimately achieving instance-level recognition and segmentation.

Although computer vision DL-based algorithms have shown promising results in detecting surface damages in tunnels, challenges still persist when applying them to real-world scenarios. These difficulties arise due to the complexity of data acquisition and the lack of promising datasets for detecting severe deterioration present in tunnel environments. Various studies present datasets suitable for pixel-wise segmentation concerning detecting tunnel defects from captured images Table 1. However, most of these studies cover limited categories of defects -mostly cracks- and a comprehensive public dataset concerning various types of defects is still missing. In this paper, we introduce a methodology for creating a multi-type defect dataset suitable for instance segmentation tasks derived from Panoramic Ultra High Resolution (UHR)

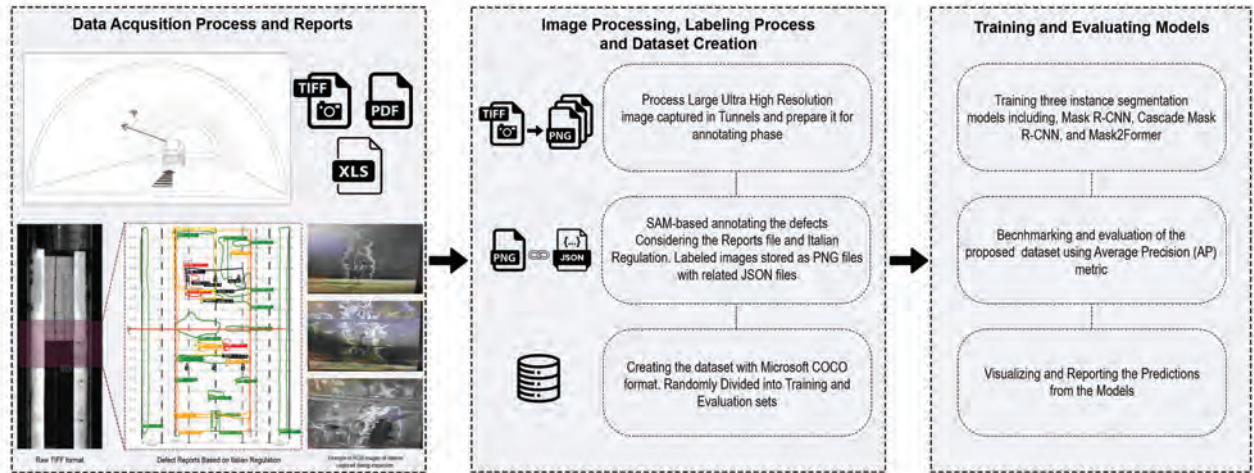


Figure 1: Overview of proposed framework

Table 1: Comparison of Existing Datasets for Tunnel Damage Detection

Dataset	Defect Type	N. Images	Resolution	Acq. Device
(Ren et al., 2020)	Crack	919	512×512	Mobile Camera
(Joshi et al., 2022)	Crack	3000	800×600	Mobile Camera
(Xu et al., 2023)	Crack Seepage	8823	1072×712	Panoramic Image Acq. Device
Ours	Seepage Spalling Corrosion Crack Dmg. Joint	1700	1024×1024	Laser Scanner

images captured from Italian concrete tunnels, and consequently, benchmark this dataset using two algorithms based on Convolutional Neural Networks (CNNs) and one Transformer-based algorithm.

## Methodology

### Overview

An overview of proposed methodology represented in Fig. 1. The first part demonstrates the data acquisition process and available reports from on-site inspection, including an illustration of the data capturing technique, an example of unprocessed raw images, and related reports. The second section details the image processing workflow and the SAM-based annotation technique used for labeling the defects, and lastly formats the dataset in compliance with the COCO standard. Finally, the last section describes the training and benchmarking procedures, utilizing three State-of-the-Art instance segmentation algorithms.

### Preparation of the Dataset

The dataset was created from images captured in two road tunnels in Italy, both constructed from concrete and in operation for more than 50 years. Their locations provide geographic diversity: one is situated in the Abruzzo region

in eastern-central Italy, while the other is in the Liguria region in the northwest. The images in this study were acquired using a Mobile Laser Scanner, as laser point clouds are commonly used for data collection under limited lighting conditions. The scanner used was the TS4 model developed by the Spacetec company (SpaceTec, 2024). This instrument utilizes advanced technology to achieve high-accuracy surveys in 2- or 3-lane tunnels. Its 360-degree laser scanner captures the entire tunnel vault and roadway by measuring distances through phase-difference calculations. With a rotating head acquiring up to 10,000 points per full rotation, it collects data while mounted on a vehicle moving at around 5 km/h. The scanner head rotates at 200 revolutions per second, generating detailed point clouds that represent the tunnel surface and provide reflectance data for material analysis. While the primary output of this instrument is a point cloud, various outputs can be derived from this, including the unwrapped tunnel panoramic images used for this study. These images are generated from a rigorous three-dimensional representation, which ensures precision and reliability in the analysis. The images obtained represent the full length of the tunnels, and due to the different lengths of each tunnel, each image has a unique dimension. For example, the images of two tunnels discussed in this paper have  $10.000 \times 158.679$  and  $10.000 \times 212.414$  pixels. The images were stored in the .TIFF format as 8-bit grayscale. Afterward, the unwrapped panoramic view is obtained, the real-world size corresponding to each pixel is calculated according to the tunnel dimensions described in the tunnel documentation.

Each tunnel has a report file containing a description of different defects based on the Italian guidelines - (Ministero delle Infrastrutture e della Mobilità Sostenibili, 2022) - represented in every 20-meter section, by experienced engineers through visual inspection. Consequently, the tunnel images were divided into 20-meter longitudinal sections along the tunnel direction and the pavement parts on

both sides were removed, leaving the tunnel lining for further investigations.

In the images, the crown part of the tunnels and the side walls have a noticeable color difference. The crown part is significantly darker, while the side walls are lighter. To decrease color differences in the images, to facilitate the labeling phase, and also to have better training performance, a gamma correction (Bradski, 2000) equal to 0.5 was applied for both tunnels and improved the quality of the images as shown in Fig. 2.

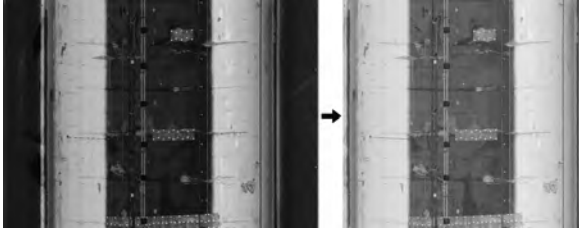


Figure 2: Comparison of a tunnel section before and after processing. The raw captured image is shown on the left, while the processed version is on the right.

To make the labeling phase more efficient, each 20-meter tunnel section is segmented into six patches of equal size, respectively, three columns and two rows. The size of these cropped images differs from each tunnel; the first was sized  $2155 \times 3539$  and the other images related to the second tunnel were sized  $2144 \times 3821$ . For the labeling phase, a semiautomatic annotation tool based on the Segment Anything Model (SAM) (Kirillov et al., 2023) (Ji and Zhang, 2023) is used, this software significantly enhances labeling efficiency by offering a prompt-based method to create object masks with just a few points, which can later be manually refined by adjusting the boundaries. Two experts were responsible for annotating the images; one person performed the initial labeling and the other person was responsible for reviewing the annotations to check the reliability and consistency of different classes.

The dataset contains five different types of defects. In the first stage, the labeling phase is carried out by annotating the exact types of defects introduced by the guideline (Ministero delle Infrastrutture e della Mobilità Sostenibili, 2022). This guideline introduces 61 types of defects in 12 categories. After implementing some tests, due to the poor result of distinguishing the different defect classes, the team decided to merge sub-categories and came up with five classes for labeling the defects. Table 2 shows the taxonomy of defects, while Fig. 3 shows an example of these classes taken from the dataset. Each labeled image contains an annotation file in *JSON* format generated by the software (SAM). Consequently, the annotation file was converted into a Microsoft COCO dataset (Lin et al., 2014) through a *Python* script.

The total number of annotated images is 306, the annotation consists of a dictionary containing two "main" keys: *info*, where the image is described, and "objects" which

contains the list of *things* that has been categorized <sup>1</sup>.

### Image Instance Segmentation

The objective of instance segmentation is to detect and delineate each individual object instance in an image, even if they belong to the same semantic category. As output, we get a set of masks or bounding boxes, each associated with a unique instance ID and a class label. In summary, instance-wise segmentation of defects, answering "Where are the individual objects and what are they?". In this study, we benchmark our proposed dataset with three widely used and representative instance segmentation algorithms: two CNN-based architectures, Mask R-CNN (He et al., 2017) and Cascade Mask R-CNN (Cai and Vasconcelos, 2021), as well as one Transformer-based model, Mask2Former (Cheng et al., 2022). In the future, we plan to conduct a more comprehensive comparison by evaluating additional CNN- and Transformer-based architectures alongside the publication of the dataset.

**Mask R-CNN:** designed for instance segmentation, distinguishes itself through its efficient, accurate, and extensible architecture, achieving high-quality instance segmentation by effectively integrating mask prediction within the Faster R-CNN framework (in a pixel-to-pixel manner for each Region of Interest (RoI)) and introducing the RoIAlign layer for precise spatial alignment (avoiding quantization by using bilinear interpolation).

**Masked-attention Mask Transformer (Mask2Former):** is presented as a novel architecture for "universal" image segmentation. It offers a flexible, efficient, and high-performing architecture because of several architectural-design choices like *Masked Attention* - restricts cross-attention within predicted mask regions, *Multi-Scale High-Resolution Features* - a feature pyramid from a pixel decoder incorporates both low- and high-resolution features) and *Optimized Transformer Decoder* - switching self- and cross-attention order, learnable query features, and removal of dropout. It represents a significant advancement towards a unified approach for diverse segmentation tasks.

**Cascade Mask R-CNN:** Cascade R-CNN introduces a multistage architecture comprising a sequence of detectors trained with progressively increasing IoU thresholds. Detectors, within the cascade, are trained sequentially with a higher IoU threshold than the preceding stage - *Sequential Detector Training*, in this way each stage refines the input of the object hypotheses for the subsequent - *Progressive Hypothesis Refinement*. Thus, each detector stage is trained with a sufficient and relevant set of positive examples - *Resampling Mechanism*. At inference time, the same

<sup>1</sup>this part will be public available along with images used for training and test.

Table 2: Taxonomy of Defects

Guidelines Categories	Code	Defects Description	New Class
Defects caused by water presence	1.1	Drippings	Seepage
	1.2	Water Ingress	
	1.3	Concretions – Deposits – Encrustations	
	1.4	Effects of frost - traces of salts	
	1.5	Efflorescence on mortar or concrete	
Defects in the coating materials (concrete)	1.19	Cracks and spalling due to reinforcement corrosion	Corrosion
Defects related to the structural elements and geometry of the tunnel	3.1	Presence of longitudinal cracks along the coating	Cracks
	3.2	Diagonal cracks	
	3.3	Vertical cracks	
	3.4	Shrinkage cracks	
	3.5	Curvilinear cracks	
Defects relating to the structural elements and geometry of the tunnel- Construction Defects	3.14	Deterioration of concrete joints	Damaged Joint
	3.15	Surface defects in concrete	Spalling

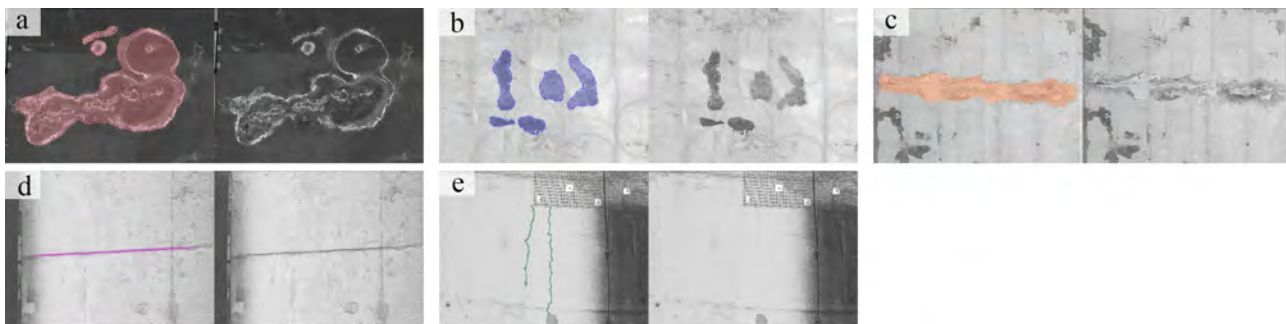


Figure 3: Categories and corresponding annotations: a) Seepage, b) Spalling, c) Corrosion, d) Damaged Joint, and e) Crack

sequential refinement process is applied, aligning the quality of hypotheses with the increasing quality of detectors in the cascade.

## Experiments

### Preprocessing

The original images in the dataset had dimensions of  $2155 \times 3539$  and  $2144 \times 3821$ , respectively. A preprocessing step was applied to standardize the input size for the models. This step involved cropping each image into six patches, arranged in three columns and two rows. Each of these patches was then centrally cropped further to obtain square images of size  $1024 \times 1024$  (model input images). To accommodate these modifications, the COCO annotation files were updated to correctly reflect the new dimensions of the images.

Once cropped in  $1024 \times 1024$ , the dataset comprises 1800 images. Among these images, 100 were found without annotations and excluded from the dataset. As a result, the dataset counts 1700 images with 6821 annotations spanning five categories. For model training, the dataset was divided into training and validation sets. Following established practices in the field, 80% of the images were randomly assigned to the training set, and the remaining 20% to the validation set. To ensure robustness and reliability, all experiments were repeated three times, each with a different train-validation split.

However, one limitation of this approach is that the split was performed on the images rather than on the categories. As a result, the categories are not uniformly distributed between the training and validation sets, leading to an imbalanced dataset. This imbalance could affect the robustness of the models. Addressing this issue will be the focus of future work, in which the authors plan to explore strategies to achieve a more balanced division of categories across the splits.

### Hardware

All experiments were conducted using NVIDIA A100 Tensor Core GPUs. The software environment consists of MMCV 2.1 and PyTorch 2.0.1 with CUDA version 11.7.

### Experiment Parameter Setting

Three instance segmentation algorithms were utilized. For each experiment, the MMDetection configuration file (Chen et al., 2019) was used for all algorithms. For data augmentation strategies, the configuration in the official setup was applied. All models were implemented using the Swin Large (Liu et al., 2021) backbone, initialized with ImageNet-22k pre-trained weights (Deng et al., 2009), to leverage transfer learning benefits and improve generalization. The input images were standardized to  $1024 \times 1024$  to balance memory constraints and segmentation accuracy. The models were trained for 100 epochs us-

ing the AdamW optimizer (Loshchilov and Hutter, 2019), selected due to its superior convergence properties compared to other optimizer types in our preliminary tests. The learning rate schedule follows a cosine annealing strategy (Loshchilov and Hutter, 2017). Training begins with a linear warm-up phase lasting 1000 iterations to prevent instability in the early training stages. The initial learning rate was set at  $1e-4$  and gradually reduced to  $1e-7$ , as preliminary experiments indicated that this schedule led to better convergence and avoided overfitting.

### Performance Metrics

The instance segmentation algorithms were used to perform the defect detection task. The evaluation of instance segmentation models consistently follows the widely recognized official COCO (Microsoft Common Objects in Context) evaluation metric for instance segmentation tasks (Lin et al., 2014). A prediction is correct if the Intersection over Union (IoU) between the predicted instance’s box or mask and the corresponding ground truth exceeds a threshold  $T$  (in our experiment we refer to Detection Evaluation metrics used by COCO for thresholds) and the expected category matches the ground truth category. Intersection Over Union (IoU) and Average Precision (AP) are widely used metrics in computer vision, particularly in object detection and segmentation tasks, to evaluate the accuracy of a predicted region against a ground truth region. IoU metric measures the overlap between two bounding boxes or regions, providing a standardized way to quantify how well a prediction aligns with the true object location. The IoU is calculated using the following equation:

$$IoU = \frac{TP}{TP + FP + FN} \quad (1)$$

where  $TP$ ,  $FN$ , and  $FP$  denote the true positive, false negative, and false positive.

To better understand the results obtained, we provided a visual inspection of the predictions of each model. This analysis has two objectives: on one side, we make a comparison between the different models, and on the other side, we try to capture which classes are poorly predicted by each single model.

A prediction is considered accurate if the  $IoU$  between the predicted instance’s result and the ground truth exceeds a certain threshold  $T$ , and the predicted category matches the ground truth category. The related evaluation method is as follows:

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \quad (2)$$

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \quad (3)$$

$$AP = \frac{1}{101} \times \sum_{r \in \{0, 0.01, \dots, 1\}} p_{interpolation}(r) \quad (4)$$

where  $TP_i$ ,  $FN_i$ , and  $FP_i$  denote the true positive, false negative, and false positive instances  $i$ , and  $p_{interpolation}(r)$

Table 3: Result of Models

Method	$AP_{box}$	$AP_{box}^{50}$	$AP_{box}^{75}$	$AP_{mask}$	$AP_{mask}^{50}$	$AP_{box}^{75}$
Mask2Former	<b>0.258</b>	0.426	0.249	<b>0.224</b>	<b>0.413</b>	<b>0.211</b>
Mask R-CNN	0.241	<b>0.452</b>	0.228	0.222	<b>0.413</b>	0.206
Cascade Mask R-CNN	0.255	0.424	<b>0.259</b>	0.213	0.383	0.206

is the precision obtained by interpolation at the given maximum recall level  $r$ . The precision is averaged over the set of 101 equally spaced recall levels  $[0.0, 0.01, 0.02, \dots, 1.0]$ . Here,  $i = 1, 2, 3, \dots, n$ , where  $n$  is the total number of instances. The evaluation of instance segmentation encompasses both bounding boxes and masks, with the results presented as  $AP_{box}$  for bounding boxes and  $AP_{mask}$  for masks.  $AP_{mask}$  is regarded as the key metric in this study, as our focus is on segmentation performance. A higher value of  $AP$  corresponds to more precise predictions of the instance box and mask, reflecting improved instance localization and segmentation accuracy. Specifically,  $AP^{50}$  evaluates the performance with an IoU threshold of 0.50, while  $AP^{75}$  is a more rigorous metric using an IoU threshold of 0.75. Consequently,  $AP^{75}$  provides a more accurate assessment of box and mask accuracy compared to  $AP^{50}$ .

### Results

This section presents the results of the experiments to evaluate the efficacy of the instance segmentation models. Table 3 shows the overall performance metrics, calculated on the validation set considering the best model for each experiment. The best model is the model that reaches the highest score of  $AP_{mask}$  on the validation set, during training. Then Table 4 shows the evaluation concerning each category.

Table 3 reveals that Mask2Former achieves the highest overall performance, with an  $AP_{mask}$  of 0.224 on the validation set. While the Mask R-CNN achieves a similar  $AP_{mask}$  of 0.222. Cascade Mask R-CNN, the other compared model, had the lowest  $AP_{mask}$  of 0.213 in the whole experiment. It is important to notice that Mask R-CNN has a slightly higher result on  $AP_{box}$  with 0.452 at IoU of 50%; however, the overall performance in  $AP_{mask}$  was not satisfactory compared to Mask2Former.

A detailed analysis per category is presented in Table 4. In terms of specific defect categories, it’s important to note the following:

- **Seepage:** Both Mask R-CNN and Mask2Former demonstrate high performance, but Mask2Former achieves the highest  $AP_{mask}$  of 0.396, indicating its efficiency in identifying this class of defects. Cascade Mask R-CNN has the lowest performance with  $AP_{mask} = 0.333$ .
- **Spalling:** Mask2Former also shows the highest performance in this case as well, outperforming other

Table 4: Comparative Analysis of instance segmentation algorithms on our dataset for specific categories.

Method	Categories	$AP_{box}$	$AP_{box}^{50}$	$AP_{box}^{75}$	$AP_{mask}$	$AP_{mask}^{50}$	$AP_{mask}^{75}$
Mask R-CNN	Seepage	0.378	0.561	0.408	0.364	0.562	0.384
	Corrosion	0.081	0.152	0.059	0.072	0.156	0.052
	Damaged Joint	0.229	0.555	0.117	0.112	0.393	0.021
	Spalling	0.407	0.549	0.447	0.408	0.545	0.464
	Crack	0.054	0.101	0.029	0.016	0.094	0.00
Mask2Former	Seepage	0.404	0.596	0.422	0.396	0.624	0.420
	Corrosion	0.081	0.216	0.070	0.088	0.210	0.063
	Damaged Joint	0.314	0.640	0.246	0.158	0.561	0.054
	Spalling	0.468	0.619	0.494	0.476	0.656	0.518
	Crack	0.023	0.061	0.013	0.003	0.012	0.00
Cascade Mask R-CNN	Seepage	0.355	0.485	0.386	0.333	0.484	0.368
	Corrosion	0.100	0.162	0.113	0.081	0.149	0.059
	Damaged Joint	0.182	0.333	0.126	0.082	0.324	0.010
	Spalling	0.418	0.510	0.460	0.417	0.513	0.456
	Crack	0.052	0.079	0.079	0.025	0.079	0.00

models with  $AP_{mask} = 0.476$ , which indicates superior precision in segmenting spalling defects. While Mask R-CNN and Cascade Mask R-CNN show a good performance, they have lower results compared to Mask2Former.

- **Corrosion:** Mask2Former had an  $AP_{mask}$  of 0.088, this is a slight improvement over both Mask R-CNN (0.072) and Cascade Mask R-CNN (0.081). This shows that all models struggle with the accurate identification and delineation of corrosion-related instances.
- **Damaged Joints:** In this category, Mask2Former achieves the highest  $AP_{mask}$  score, of 0.158. The results reveal that Mask R-CNN and Cascade Mask R-CNN performances are lower compared to other models, indicating an overall difficulty in detecting and segmenting damaged joints.
- **Cracks:** All three algorithms exhibit a poor performance in detecting cracks, the highest  $AP_{mask}$  of Mask R-CNN was only 0.016, and Mask2Former and Cascade Mask R-CNN both obtained lower results of 0.003 and 0.025 respectively.

These findings suggest that while Mask2Former generally performs best across the dataset and particularly excels in segmenting *Seepage* and *Spalling*, the limited performance of all models concerning detecting *Cracks* needs to be addressed. Although in literature, there exist many data sets which focus mainly on *Crack* defects like MCrack1300 (Ye et al., 2024). We claim that our data set can be combined with existing data sets to fill this performance gap. Some visualized results derived from experiments are shown in Fig. 4. These visualizations enable a qualitative assessment of the models’ performance, helping to understand where the algorithms succeed or fail. For example, Mask2Former correctly identifies large seepage regions,

sometimes underestimating the size of smaller instances. Mask R-CNN shows decent performance in spalling but can struggle with accuracy when there are significant variations in texture or contrast. Cascade Mask R-CNN displays a consistent but moderate performance compared to both Mask R-CNN and Mask2Former.

## Conclusions

In this paper, we present a novel method for processing high-resolution panoramic images from road tunnels to develop a database suitable for computer vision models to enhance the defect detection process concerning structural health monitoring criteria to satisfy the new Italian regulation concerns regarding to guide maintenance decision-making, aiming to minimize the risk of dangerous situations and prevent the need for urgent interventions. The database consists of five different categories of concrete defects based on Italian regulations, including seepage, spalling, crack, damaged joint, and corrosion. Subsequently, the database was trained and evaluated with instance segmentation algorithms to verify efficacy and performance.

The Mask2Former has shown the overall best performance with  $AP_{mask}$  equal to 0.224. Regarding the categories, seepage and spalling have a satisfactory result, while damaged joints and corrosion have shown a slightly moderate performance. For the cracks, performance was not satisfactory. This may have been due to the lack of subsequent data on this category in the database.

There are some limitations in our research. Firstly, the imbalance of the dataset between different classes affects the performance and efficacy of the model. Furthermore, more state-of-the-art architectures could be explored to better evaluate the dataset and enhance performance analysis.

In future work, our aim is to address the class imbalance by introducing more instances in underrepresented classes. In addition, we plan to incorporate diverse architectures

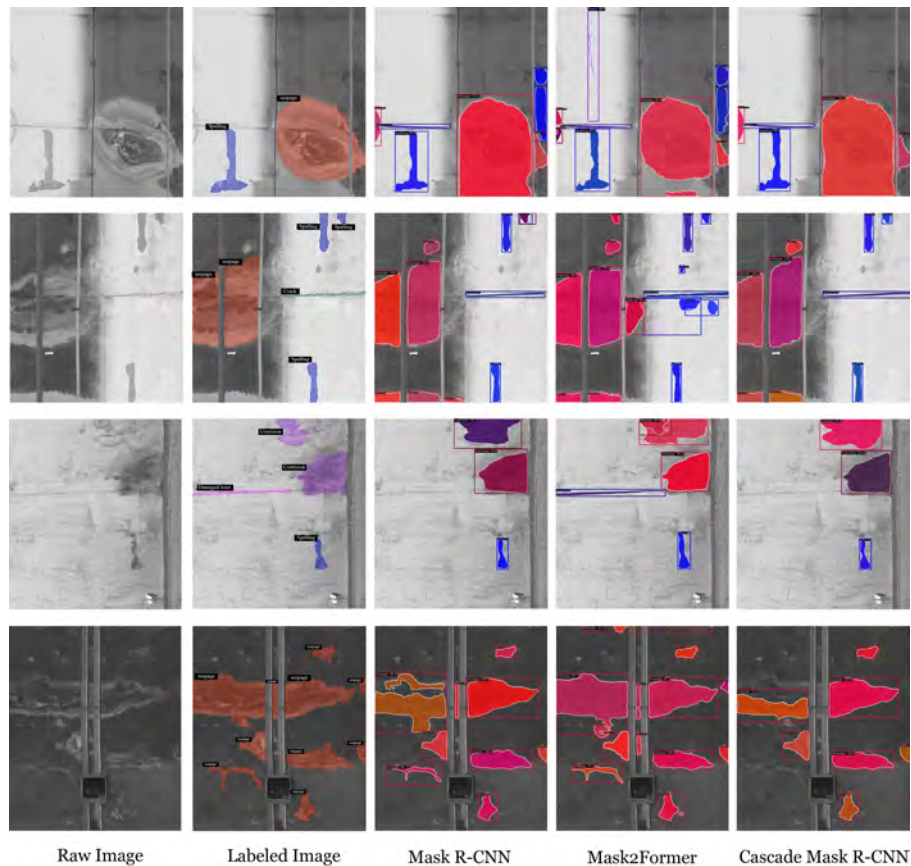


Figure 4: Example of predictions

to gain deeper insights into dataset performance. Lastly, integrating multimodal data sources, such as thermal and RGB-D cameras, as well as data-driven Ground Penetrating Radar (GPR) methodologies, could further enhance overall model performance.

## Acknowledgments

The computations described in this research were performed using the Baskerville Tier 2 HPC service (<https://www.baskerville.ac.uk/>). Baskerville was funded by the EPSRC and UKRI through the World Class Labs scheme (EP/T022221/1) and the Digital Research Infrastructure programme (EP/W032244/1) and is operated by Advanced Research Computing at the University of Birmingham. The computations described in this paper were also performed using the University of Birmingham's BlueBEAR (<http://www.birmingham.ac.uk/bear>) HPC service, which provides a High-Performance Computing service to the University's research community.

The engineering company TECNE underpinned the research and contributed all necessary data. Their ongoing collaboration has made it possible to engage with practical applications relevant to the field.

## References

- Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools.
- Cai, Z. and Vasconcelos, N. (2021). Cascade r-cnn: High quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(5):1483–1498.
- Cha, Y.-J., Ali, R., Lewis, J., and Büyük ztürk, O. (2024). Deep learning-based structural health monitoring. *Automation in Construction*, 161:105328.
- Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C. C., and Lin, D. (2019). MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*.
- Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., and Girdhar, R. (2022). Masked-attention mask transformer for universal image segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1280–1289.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database.
- Dong, Y., Wang, J., Wang, Z., Zhang, X., Gao, Y., Sui, Q., and Jiang, P. (2019). A deep-learning-based multi-

- ple defect detection method for tunnel lining damages. *IEEE Access*, 7:182643–182657.
- Hassan, S. I., Dang, L. M., Mehmood, I., Im, S., Choi, C., Kang, J., Park, Y.-S., and Moon, H. (2019). Underground sewer pipe condition assessment based on convolutional neural networks. *Automation in Construction*, 106:102849.
- He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.
- Huang, M., Ninić, J., and Zhang, Q. (2021). Bim, machine learning and computer vision techniques in underground construction: Current status and future perspectives. *Tunnelling and Underground Space Technology*, 108:103677.
- Ji, S. and Zhang, H. (2023). ISAT with Segment Anything: An Interactive Semi-Automatic Annotation Tool. Updated on 2023-06-03.
- Jiang, Y., Wang, L., Zhang, B., Dai, X., Ye, J., Sun, B., Liu, N., Wang, Z., and Zhao, Y. (2023). Tunnel lining detection and retrofitting. *Automation in Construction*, 152:104881.
- Joshi, D., Singh, T. P., and Sharma, G. (2022). Automatic surface crack detection using segmentation-based deep-learning approach. *Engineering Fracture Mechanics*, 268:108467.
- Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al. (2023). Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026.
- Li, D., Xie, Q., Gong, X., Yu, Z., Xu, J., Sun, Y., and Wang, J. (2021). Automatic defect detection of metro tunnel surfaces using a vision-based inspection system. *Advanced Engineering Informatics*, 47:101206.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.
- Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002.
- Loshchilov, I. and Hutter, F. (2017). Sgdr: Stochastic gradient descent with warm restarts. Cited by: 2118.
- Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*.
- Ministero delle Infrastrutture e della Mobilità Sostenibili (2022). Linee guida per la classificazione e gestione del rischio, la valutazione della sicurezza ed il monitoraggio delle gallerie esistenti (guidelines for risk classification and management, safety assessment, and monitoring of existing tunnels). Technical Report Parere n. 29/2022, Consiglio Superiore dei Lavori Pubblici, Rome, Italy. Espresso dall’Assemblea Generale in data 08.04.2022.
- Ren, Y., Huang, J., Hong, Z., Lu, W., Yin, J., Zou, L., and Shen, X. (2020). Image-based concrete crack detection in tunnels using deep fully convolutional networks. *Construction and Building Materials*, 234:117367.
- SpaceTec (2024). Ts4 product page. Accessed: 31 March 2025.
- Wang, J., Zhang, S., Guo, H., Tian, Y., Liu, S., Du, C., and Wu, J. (2024a). Stereoscopic monitoring of transportation infrastructure. *Automation in Construction*, 164:105472.
- Wang, Y., Liao, W., Dong, A., Xu, L., Zhu, L., Shi, H., and Yu, Z. (2024b). High-speed acquisition and intelligent tunnel surface defects recognition. *Tunnelling and Underground Space Technology*, 144:105572.
- Xu, L., Wang, Y., Dong, A., Zhu, L., Shi, H., and Yu, Z. (2023). Image-based intelligent detection of typical defects of complex subway tunnel surface. *Tunnelling and Underground Space Technology*, 140:105266.
- Xu, Y., Li, D., Xie, Q., Wu, Q., and Wang, J. (2021). Automatic defect detection and segmentation of tunnel surface using modified mask r-cnn. *Measurement*, 178:109316.
- Ye, Z., Lovell, L., Faramarzi, A., and Ninić, J. (2024). Sam-based instance segmentation models for the automation of structural damage detection. *Advanced Engineering Informatics*, 62:102826.
- Zhao, S., Zhang, D., Xue, Y., Zhou, M., and Huang, H. (2021). A deep learning-based approach for refined crack evaluation from shield tunnel lining images. *Automation in Construction*, 132:103934.
- Zhao, S., Zhang, D. M., and Huang, H. W. (2020). Deep learning-based image instance segmentation for moisture marks of shield tunnel lining. *Tunnelling and Underground Space Technology*, 95:103156.