Improved Robotic Platform to perform Maintenance and Upgrading Roadworks: The HERON Approach

Grant Agreement Number: 955356

D3.1: AI - driven image segmentation and feature extraction

Work package	WP3: AI-based algorithms and tools Recognition, Classification and Localisation of the Points of Interest
Activity	Task 3.1: AI-driven image segmentation and feature extraction
Deliverable	D3.1: AI - driven image segmentation and feature extraction
Authors	Iason Katsamenis, Eftychios Protopapadakis, Thanasis Sakelliou, Charalampos Zafeiropoulos, Anastasios Doulamis, Nikolaos Doulamis, Matthaios Bimpas, Dimitris Kalogeras, Nikos Frangakis
Status	Final (F)
Version	1.0
Dissemination Level	Public (PU)
Document date	31/05/2022
Delivery due date	31/05/2022
Actual delivery date	31/05/2022
Internal Reviewers	Miquel Cantero (ROB), Lionel Ott (ETHZ)
	This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 955356.

Document Control Sheet

Version history table			
Version	**Date**	**Modification reason**	**Modifier**
0.1	01/04/2022	Initial Table of Contents and basic structure of the deliverable	Iason Katsamenis, Eftychios Protopapadakis
0.2	15/04/2022	Introduction and figures of the deliverable	Iason Katsamenis
0.3	29/04/2022	Deep learning algorithms: Object detection models	Iason Katsamenis
0.4	26/05/2022	Sensing and hardware specification	Thanasis Sakelliou, Iason Katsamenis
0.5	26/05/2022	Deep learning algorithms: Segmentation model	Thanasis Sakelliou
0.6	27/05/2022	Scheduled Inspection Procedures	Iason Katsamenis Charalampos Zafeiropoulos
0.7	27/05/2022	Conclusions	Iason Katsamenis
1.0	31/05/2022	Final version ready for submission	Anastasios Doulamis

Legal Disclaimer

Table of Contents

TABLE OF CONTENTS 3
LIST OF TABLES 4
LIST OF FIGURES 5
ABBREVIATION LISTS 6
GLOSSARY OF TERMS 8
EXECUTIVE SUMMARY 9
1 INTRODUCTION 10
1.1 PURPOSE OF THE DOCUMENT 10
1.2 INTENDED AUDIENCE 10
1.3 INTERRELATIONS 10
2 SCHEDULED INSPECTION PROCEDURES 12
2.1 EMPLOYED INSPECTION PROCESSES AND MISSION PLANNING 13
2.1.1 Monitoring objects 13
Cracks 14
Potholes 14
Road surface markings 14
Removable urban pavement 14
Traffic cones 14
2.1.2 Monitoring impact 15
2.1.3 Monitoring strategy - mission planning 15
2.2 HERON PILOT SITES 19
2.2.1 Spanish pilot (ACCI) 19
2.2.2 French pilot (UGE) 20
2.2.3 Greek pilot (OLO) 20
3 SENSING AND HARDWARE SPECIFICATION 22
3.1 SENSORS 22
3.2 PROCESSING COMPONENTS SPECIFICATION 23
4 DEEP LEARNING ALGORITHMS 25
4.1 COMPUTER VISION TASKS 26
4.1.1 Classification 28
Types of image classification techniques 29
Disadvantages of classification methods 29
Image classification using a general multi-label CNN classifier 29
4.1.2 Object detection 30
How object detection works 30
Disadvantages of object detection methods 31
Object detection using YOLOv5 (You Only Look Once) model 31
4.1.3 Image segmentation 32
Semantic segmentation 32
Semantic segmentation using a U-Net model 32

4.2 DEGRADATION AND ROAD DEFECT TYPES 34
4.3 3D INFORMATION EXTRACTION 35
4.4 REAL-TIME TRAFFIC CONE DETECTION 36
4.4.1 Object detection model 37
4.4.2 Dataset description 37
4.4.3 Experimental setup - Model training 39
4.4.4 Evaluation metrics 39
4.4.5 Experimental validation 39
4.4.6 Evaluation of the object detector on road images with traffic cones 40
4.4.7 Evaluation of the object detector on road images without traffic cones 43
4.5 REAL-TIME ROAD DEFECT DETECTION 48
4.5.1 Object detection model 48
4.5.2 Dataset description 48
4.5.3 Experimental setup - Model training 50
4.5.4 Evaluation metrics 51
Detection task 51
Classification task 51
4.5.5 Experimental validation 51
4.5.6 Evaluation of the object detector on road images with cracks 53
4.5.7 Evaluation of the object detector on road images with potholes 56
4.5.8 Evaluation of the object detector on road images blurred road markings 57
4.5.9 Evaluation of the object detector on road images with more than one defects 59
4.5.10 Evaluation of the object detector on road images without road defects 60
4.6 REAL-TIME ROAD SURFACE MONITORING USING UAV IMAGES 65
4.6.1 Object detection model 65
4.6.2 Dataset description 66
4.6.3 Experimental setup - Model training 67
4.6.4 Evaluation metrics 67
4.6.5 Experimental validation 67
4.6.6 Evaluation of the object detector on UAV images with cracks and potholes 69
4.7 PIXEL-LEVEL CRACK SEMANTIC SEGMENTATION 73
4.7.1 Dataset description 73
4.7.2 Semantic segmentation model 74
4.7.3 Evaluation metrics 74
4.7.4 Model training and results 76
4.7.5 Evaluation of the U-Net model 76
5 CONCLUSIONS 78
REFERENCES 78

List of Tables

Table 1: Abbreviations 6

Table 2: Abbreviations of the Partners' names 7
Table 3: Glossary of terms 8
Table 4: Summary of the expected reduction of resources due to the usage of the HERON system. ... 15
Table 5: Monitoring strategy and technique for each of the identified use cases of the HERON project. 16
Table 6: Specifications of the Zed 2i. 22
Table 7: Specifications of the MER2-2000-19U3C. 23
Table 8: Technical Specifications of Jetson TX2. 24
Table 9: Number of instances per damage type in the train dataset. 49
Table 10: Road damage types and definitions considered in the work of [33]. 49
Table 11: Average metrics of the current training setup on the test dataset. 76
Table 12: Average metrics for each class on the test dataset. 76

List of Figures

Figure 1: Interrelation between the various WPs of the HERON project. 11
Figure 2: From traditional tools to robotic sensors and actuators. 12
Figure 3: HERON's concept. 13
Figure 4: Employed inspection processes. 13
Figure 5: Identified use cases of the HERON project and mission planning of the maintenance process. 16
Figure 6: The main maintenance procedures that will be performed by the HERON system, i.e., (A) patching potholes, (B) replacement of RUP elements, (C) sealing cracks, (D) painting blurred road markings, and (E) dispensing and removing traffic cones in an automated and controlled manner. 18
Figure 7: UAV images of A2 showing the typical maintenance after cracks sealing and patching. 19
Figure 8: The demonstration sites to be offered by UGE in France. 20
Figure 9: Urban Pavement for smart city planning: the pre-fabricated road. 20
Figure 10: Photos from the initially selected demonstration sites to be used from OLO, including tunnels (left), bridges, and interchanges (right). 21
Figure 11: Zed 2i - industrial AI stereo camera. 22
Figure 12: MER2-2000-19U3C – scan industrial camera. 23
Figure 13: Jetson TX2 Module. 24
Figure 14: Non-deteriorated road surfaces (a, d) tend to present a more uniform distribution of colors compared to defected ones (b, c, e). 25
Figure 15: Comparison of the three different deep learning approaches, in the crack identification task. 26
Figure 16: Representative photographic examples of the three road defects (i.e., crack, pothole, and blurred road marking) as well as their automated localization using a deep CNN classifier. 27
Figure 17: Overview of object recognition computer vision tasks. 28
Figure 18: The architecture of a CNN model for multi-label defect detection. 30
Figure 19: The architecture of the model YOLOv5, which consists of three parts: (i) Backbone: CSPDarknet, (ii) Neck: PANet, and (iii) Head: YOLO Layer. The data are initially input to CSPDarknet for feature extraction and subsequently fed to PANet for feature fusion. Lastly, the YOLO Layer outputs the object detection results (i.e., class, score, location, size). 31
Figure 20: A schematic representation of the global-local analysis performed by a fully convolutional network model. 33
Figure 21: Architecture of the U-Net model presented in the work of [46]. 33
Figure 22: Sensing interface and AI component of the HERON system. 34
Figure 23: Each RGB image (a) has a corresponding .txt file with the bounding box information (b) of the traffic cones (ID, x, y, w, h). 37
Figure 24: Indicative images from the traffic cone detection dataset. 38
Figure 25: Calculation of the IoU metric. The predicted bounding box is depicted in green color and the ground truth in red. 39
Figure 26: Automated localization of traffic cones (red bounding boxes) on the test set of a custom dataset using small YOLOv5 deep model. 43

Figure 27: Evaluation of the YOLOv5 object detector on road images without traffic cones. 47
Figure 28: Sample images from the training dataset for potholes, cracks, and blurred marking defects, captured in Japan (a), India (b), and the Czech Republic (c). 50
Figure 29: Micro, macro and weighted IoU scores. 52
Figure 30: Classification scores calculated with micro, macro, and weighted averaging, respectively. 52
Figure 31: Classification scores calculated per class 53
Figure 32: Automated localization of cracks (red bounding boxes) on the test set of a custom dataset using small YOLOv5 deep model trained on the dataset of [6]. 56
Figure 33: Automated localization of potholes (pink bounding boxes) on the test set of a custom dataset using small YOLOv5 deep model trained on the dataset of [6]. 57
Figure 34: Automated localization of blurred road markings (orange bounding boxes) on the test set of a custom dataset using small YOLOv5 deep model trained on the dataset of [6]. 58
Figure 35: Automated localization of (i) cracks (red bounding boxes), (ii) potholes (pink bounding boxes), and (iii) blurred road markings (orange bounding boxes) on challenging images containing more than one road defect using small YOLOv5 deep model trained on the dataset of [6]. 60
Figure 36: Evaluation of the YOLOv5 object detector on road images without defects. 64
Figure 37: Sample images from the dataset [51] that contain UAV images for crack and pothole recognition. 66
Figure 38: Micro, macro and weighted IoU scores. 67
Figure 39: Classification scores calculated with micro, macro, and weighted averaging, respectively. 68
Figure 40: Classification scores calculated per class 68
Figure 41: Automated localization of (i) cracks (pink bounding boxes) and (ii) potholes (red bounding boxes) on UAV images using small YOLOv5 deep model trained and tested on the dataset of [50]. 72
Figure 42: Example of the original image (a) and the masks corresponding to the road region (b), pothole (c), and crack (d). 73
Figure 43: Representation of the HDV used by NDTI: (a) satellite tracking system (b) high-resolution camera (c) recording cameras (d), precision odometer and (e) laser sensors. 74
Figure 44: Precision and recall. 75
Figure 45: Illustration of Dice Coefficient. 2×Overlap/Total number of pixels. 75
Figure 46: Evaluation of the U-Net model on road images with crack defects. 77

Abbreviation Lists

Table 1: Abbreviations

Abbreviation	Definition
AI	Artificial Intelligence
AR	Augmented Reality
CCTV	Closed-Circuit Television
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CUD	Chaussée Urbaine Démontable (Demountable Urban Roadway)
CV	Computer Vision
DL	Deep Learning
DSLR	Digital Single Lens Reflex
EEM	Early Equipment Management
FCN	Fully Convolutional Network
FHD	Full High-Definition
FN	False Negative
FP	False Positive
GPS	Global Positioning System
GPU	Graphics Processing Unit

Abbreviation	Definition
GRDDC	Global Road Damage Detection Challenge
HDV	Highway Diagnostic Vehicle
HSV	Hue Saturation Value
IoU	Intersection over Union
MDOF	Multi-Degree-Of-Freedom
ML	Machine Learning
OSH	Occupational Safety and Health
PoIs	Points of Interest
RAM	Random Access Memory
RGB	Red Green Blue
RI	Road Infrastructure
RoI	Region of Interest
RUP	Removable Urban Pavement
SSD	Single Shot Detector
TN	True Negative
TP	True Positive
UAV	Unmanned aerial vehicle
UGV	Unmanned Ground Vehicle
VRAM	Video Random Access Memory
VS	Visual Servoing
YOLO	You Only Look Once

Table 2: Abbreviations of the Partners' names

Short name	Participant organization name
ICCS	Institute of Communications and Computer Systems
ACCI	Acciona Construcción S.A.
OLO	Olympia Odos Operation S.A.
UGE	Université Gustave Eiffel
ETHZ	Eidgenössische Technische Hochschule Zürich
ROB	Robotnik Automation
CORTE	Confederation of Organisations in Road Transport Enforcement
STWS	SATWAYS - Proionta Kai Ypiresies Tilematikis Diktyakon Kai Tilepikinoniakon Efarmogon Etairia Periorismenis Efthinis EPE
RISA	RisaSicherheitsanalysen Gmbh
INAC	InnovActs
IKH	Ainoouchaou Pliroforiki SA -IKnowHow-
RG	Resilience Guard Gmbh

Glossary of Terms

Table 3: Glossary of terms

Term	Explanation
Visual servoing	Visual servoing, also known as vision-based robot control and abbreviated VS, is a technique which uses feedback information extracted from a vision sensor (visual feedback) to control the motion of a robot.

Executive Summary

This deliverable is written in the framework of WP3 – AI-based algorithms and tools Recognition, Classification and Localisation of the Points of Interest of the HERON project under Grant Agreement No. 955356. Deliverable 3.1, namely "AI - driven image segmentation and feature extraction", provides a detailed description of the data flow that derives from the various sensors of the HERON system for the purpose of the efficient utilization of the RGB data in order to extract semantic information for the recognition, classification, localization, and segmentation of the points of interest (e.g., cracks, potholes, and blurred road markings). This report illustrates the outcomes of Task 3.1, titled: "AI-driven image segmentation and feature extraction" corresponding to M4-M12 of the HERON project's period.

To this end, the document presents the training, application, and evaluation process of advanced deep learning algorithms such as CNNs for classification, YOLO detector (You Only Look Once) for object localization, and FCNs (e.g., U-Net model) for image semantic segmentation and modeling towards identification, classification and localization of the PoIs. In particular, specific computer vision toolkits have been developed for feature representation of the HERON maintenance and upgrading tasks, such as patching potholes, replacement of RUP (removable urban pavement) elements, sealing cracks, painting blurred road markings, as well as dispensing and removing traffic cones in an automated and controlled manner. The methodology framework will address all identified categories of RI degradation so that HERON's AI system is able to effectively initiate and guide, coordinate, and evaluate the various road maintenance procedures.

1 Introduction

1.1 Purpose of the Document

The specific document contains D3.1 “AI - driven image segmentation and feature extraction”. More specifically, D3.1 is the first deliverable within WP3, namely “AI-based algorithms and tools Recognition, Classification and Localisation of the Points of Interest” of the HERON project and it is a compilation of the work that was completed in the framework of task 3.1 “AI-driven image segmentation and feature extraction”.

The objective of this task is to apply state-of-the-art deep machine learning algorithms, such as CNNs and FCNs, for object detection, image semantic segmentation, and modeling toward recognition, classification, and localization of the PoIs. In particular, specific artificial intelligence toolkits that utilize optical data from various sensors (e.g., RGB, stereo cameras) have been developed for feature representation of the various HERON maintenance and upgrading tasks, such as for instance, potholes, blurred road markings, and cone detection as well as crack localization and segmentation.

Thereby, in this report, the conceptual AI monitoring framework and the related information regarding the RI degradation detection are demonstrated and analyzed in detail. Furthermore, the optimal combination of the remote sensing platform and sensors (i.e., RGB, stereo cameras mounted on the UGV and/or UAV) is demonstrated.

The remainder of this document is organized as follows: Initially, Section 2 discusses the scheduled inspection procedures, while Section 3 presents the sensing and hardware specification. Subsequently, Section 4 discusses and analyzes in detail the employed deep learning algorithms of the system, for the efficient classification, detection, and detection of the various defects of the road infrastructures. Lastly, Section 5 concludes this report.

1.2 Intended Audience

The specific deliverable report is public and therefore can be accessed by any interested stakeholder. Envisioned stakeholders involve, amongst others, the HERON end users. In particular, these are road operators, who are agents in the monitoring procedure as well as monitoring information consumers. Other envisioned stakeholders could be those interested in consuming tracking information to develop information products. These may incorporate risk and health assessment modules that need monitoring feedback and information in order to provide up-to-date hazard, risk, and vulnerability assessments.

1.3 Interrelations

The outcomes of HERON deliverables D2.1 and D2.2, namely “End-user needs and KPIs report” and “Architecture specification” respectively, serve as the guiding principles while composing the specific document. In particular, D2.1, which is related to the users’ requirements, contains the analysis of current practices, needs, and expectations from infrastructure stakeholders. In parallel, D2.2, which is related to the system architecture and design, includes the specifications of the HERON platform architecture, guidelines, and toolset

for development activities. Therefore, the two aforementioned deliverables directly involve the challenges and limitations that the AI monitoring framework of the HERON system should analyze and overcome, in order to efficiently facilitate the RI maintenance procedures.

As can be observed in Figure 1, the fundamental AI-based toolkits that are presented in the present deliverable interact with all the rest technical and development WPs, such as the automation and control technologies (WP4), the design and construction of the automated UGV system (WP5), and the back-end system that will support the decisions from the road operators and managers (WP6). Subsequently, all these outcomes combined will feed the activities of WP7 for the assessment and validation of the HERON solutions in all the demonstration sites.

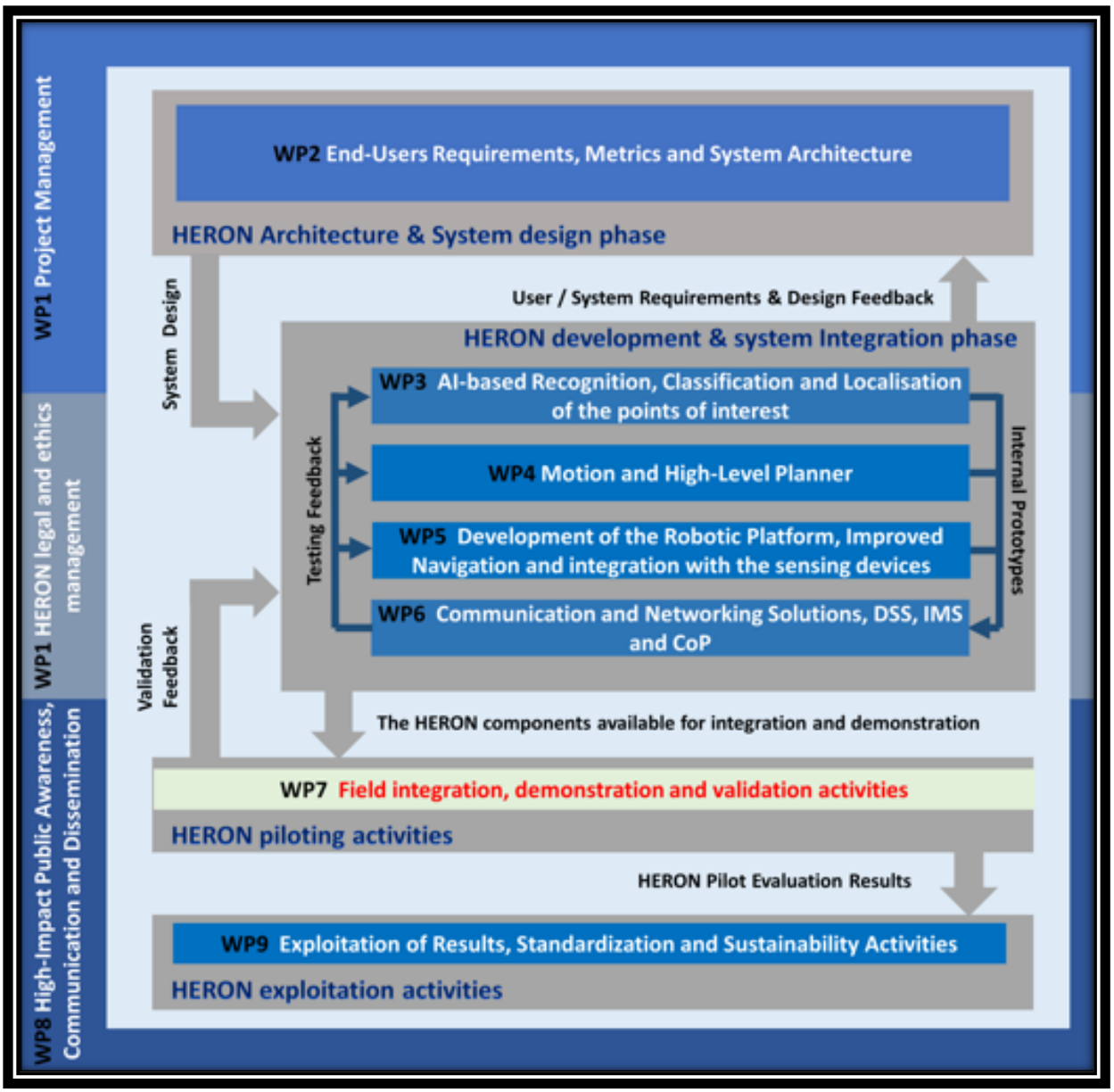


Figure 1: Interrelation between the various WPs of the HERON project.

2 Scheduled Inspection Procedures

Deterioration and defects on pavement lead to skidding, driving off tracks, improper maneuvering to avoid the road defects, and also prolonged driver braking distance which needs serious attention by traffic authorities. Besides that, poor surface macrotexture and microtexture could lead to hydroplaning and inconsistent tire pavement contact and also a reduction in tire gripping the pavement which eventually causes accidents. Any of the following applies to the specific category: roughness, rutting (deformation), potholes, and blurred road markings.

Automated methods for visual inspection based on image processing and machine learning techniques have been applied in various infrastructure monitoring applications cases including roads [34], pavements [10], bridges [13], and tunnels [42]. In most cases, deep learning approaches serve as the core mechanism and in more specific scenarios, DL is coupled with knowledge-based post-heuristics, to boost the detection capabilities of the models [43].

To this end, the (semi-) automated HERON system relies on improved intelligent control of a multi-degree-of-freedom (MDOF) robotized vehicle, improved computer vision (CV), and Artificial Intelligence (AI)/Machine Learning (ML) techniques combined with proper sensors (see Figure 2), decision-making algorithms and AR components to perform corrective and preventive maintenance and upgrading of roadworks is considered an advanced solution, which pushes routine roadwork activities quite beyond the state-of-the-art. At the same time, by using advanced data coming from various sources (including V2I and aerial drone surveillance) and well-established methods (from existing know-how from research and industrial projects), the automated system will be able to provide some non-routine (emergency) maintenance operations when required.

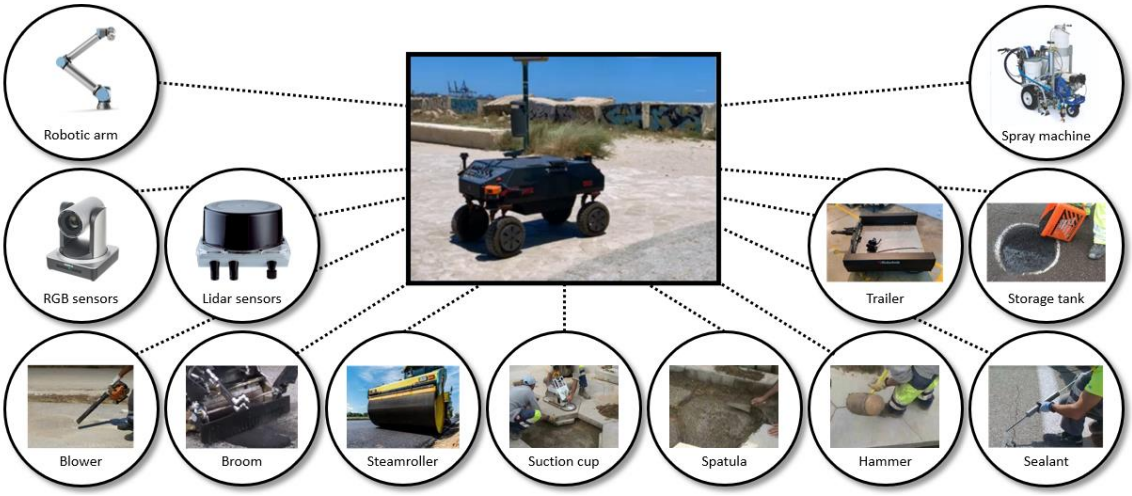


Figure 2: From traditional tools to robotic sensors and actuators.

Towards that direction, HERON targets the development and prototype validation of an innovative, automated intelligent robotic platform that utilizes state-of-the-art computer vision techniques for performing the necessary maintenance tasks safely, promptly, reliably, and modularly (see Figure 3). In this section, the employed inspection procedures and the mission planning are presented in the context of the three HERON pilot sites.

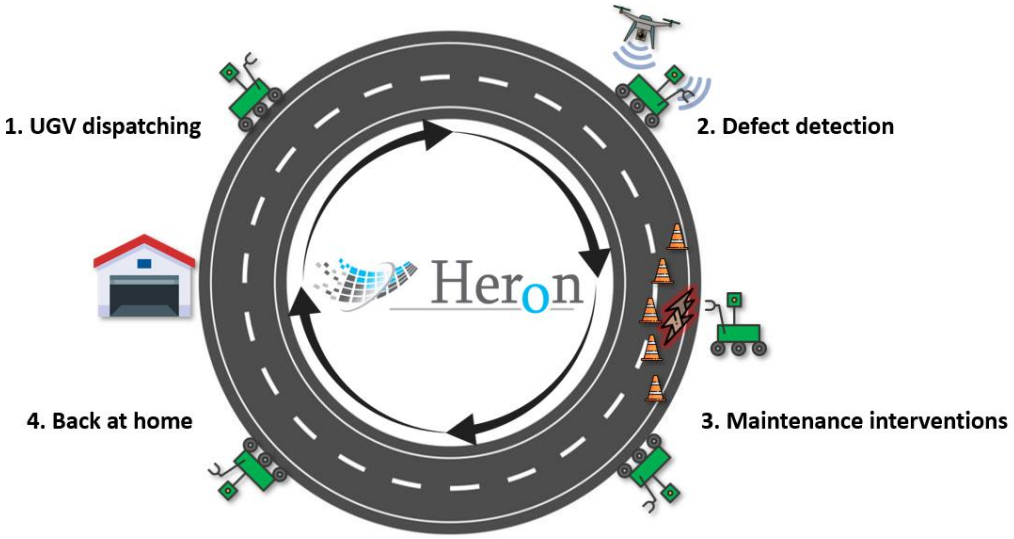


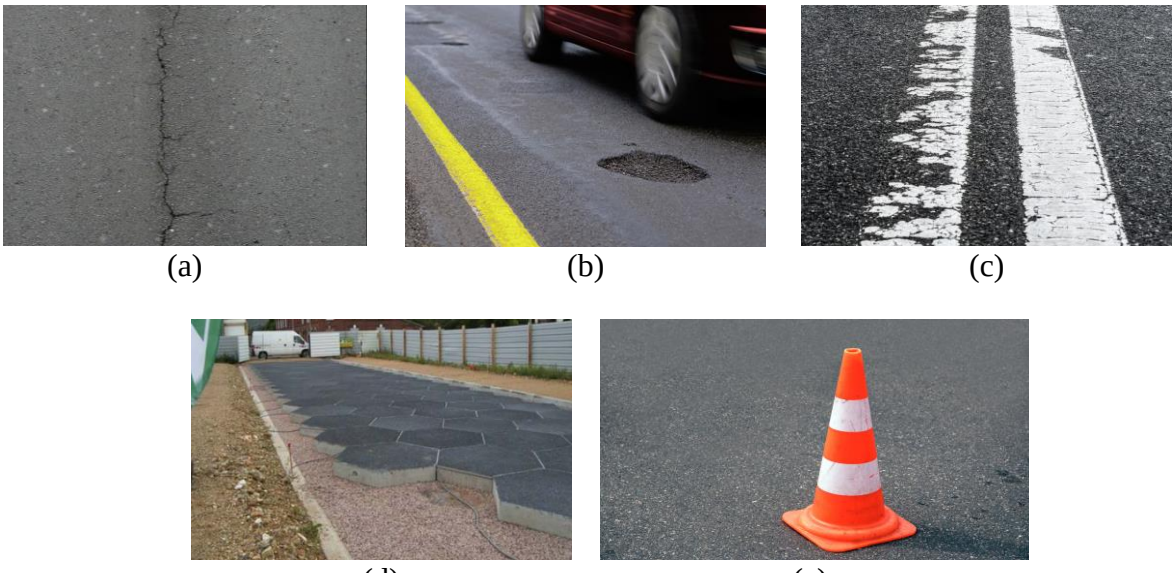
Figure 3: HERON's concept.

2.1 Employed inspection processes and mission planning

2.1.1 Monitoring objects

HERON AI framework will apply advanced DL algorithms such as for instance CNNs for modeling towards recognition, classification, and localization of the PoIs and FCNs for image semantic segmentation. In particular, specific DL toolkits will be developed and be analyzed in the following sections for feature representation of the HERON maintenance and upgrading tasks can be summarized in the following:

- Crack features (see Figure 4a)
- Pothole features (see Figure 4b)
- Road marking features (see Figure 4c)
- Removable urban pavement (RUP) features (see Figure 4d)
- Traffic cone features (see Figure 4e)



(a) (b) (c)

(d) (e)

Figure 4: Employed inspection processes.

Cracks

Pavement cracks are distresses of the pavement that occur on its surface. There are different types of pavements that create different cracks. The type of each cracking is highly related to the climate and traffic. The most common factor of crack pavement is moisture. Once moisture enters the pavement, it causes it to break down. In particular, moisture removes the sand and gravel from the base. As a result, the asphalt surface breaks or cracks. Therefore, pavement maintenance is required. The pavement restoration procedure includes the purification of the crack and the implementation of the material in order to fill the crack.

Potholes

Potholes are defined as depressions in the asphalt pavement. Potholes are usually caused by water that weakens the underlying soil structure and traffic that breaks the asphalt surface which is in poor condition. As a result, both asphalt and underlying soil material are removed, creating a hole in the pavement. Other reasons that cause potholes include insufficient pavement endurance to support traffic during extreme weather periods and lack of maintenance on the pavement. Pothole restoration procedures include the cleaning of the pothole, the implementation of the material to fill the pothole and the leveling of the material.

Road surface markings

Road surface marking is mainly material that is used on a road surface in order to provide official information to the drivers. They are usually placed with road marking machines. They can also be applied in other situations to mark parking spaces or define areas for other uses. Due to technological development, there is a big effort to improve the road marking system at a low cost. Today, road markings are used to communicate a wide range of information to the driver regarding safety and enforcement issues. This leads to their use in the road environment through advanced driver-assistance systems. As a result, the development of autonomous road vehicles is considered in the future.

Removable urban pavement

A removable urban pavement (RUP) is a pavement that can be opened and closed quickly by using lightweight equipment. Their main purpose is to provide easy access to underground networks. The idea of this concept comes from certain military paths or industrial soils. However, these pavements demand frequent work, even after construction or maintenance. Therefore, RUP causes major disturbance to the human environment, causing noise, air pollution and traffic congestion. In addition, RUP might have a negative impact on the street's architectural harmony.

Traffic cones

Traffic cones are most commonly cone-shaped markers that are placed on roads or footpaths to provide effective road management during road operations or automobile accidents. Traffic cones are also used to advance warning of hazards or dangers, or the prevention of traffic. Traffic cones are also used when children are playing or to block off an area. During night-time or low-light situations, traffic cones are used to increase the visibility of road limits. In some cases, traffic cones may also be fitted with flashing lights for the same reason. Cones are easy to move or remove. However, when purchasing traffic cones, it is important to be aware of safety standards and follow the appropriate guidelines in each area.

2.1.2 Monitoring impact

The development of the HERON's AI framework will have an impact on three fundamental areas: technical, economic, and occupational safety and health (OSH) administration.

At a technical level, HERON will showcase that it is possible to make a fully functional and efficient (semi-) automated system composed of commercial machines that work in a fully integrated manner, using the latest technologies in localization, navigation, AI/ML, comprehensive planning, decision-making, and automated manipulators to carry out complex maintenance and upgrading roadworks.

In parallel, economically, once the HERON technology will be technically proven and widely accepted by both road authorities and the public, then it will reach a point of commercialization. HERON is expected to have an impact in reducing the personnel needed to perform on-site roadworks. These people could be assigned to perform other tasks in the same company, which would reduce overall execution times or carry out a greater number of activities. Specifically, focusing on actions on the roadworks, a summary of the estimated reduction in resources per task is shown in Table 4. This local reduction of workers (although the total number will remain the same in the company) will make it possible to amortize the technological investment of the systems included in machines and their integration, generating economic savings from year "n" of amortization and holding jobs for people.

Table 4: Summary of the expected reduction of resources due to the usage of the HERON system.

Task	Workers needed conventionally	HERON estimated number of workers	HERON impact
Sealing cracks and patching potholes	3 or 4 in-situ	1 supervisor of the UGV assisted by 1 drone pilot (if needed)	33% to 50%
Painting road markings	2 or 3 in-situ		Up to 50%
Dispensing and removing traffic cones	2 or 3 in-situ		Up to 50%
Visual inspections	2 or 3 in-situ		Up to 50%

Lastly, at the occupational health and safety (OSH) level, HERON allows fewer people to work on a road while there is still traffic on the rest of the lanes, which reduces the chances of being run over in the event of an accident. On the other hand, the number of people who are in contact with hot substances that emit toxic fumes (e.g., bitumen and paints) will be reduced to a minimum level.

2.1.3 Monitoring strategy - mission planning

Regarding the monitoring strategy and planning as well as data acquisition of the HERON's sensing interface and AI component, as can be observed in Figure 3, the following steps below must be performed:

1. The UGV has to know in advance the rough location (with an accuracy of a few centimeters) of the RoI (e.g., crack, pothole, blurred road marking) by acquiring and utilizing knowledge (e.g., GPS coordinates) from an available source (e.g., the Operator specifies the location or a UAV localizes the damage of the road surface).
2. The UGV is activated and approaches the specified maintenance location.
3. The UGV examines the area by utilizing the mounted sensors in order to precisely localize and possibly segment the defected region of the RI.

4. The Operator provides the final approval and possibly calibrates the maintenance details, based on the feedback (i.e., detection of the RoI) received by the UGV sensors.
5. The UGV initiates the maintenance procedure (see Figure 5) by dispensing traffic cones on the road in an automated and controlled manner.
6. The UGV implements the maintenance work (see Figure 5) by constantly receiving data in real-time and re-evaluating the process, based on the mounted sensors.
7. The UGV finalizes the maintenance process (see Figure 5) by localizing and removing the traffic cones from the road in an automated and controlled manner, and then the Operator finalizes the mission and recalls the UGV to its base.

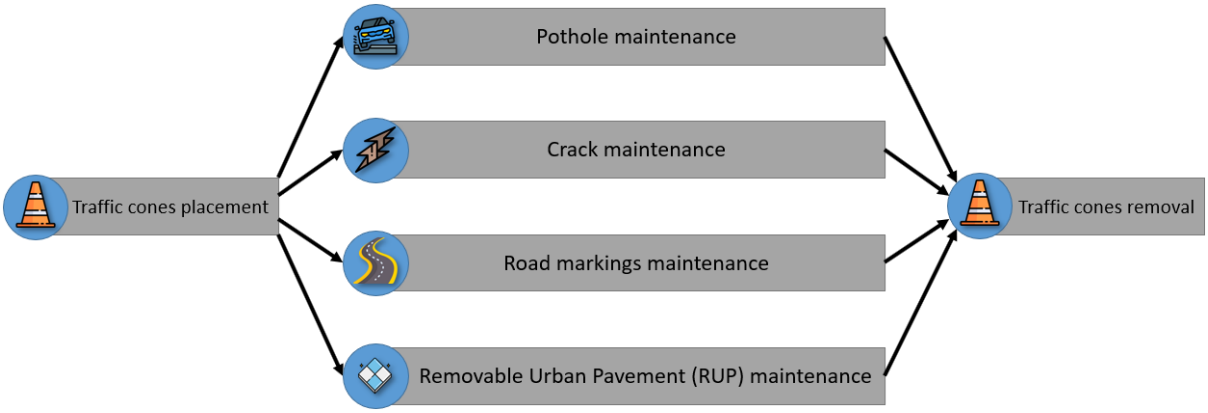


Figure 5: Identified use cases of the HERON project and mission planning of the maintenance process.

It is noted that regarding steps 5-7, additional information can be found in D2.1: End-user needs and KPIs report. Regarding the aforementioned objects (such as for instance, as depicted in Figure 5, cracks, potholes, blurred road markings, and traffic cones) that the AI component will be able to recognize, classify, and localize, Table 5 presents an overview of the proposed monitoring strategy which will act as the flagship of HERON's AI system by initiating, guiding, coordinating, and evaluating the road maintenance process.

Table 5: Monitoring strategy and technique for each of the identified use cases of the HERON project.

Object to be identified	Sensor	Processing technique	Monitoring information
Crack	• Optical camera on the UGV • Optical camera on the UAV • Stereo camera on the UGV	• Image processing • Deep learning • Object detection • Semantic segmentation	• Location • Size • Depth • Type (if needed)
Pothole	• Optical camera on the UGV • Optical camera on the UAV • Stereo camera on the UGV	• Image processing • Deep learning • Object detection	• Location • Size • Depth
Blurred road marking	• Optical camera on the UGV	• Image processing • Deep learning • Object detection	• Location • Size
Traffic cone	• Optical camera on the UGV • LiDAR (if needed)	• Image processing • Deep learning • Object detection	• Location

In particular, as illustrated in Figure 6A, regarding the pothole maintenance task, through the precise localization of the defect by the AI component utilizing RGB and stereoscopic data from the mounted sensors, the UGV will be able to:

i. clean the correct part of the road of dust and debris,
ii. calculate the depth of the pothole, by utilizing the input of a mounted stereo camera, in order to calculate the amount of the material that has to be poured,
iii. be positioned in the right spot in order to:
    a. fill the pothole with patching material,
    b. level the material, and
    c. compact the material.

Furthermore, as presented in Figure 6B, concerning the RUP replacement procedure (which is further described in Section 2.2.2), through the accurate detection of the slab as well as the potential defect on it by the AI component utilizing RGB data from the mounted sensors, the UGV will be able to:

i. lift the defected slab by using the robotic arm,
ii. place the new non-defected slab by using the robotic arm,
iii. be positioned in the right spot in order to:
    a. manipulate the sand-like material,
    b. possibly remove any debris (e.g., stones) that are present in the area,
    c. possibly fill the region with extra sand-like material, and
    d. level the replaced slab.

In parallel, as depicted in Figure 6C, regarding the crack maintenance process of the road infrastructure, through the localization and pixel-wise segmentation of the defect by the AI component utilizing RGB and stereoscopic data from the mounted sensors, the UGV will be able to:

i. clean the correct part of the road of dust and debris,
ii. calculate the depth of the crack, by utilizing the input of a mounted stereo camera, in order to calculate the amount of the material that has to be poured,
iii. guide effectively the robotic arm (visual servoing) in order to correctly apply the sealant material to the crack faces, and
iv. constantly reposition itself along the entire length of the crack defect.

Subsequently, as for the blurred road marking painting process (see Figure 6D) through the precise identification of the faded road line by the deep learning framework that is fed with RGB data deriving from the optical sensors, the UGV will be able to:

i. repaint the correct (deteriorated) part of the road marking,
ii. constantly reposition itself along the entire length of the road marking.

Lastly, as shown in Figure 6E, the effective traffic cone automated localization through the deep network that utilizes optical data from the mounted sensors will result in the HERON system being able to:

i. dispense the cones on the road in an automated and controlled manner,
ii. remove the cones from the road in an automated and controlled manner, and
iii. assist the fully autonomous operational and safe navigation and positioning of the UGV.

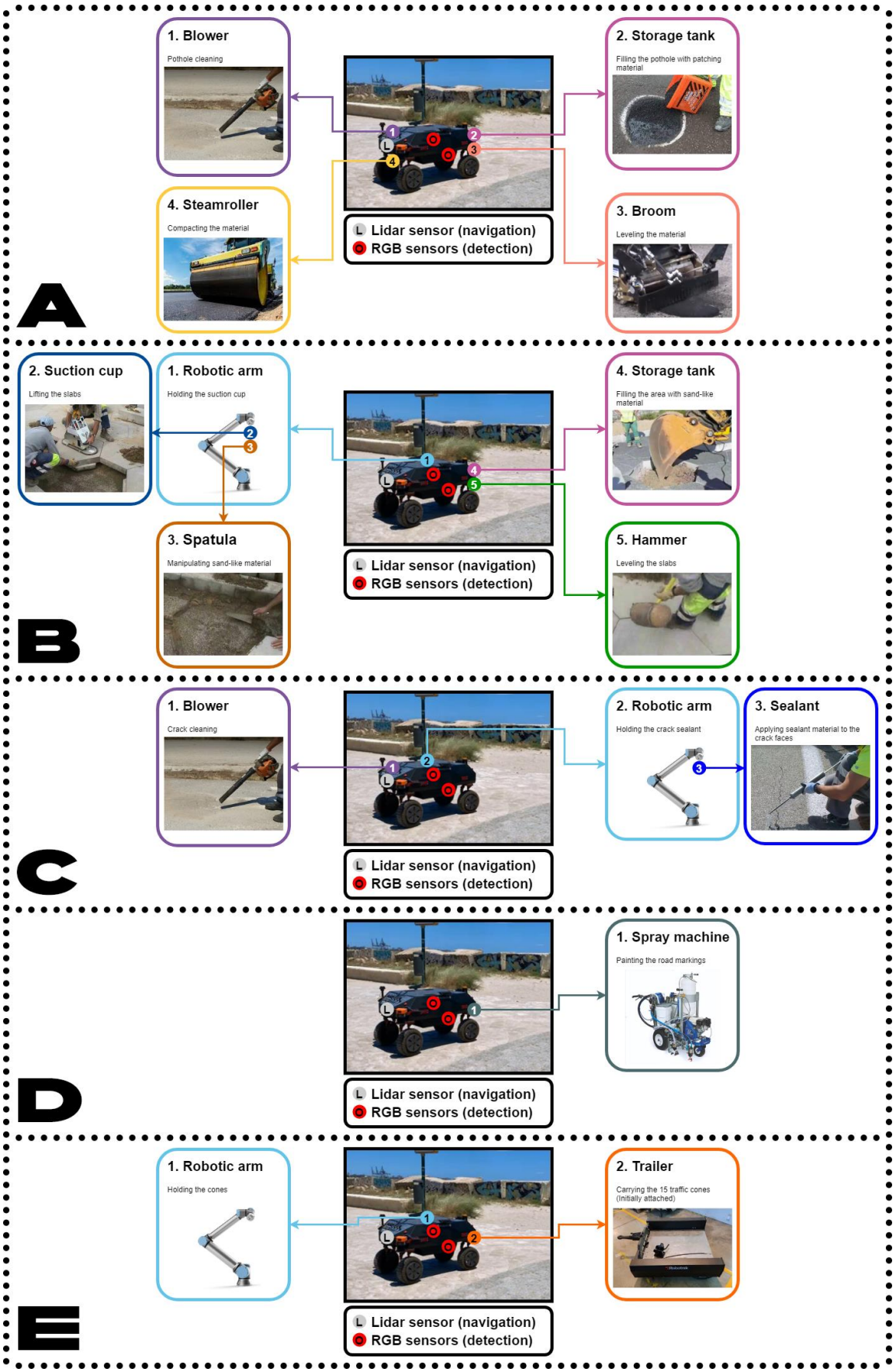


Figure 6: The main maintenance procedures that will be performed by the HERON system, i.e., (A) patching potholes, (B) replacement of RUP elements, (C) sealing cracks, (D) painting blurred road markings, and (E) dispensing and removing traffic cones in an automated and controlled manner.

2.2 HERON pilot sites

HERON will perform extensive tests in three demonstration sites, in Spain, France, and Greece. The adoption of three separate pilots instead of one to demonstrate the whole system and all of its components is selected since in real-world maintenance scenarios the system to be implemented will be directed to the indicated PoIs along with the RIs and will be targeted to specific applications that are to say “seal a crack” or “patch a pothole” or “CUD-feature”, etc. plus the assisting roadworks (e.g., spraying, put/remove cones, etc.). Different scenarios can be included in the same pilot test and executed in a subsequent manner. The specific section will underline the features and monitoring needs of the HERON end-users that have been identified in the deliverable report D2.1. For a more in-depth description and analysis of all end-user needs, refer to the corresponding documentation.

2.2.1 Spanish pilot (ACCI)

The pilot will be deployed in the A2 Motorway stretch (see Figure 7) maintained by the company (R2–CM42 stretch, coming from Madrid, and finishing in the limit between the provinces of Guadalajara and Soria, Spain), and in the traffic control center of the stretch located near the village of Torija. The motorway is owned by the Spanish National Road Authority and the section selected has a length of 77.5 km. The section has 4 lanes (2 per traffic direction) and crosses a region with Continental-Mediterranean climate, with long and severe winters, long, dry and hot summers and high heavy traffic levels, so the pavement is exposed to severe requirements and maintenance is crucial to preserve the optimum pavement conditions required. A2 is one of the main motorways in Spain, connecting Madrid with Barcelona, it is part of the Trans-European Transport Network (TEN-T) and the CEF corridor.

Figure 7: UAV images of A2 showing the typical maintenance after cracks sealing and patching.

The Toraja's traffic control center is in charge of monitoring the motorway status, visualizing and assessing the data provided by CCTV, inductive loops, GPS-based fleets, weather stations, weigh in motion systems, etc. It is also the basecamp for all assets needed for maintenance (e.g., machinery) and can be the ideal location for the preliminary trials of the different functionalities and robotic abilities developed in HERON.

The tailor-made image processing system for visual drones and sensing and CV installed in the robotic vehicle will be validated using real video and images and correlated with the information included in the existing road project and gathered during regular visual inspections and general patrolling.

After the initial validation of the automated vehicle and maintenance and upgrading functionalities, if the traffic authority permits the use of automated vehicles, a full-scale trial in a controlled stretch with certain defects and maintenance/upgrading needs will be carried out counting on the support of the Spanish National Roads Authority.

2.2.2 French pilot (UGE)

Transpolis (see Figure 8) is a proving ground of more than 80ha, which has been created by 5 entities among which UGE and which has been opened officially in 2019. It is typically used to test autonomous vehicles in a secure and controlled environment, also by assessing the V2I communication possibilities (several types of Road-Side- Units- RSUs- and communication means) are already installed on-site. It also is composed of several kilometers of road and all reinforced concrete buildings. Many types of V2X and I2V (Infrastructure to Vehicle) communication means are available, as well as camera monitoring, all of them will be used during the HERON activities.

Figure 8: The demonstration sites to be offered by UGE in France.

Another experimental site (see Figure 9), part of the French project R5G (Route de 5ème generation, the French declination of the European Forever Open Road programme), is proposed: the site LaVallée, also called E3S, is an urban development project at the former place of Ecole Centrale de Paris where several new mobility infrastructures and services will be implemented to create an evolutive, energy-neutral and cooperative road. In particular, the concept of urban removable pavement will be studied, using hexagonal concrete slabs prefabricated. These removable tiles allow quick access to networks, improve the durability of surface properties of roadways and can be recycled. Their prefabrication should make it possible to offer other integrated functions (various textures, porous, silent or depolluting surfaces, insertion of sensors, etc.). Currently, this CUD concept is not fit for TEN-T traffic, so inspecting these tiles regularly for cracks and repairing them is crucial. The damages are spalling at the interfaces between CUD elements and cracks among them. Currently, their initial installation and their replacement are done using motorized arms piloted manually.

Figure 9: Urban Pavement for smart city planning: the pre-fabricated road.

2.2.3 Greek pilot (OLO)

OLO has undertaken the traffic management and routine maintenance of the Elefsina-Korinthos- Patra motorway (in the heart of the Greek highway networks), which has 202 km total length and includes more than 25 km of tunnels and a large number of bridges, culverts, and ancillary structures (see Figure 10). It includes corrective and preventive maintenance both

of civil works equipment and Early Equipment Management (EEM) of open roads and tunnels. OLO will provide a part of the motorway, where extensive tests of the automated vehicle can take place, issuing the necessary permits in cooperation with the relevant Authorities (Public Service and Traffic Police) and ensuring safety conditions both for the road users and the people working for the project. The area that will be examined during the pilot program is the ELKO section, which is a dual carriageway with three lanes (3.5m width left lane, 3.75m width middle and right lane) and an emergency lane (varies from 2.5 to 3.5m) per direction with concrete New Jersey safety barriers in the central axis of the motorway. Some major technical features of the ELKO section:

- Total length: 64 km
- Interchanges (I/C): 12
- Bridges: 16
- Tunnels: 5 (total length of 4,473 km).

Figure 10: Photos from the initially selected demonstration sites to be used from OLO, including tunnels (left), bridges, and interchanges (right).

3 Sensing and hardware specification

3.1 Sensors

To achieve recognition and localization of the various road defects and objects of interest, real-time 2D and 3D visual information will be needed. Most of the quoted Deep Learning models work with 2D RGB images, nevertheless, depth information is also required to deduce the relevant position of a detected object to the UGV's frame of reference, so it can proceed with the mending tasks.

For this purpose, Zed 2i cameras will be used (see Figure 11 and Table 6). They will be mounted circumferentially to the UGV and due to their wide angle of view, the whole of the robot working space will be covered. Zed 2i is a stereoscopic camera, which means that it uses a pair of normal RGB cameras to perceive depth via triangulation, thus acquiring normal RGB images, together with depth information of each pixel.

Figure 11: Zed 2i - industrial AI stereo camera.

Table 6: Specifications of the Zed 2i.

General Specifications	
Output Resolution	Side by Side 2x (2208x1242) @15fps 2x (1920x1080) @30fps 2x (1280x720) @60fps 2x (662x376) @100fps
Interface	USB Type C - External cable (up to 10m)
Baseline	12cm (4.72 in)
RGB Sensors	Dual 1/3" 4MP CMOS 2688 x 1520 pixels 2µm x 2µm Rolling shutter YUV 4:2:2 - UYV (8bits)
Motion Sensors	Gyroscope, Accelerometer, Magnetometer
Environmental Sensors	Barometer Temperature
Warranty	2-year hardware warranty
In the Box	ZED 2i stereo Camera 1.5m long USB Type-C cable

Physical	
Dimension	175.25x 30.25x 43.1 mm (6.90 x 1.69")
Weight	166g (0.36 lb.)
Operating Temp.	-10°C to +45° (14°F to 113°F)
Power	380mA / 5V USB powered

System Requirements	
GPU	NVIDIA GPU ≥ 2GB Memory NVIDIA Compute capability ≥ 3.0 Compatible with: - NVIDIA Jetson Nano - NVIDIA Jetson TX2 - NVIDIA Jetson Xavier
CPU	Dual-core≥2.4GHz processor Minimum 4GB RAM
OS	Windows 10 - 64bit Ubuntu 16.04/18.04 - 64 bit Debian CentOS (via Docker) Jetson L4T

This camera covers all the needs of computer vision algorithms. Nevertheless, because of the wide-angle of view and the small imaging sensor that this camera has, as well as the slow shutter speed, some information and details of the road will be lost, which might affect negatively some of the detection tasks. If this is the case, a high-end industrial camera (MER2-2000-19U3C) is selected to be also deployed alongside Zed 2i, in case of detection accuracy is less than the target value (see Figure 12 and Table 7). This camera uses a Sony IMX183 sensor which has much fewer distortions that Zed2i, and more than double its resolution (5496×3672 instead of FHD).

Figure 12: MER2-2000-19U3C – scan industrial camera.

Table 7: Specifications of the MER2-2000-19U3C.

Interface	USB3
Resolution	5496×3672
Frame rate	19fps
Pixel Size	2.4uM
Color/Mono	Color
Sensor Type	Sony IMX183
Optical Size	1"
Shutter Type	Rolling Shutter
Shutter time	12us~1s
ADC Bit Depth	12bit
Pixel Bit Depth	8bit, 12bit
Digital gain	0dB~24dB
Pixel Data Format	BayerRG8 / BayerRG12
Synchronization	Hardware trigger, software trigger
I/O	1 opto-isolated input line and 1 opto isolated output line, 2 GPIO
Operating Temp.	0°C~45°C
Operating Humidity	10%~80%
Lensmount	C
Dimensions	29×29×29mm
Software	Windows / Linux / Android
Power Consumption	<2.7W@5V
Weight	65g
Conformity	RoHS, CE, FCC, USB3 Vision, Genicam

3.2 Processing components specification

There are multiple needs for processing power for each software module and the most important will be discussed. Firstly, Zed SDK requires both CPU and GPU processing power for the relevant triangulation processes, namely at least a Quad-core processor of 2,7GHz, 8GB

of RAM, and NVIDIA 1060GTX GPU. Secondly, the Deep Learning models described in this report require a recent dedicated NVIDIA GPU with CUDA cores, and at least 6GB of VRAM. The higher the VRAM is, the better DL detection capabilities will enable (12GB of VRAM will be enough to infer any size of the image through the neural networks). Lastly, the various pre and post-processing computer vision algorithms will need a powerful processor with multiple threads, like Intel Core i7-11700 and RAM of at least 16GB, as they will need to run in parallel with the robot navigation processes.

It is noted that in case of size and power constraints, the Deep Learning models as well as the other software can be optimized to run in an embedded device like Jetson (see Figure 13 and Table 8). However, there will be a trade-off between detection accuracy and processing power if the hardware capabilities are lower than the previously mentioned, thus an embedded device cannot be specified yet, because it must reach first the desired criteria.

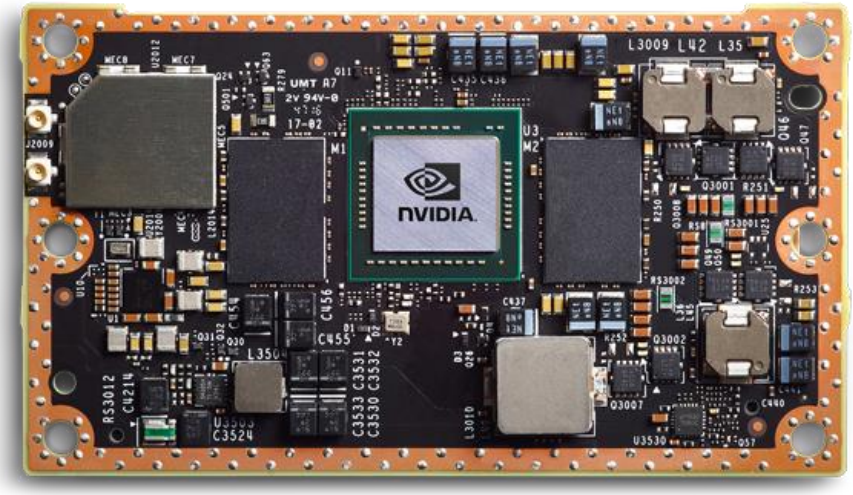


Figure 13: Jetson TX2 Module.

Table 8: Technical Specifications of Jetson TX2.

GPU	256-core NVIDIA Pascal™ GPU architecture with 256 NVIDIA CUDA cores
CPU	Dual-Core NVIDIA Denver 2 64-Bit CPU Quad-Core ARM® Cortex®-A57 MPCore
Memory	8GB 128-bit LPDDR4 Memory 1866 MHx - 59.7 GB/s
Storage	32GB eMMC 5.1
Power	7.5W / 15W

4 Deep learning algorithms

The traditional road inspection procedures, carried out by engineers/technicians, do not only pose safety risks, that may lead to worker injuries and accidents, but are also costly and time-consuming. Furthermore, they require heavy machinery that results in hindrances in road and traffic. In addition, it is clear that identification of the defected areas is performed using manual visual inspection, potentially leaving damage unnoticed in the inaccessible areas of the road infrastructures. Therefore, it is beneficial to adopt computerized methods, such as computer vision and machine learning, for efficient defect inspection and classification.

Indeed, precise localization and classification of defects overcome all the aforementioned limitations in inspecting defects on road infrastructures. Additionally, they trigger the novel concept of automated maintenance [31] by (i) precisely assessing the damage and extracting accurate measurements of it through the use of computerized methods, (ii) driving maintenance robots to repair the damage, especially in difficult to access areas, and (iii) stimulating the concept of prefabrication through which the repaired components are constructed off-site and then are transferred to the road infrastructure to be easily replaced.

Currently, there is a great research interest in automatic visual inspection of defects on road infrastructures, such as for instance cracks, potholes, and blurred road markings, by analyzing visual data. Generally, the deteriorated road surface produces rough surfaces (see Figure 14). For this reason, usually color distributions, in different color spaces, i.e., HSV, are utilized to detect road surface defects [22].

(a) Non-deteriorated road surface (b) Road surface with cracks (c) Road surface with potholes

(d) Non-deteriorated road marking (e) Blurred road marking

Figure 14: Non-deteriorated road surfaces (a, d) tend to present a more uniform distribution of colors compared to defected ones (b, c, e).

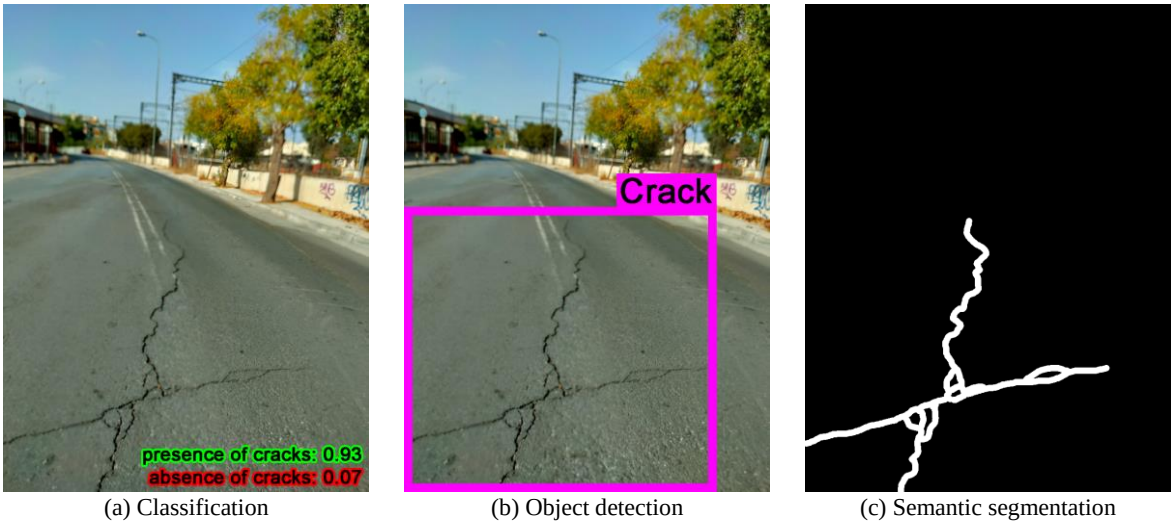
These distributions are represented through a mixture of Gaussian models or reconstructed using unsupervised machine learning schemes such as the k-means [28]. The key idea behind these methods is that the defected areas tend to present non-uniformity (see Figure 14a and d), in contrast to non-defected ones (see Figure 14b, c and e) where the colors are smoother and consistent [23]. In this context, image analysis based on color and textural characteristics can be exploited as an auxiliary tool, through which we are able to discriminate and quantify material deterioration more effectively [29].

Although these techniques are computationally efficient and easy to be implemented, even with a limited number of ground truth data, they fail to precisely localize the contours of the defected regions, and thus they are not suitable for robotic-driven maintenance and prefabrication procedures. Moreover, the defect detection performance also deteriorates in cases where the color properties of the non-defected regions are similar to the defected ones [22].

4.1 Computer vision tasks

Recently, with the progress of AI technology and deep learning, automatic visual inspection of road infrastructures is performed using Convolutional Neural Networks (CNNs) architectures on RGB data for road health monitoring [41]. The main advantage of such approaches is that they increase detection and classification accuracy compared to the color distribution modeling since ground truth (annotated) data are exploited throughout the learning process, making the CNN structures better identification architectures of defected regions.

In general, deep learning-driven defect detection (e.g., crack, pothole, and blurred road marking identification), is a computer vision problem that can be handled as a (i) classification, (ii) object detection, or (iii) semantic segmentation task. In the first approach, the deep learning algorithm provides a binary outcome with some metric (e.g., probability), with a positive outcome indicating the presence of at least one defected region in the image, and a negative outcome indicating that the whole road of the image is defect-free (see Figure 15a). In the second procedure, the deep learning model returns bounding boxes that localize and indicate the dimensions of every instance of defected areas (see Figure 15b). The last method involves pixel-based classification techniques that create segmentation masks, thus giving us a granular understanding of the shape of the defected surface (see Figure 15c).



(a) Classification (b) Object detection (c) Semantic segmentation

Figure 15: Comparison of the three different deep learning approaches, in the crack identification task.

Although the first approach is useful for visual inspection, as it indicates the presence of road damages in certain areas of the infrastructure, it cannot be suitable for automated maintenance since it cannot precisely localize and classify the defected regions. On the contrary, precise object detection (see Figure 15b) or pixel-wise classification (see Figure 15c) gives us detailed information related to various metrics, such as the area, maximum distance, aspect ratio, and shape of the defected region, useful for prefabrication, a practice through which we are able to design and fabricate individual precast components outside of the infrastructure. It is noted that prefabrication has several benefits such as reduced working risks, improved traffic flows, and (cost and time) maintenance optimization. For instance, it has been estimated that adopting prefabrication as a construction methodology could result in 70% time savings and 43% labor cost reductions [21].

Apart from a precise localization of a damaged region, a simultaneous classification of its defect type is required to properly simulate a maintenance procedure. Thus, we need to categorize a defected region into damage type (see Figure 16) which, in the sequel, trigger different maintenance actions from the robotic vehicle. To achieve, however, precise localization of a defected region, we need to move from local image processing, like the one deriving from the CNN deep learning models to a global-local one.

Local processing analyzes small size image patches independently and thus it may lead to misclassification mainly where similarities (color, texture, etc.) between the defected and non-defected regions are encountered. Instead, a global-local data processing first decomposes the image data into scales and then proceeds with a local analysis. Global-local data processing has been proposed for detecting regions of interest in medical data through the use of Fully Convolutional Networks (FCNs) [30] and their variants U-Nets [46]. Global features represent the whole image content, whereas local features are responsible for the precise localization of the region contours.

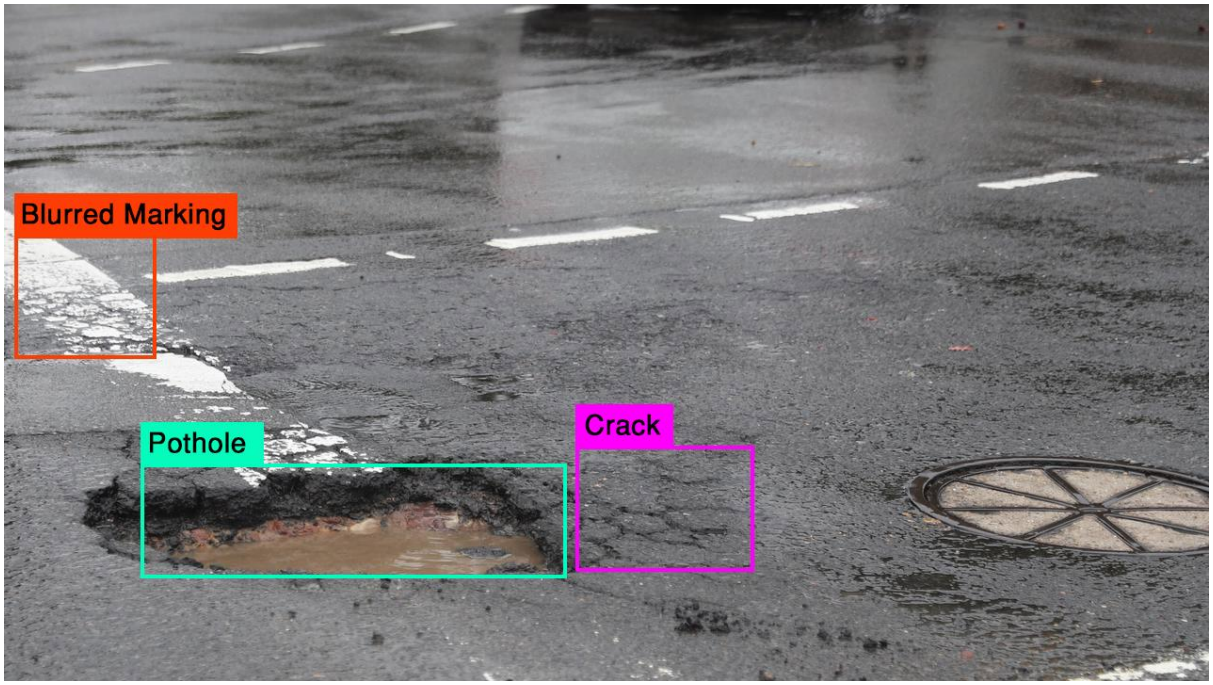


Figure 16: Representative photographic examples of the three road defects (i.e., crack, pothole, and blurred road marking) as well as their automated localization using a deep CNN classifier.

In a nutshell, as shown in Figure 14, automated road defect recognition can be distinguished between the following three computer vision tasks, which are further analyzed in the following sections below (see Sections 4.1.1-4.1.3):

- Image Classification: Predicts the class of an object (defect) in an image.
- Object Detection: Locates the presence of objects (defects) with a bounding box and classifies the located objects (defects) in an image.
- Object (or semantic) segmentation: Locates the presence of objects (defects) by highlighting the specific pixels of the object (defect) instead of a coarse bounding box.

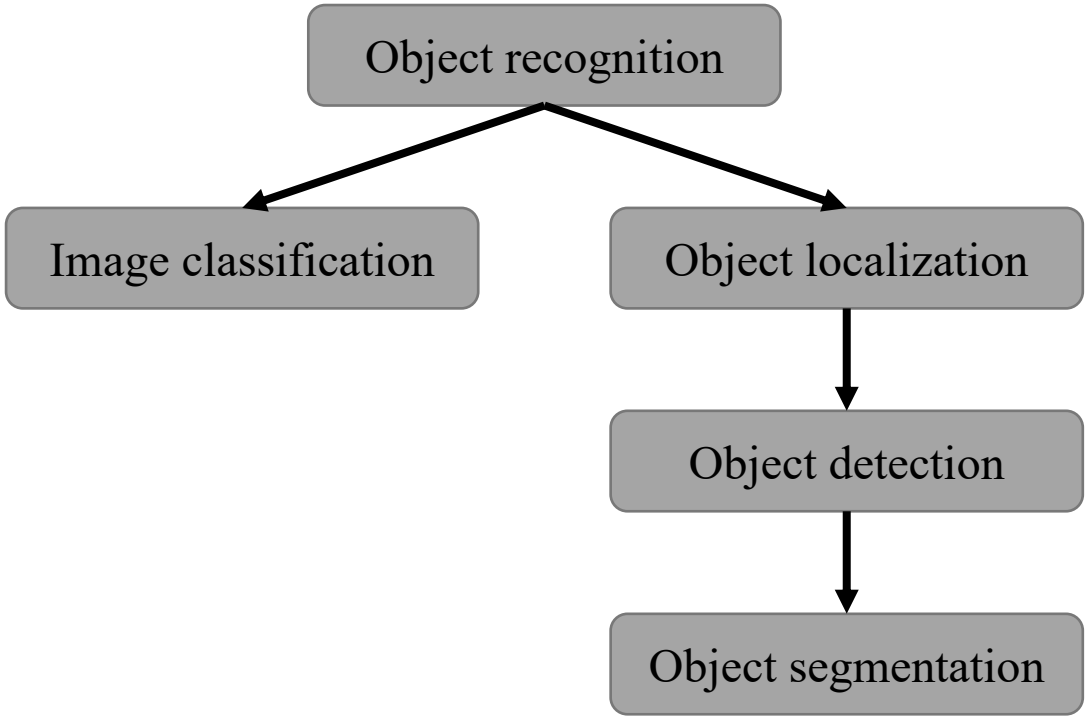


Figure 17: Overview of object recognition computer vision tasks.

4.1.1 Classification

Image classification, as well as object detection and image segmentation techniques, is a method highly related to the domain of computer vision. In particular, these techniques help machines understand and identify objects and environments in real-time via inputs such as images. Until today, computer vision techniques are highly used in several sectors, including healthcare, manufacturing, retail, etc. Despite the fact that both methods are used in object identification, there are many differences between them. In simple words, image classification is a technique that is used to classify or predict the class of a specific object in an image. The basic scope of this technique is to identify the features in an image.

In general, the image classification techniques are divided into either parametric and non-parametric or supervised and unsupervised as well as hard and soft classifiers. For supervised classification, this technique delivers results based on the input and output provided while training the model. On the other hand, the unsupervised classification technique provides the result based on the analysis of the input dataset. In this case, features are not directly fed to the models.

The main steps of image classification techniques include the identification of a suitable classification system, feature extraction, selecting good training samples, image pre-processing

and selection of appropriate classification method, post-classification processing, and finally assessing the overall accuracy. During this technique, the inputs are usually an image of a specific object and the outputs are the predicted classes that define and match the input objects. Convolutional Neural Networks (CNNs) is the most popular neural network model that is used for image classification problem.

Types of image classification techniques

The supervised image classification techniques include the parallelepiped technique, minimum distance classifier, and maximum likelihood classifier, among others. Several other types of image classification techniques as mentioned below. In particular, image classification can be based on the:

- information acquired from different sensors,
- nature of the training sample,
- basis of the various parameter used in data,
- nature of pixel information used in data,
- number of outputs generated for each spatial data element, and
- nature of spatial information.

Disadvantages of classification methods

In supervised and unsupervised image classification techniques, the disadvantages include the extensive amount of time required during the training phase and the fact they can't deal with big data. Moreover, the classification methods would output only a probability distribution of interested classes, and are not able to localize an object in a given image.

Image classification using a general multi-label CNN classifier

A general defect classifier is a multi-label CNN architecture. It is noted that a CNN consists of three main types of neural layers, namely (i) convolutional layers, (ii) pooling layers, and (iii) fully connected layers. Each type of layer plays a different role:

- **Convolutional layers:** In the convolutional layers, CNN utilizes various kernels to convolve the whole image as well as the intermediate feature maps, generating various feature maps. Because of the advantages of the convolution operation, several works (e.g. [36]) have proposed it as a substitute for fully connected layers with a view to attaining faster learning times.
- **Pooling layers:** Pooling layers are in charge of reducing the spatial dimensions (width x height) of the input volume for the next convolutional layer. The pooling layer does not affect the depth dimension of the volume. The operation performed by this layer is also called subsampling or downsampling, as the reduction of size leads to a simultaneous loss of information. However, such a loss is beneficial for the network because the decrease in size leads to less computational overhead for the upcoming layers of the network, and also it works against over-fitting. Average pooling and max pooling are the most commonly used strategies. In [9] detailed theoretical analysis of max pooling and average pooling performances is given, whereas in [48] it was shown that max-pooling can lead to faster convergence, select superior invariant features and improve generalization.
- **Fully-connected layers:** Following several convolutional and pooling layers, the high-level reasoning in the neural network is performed via fully-connected layers. Neurons in a fully connected layer have full connections to all activations in the previous layer, as their name implies. Their activations can hence be computed with a matrix

multiplication followed by a bias offset. Fully-connected layers eventually convert the 2D feature maps into a 1D feature vector. The derived vector could be either fed forward into a certain number of categories for classification [25] or could be considered as a feature vector for further processing [15].

As illustrated in Figure 18 the architecture of CNNs employs three concrete ideas: (i) local receptive fields, (ii) tied weights, and (iii) spatial subsampling. Based on the local receptive field, each unit in a convolutional layer receives inputs from a set of neighboring units belonging to the previous layer. This way, neurons are capable of extracting elementary visual features such as edges or corners. These features are then combined by the subsequent convolutional layers in order to detect higher-order features.

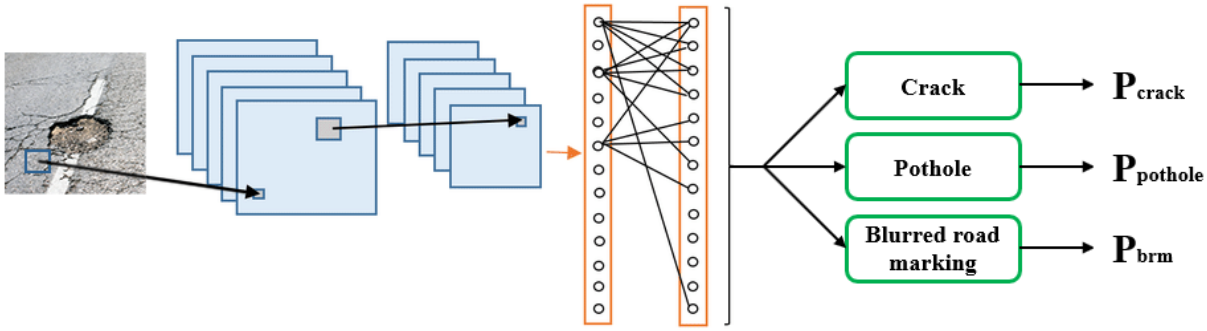


Figure 18: The architecture of a CNN model for multi-label defect detection.

However, as already mentioned in the previous section, though the classification approach is useful for a rough visual inspection of the road infrastructure, since it indicates the presence of road defects in certain areas of the highway, it cannot be suitable for automated maintenance since it cannot effectively localize in detail and classify the damaged areas (i.e., cracks, potholes, and blurred road markings). On the other hand, as presented in the next two subsections, an object detection task (see Section 4.1.2) or a precise pixel-wise classification (see Section 4.1.3) gives us detailed feedback related to the damaged area.

4.1.2 Object detection

The scope of object detection is to determine where objects are located in a given image such as object localization and which category each object belongs to, like object classification. In general, object detection is a type of image classification technique that also identifies the location of the object instances from a large number of predefined categories in images.

This technique can also search for a specific class of objects, such as cars, people, animals, etc. Moreover, object detection techniques are also used in the next-generation image as well as video processing systems. The recent advancements in this technique have only become possible through deep learning methodologies. Object detection techniques can be used in real-world projects such as face detection, pedestrian detection, vehicle detection, traffic sign detection, video surveillance, etc.

How object detection works

The pipeline of traditional object detection models can be mainly divided into three stages, i) informative region selection, ii) feature extraction, and iii) classification. There are several deep learning-based models for object detection, which have been used by big organizations in order to achieve efficiency as well as accurate results in detecting objects from images. The most popular models include MobileNet, You Only Look Once (YOLO), Mark-RCNN, RetinaNet.

Disadvantages of object detection methods

The main issue of object detection techniques involves poor performance when objects are placed in arbitrary poses in a cluttered and occluded environment.

Object detection using YOLOv5 (You Only Look Once) model

YOLO is a fast real-time multi-object detection algorithm, which was first outlined in 2015 [44] and since its first inception, many modifications have been proposed to improve and speed up the detection process. YOLO is an acronym for 'You only look once' and is a target detection algorithm based on a regression algorithm that uses Neural Networks to provide real-time object detection. Its usefulness comes due to the fact that it completes the prediction of the classification and location information of the objects according to the calculation of the loss function, so it transforms the target detection problem into a regression problem [27]. This algorithm uses the most advanced detection technologies available at the time and optimizes the implementation for best practice [14].

In this implementation, we utilize YOLOv5, which holds the best performance among YOLO algorithms. It is based on the PyTorch framework and its functionality comes from the fact that it is a suitable lightweight detector that can balance detection accuracy and model complexity under the constraints of processing platforms with limited memory and computation resources [55]. As can be seen in Figure 19 the architecture of the model YOLOv5 consists of three parts: (i) Backbone: CSPDarknet, (ii) Neck: PANet, and (iii) Head: YOLO Layer. The data are initially input to CSPDarknet for feature extraction and subsequently fed to PANet for feature fusion. Lastly, the YOLO Layer outputs the object detection results (i.e., class, score, location, size). The architecture of the model can be seen in Figure 19.

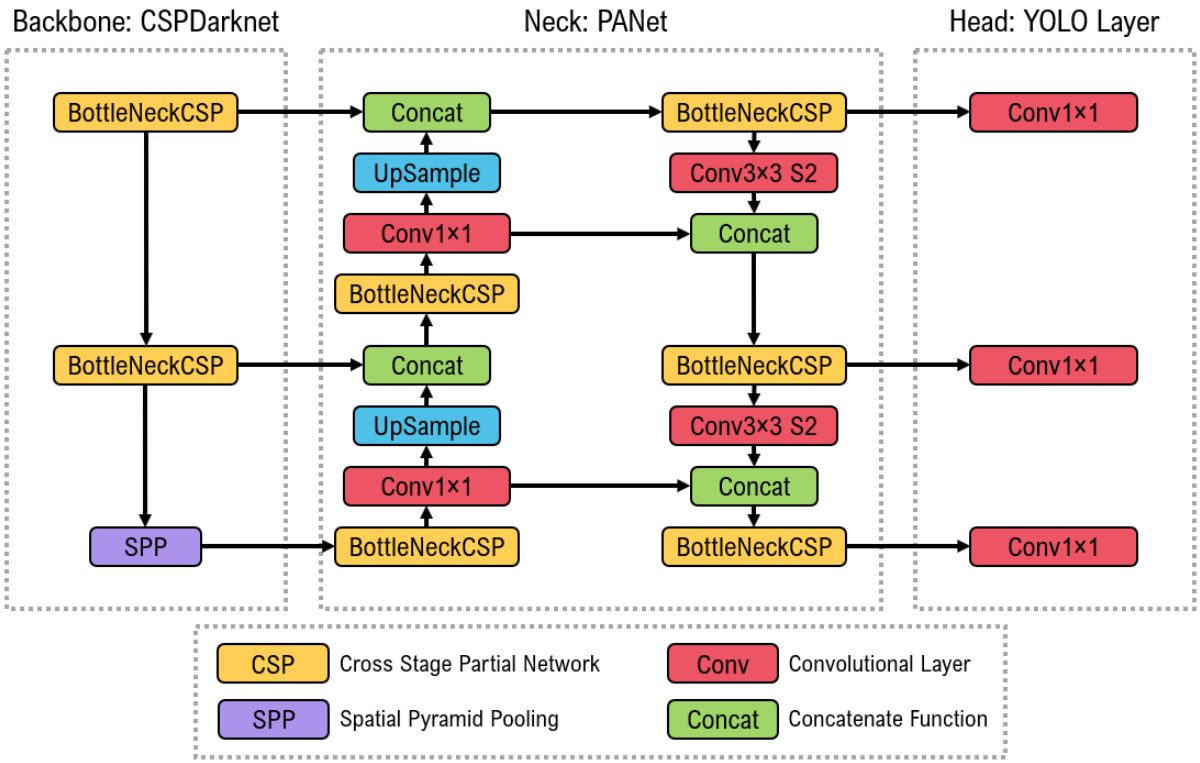


Figure 19: The architecture of the model YOLOv5, which consists of three parts: (i) Backbone: CSPDarknet, (ii) Neck: PANet, and (iii) Head: YOLO Layer. The data are initially input to CSPDarknet for feature extraction and subsequently fed to PANet for feature fusion. Lastly, the YOLO Layer outputs the object detection results (i.e., class, score, location, size).

4.1.3 Image segmentation

Image segmentation can be considered as a further extension of object detection. Through image segmentation, we can detect objects (as in object detection) via pixel-wise masks for each image. Therefore, image segmentation helps us gain a deeper understanding of the shapes/curves of objects. In addition, we can also define the class of each pixel in the image. Image segmentation can be specifically helpful if we want to have more information about each segment of an object.

Semantic segmentation

Semantic segmentation labels each pixel with its class label without differentiating about instances. In semantic segmentation, a model is trained in order to produce high-resolution semantic segmentation. On the one hand, semantic segmentation an encoder/decoder structure utilizes downsampling and upsampling. Downsampling produces lower-resolution feature mappings which help with differentiating the classes. On the other hand, upsampling produces a higher-resolution segmented image. Both techniques are ideal for simple object detection in an image. However, image segmentation provides more information about the image.

Semantic segmentation using a U-Net model

The simplest method is to localize the damage as a boundary box indicating areas of interest (see Figure 15b). Although such a boundary box detection approach assists maintenance engineers by accelerating the time of supervised inspections, especially when they are dealing with large-scale critical infrastructures (i.e., highways and national major roads), they fail to cope with the concept of prefabrication as well as of providing precise geometric measurements of the detected corrupted regions, useful for robotic-driven maintenance. Consequently, we require high-quality image information on a pixel level basis to determine the detailed shape of the cracked regions and to speed up the maintenance procedure of the road infrastructure. That can be achieved through semantic segmentation techniques, which aim to label each pixel of a given image with a corresponding class, thus providing masks of the cracked areas (see Figure 15c).

In the present section, we address the problem of crack detection as a pixel-based semantic segmentation task. In this context, apart from accurately identifying, localizing, and classifying the material deterioration of the road infrastructure, we can determine its exact shape on a pixel level accuracy, useful for prefabrication, precise measurement, and automatic (e.g., robotic-driven) maintenance.

In contrast to local data processing that characterizes for instance a sliding window CNN classifier [22], recently global-local data processing architectures have been utilized especially in medical imaging. An example of a classifier for global-local data analysis is the FCNs [30] (see Figure 20) including their variants U-Nets [46]. An FCN model consists of two main parts: (a) a convolutional encoder with the main purpose of transforming the whole image into different scales (global processing) and (b) a classification part that maps the scaled images into class categories (local processing). Global-local data processing increases the classification accuracy since it provides a multi-scale image analysis framework, instead of classifying image local patches in an independent way one from another. It is noted that the term “deconvolution” in Figure 20 describes the inverse convolution (InvConv) process. Furthermore, the max-pooling operation of the encoder is non-linear, and thus, there is not a direct inverse procedure. Lastly, in the downsampling path, the FCN generates “intermediate predictions” at different scales, which are then merged together during the upsampling process to yield the final segmentation mask.

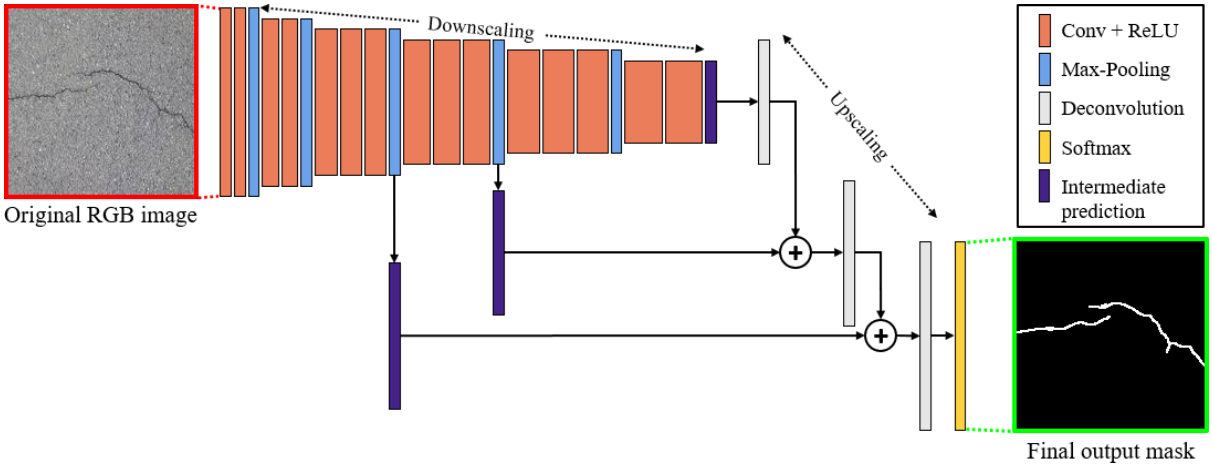


Figure 20: A schematic representation of the global-local analysis performed by a fully convolutional network model.

In this work, we utilize a global-local data processing framework for crack defect localization using a U-Net structure (see Figure 21). U-Net [46] is an FCN variant with small modifications, originally designed for biomedical segmentation problems. U-Net's main differences compared to an FCN are: (i) its structure is symmetric, and (ii) the skip connections between the downsampling and the upsampling path apply a concatenation operator instead of a sum. Such a skip connection modification, as well as the symmetrical organization of the U-Net model, allows for better global-local information exchange.

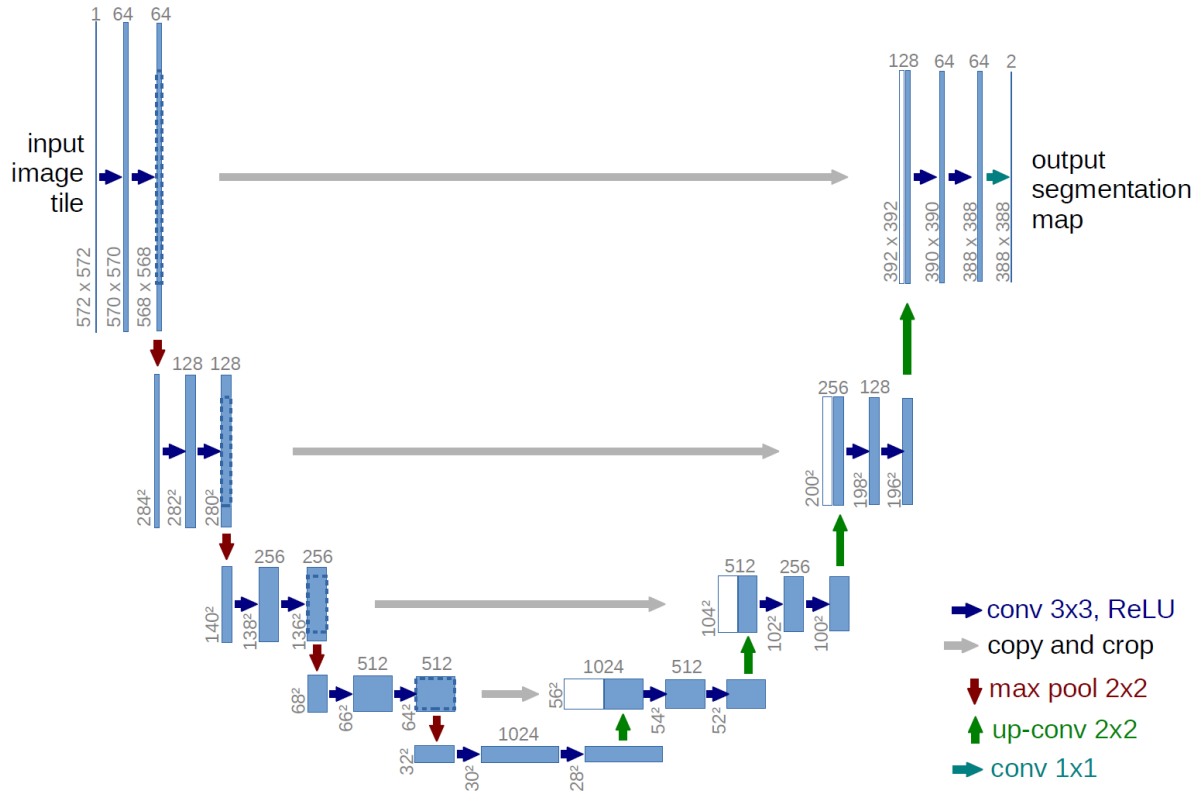


Figure 21: Architecture of the U-Net model presented in the work of [46].

4.2 Degradation and road defect types

As already mentioned, a variety of road defects and degradation are considered (e.g., cracks, potholes, blurred road markings). Regarding the inputs, analysis, and outputs of the sensing interface and AI component (see Figure 22) additional information can be found in the deliverable report D2.2: Architecture Specification.

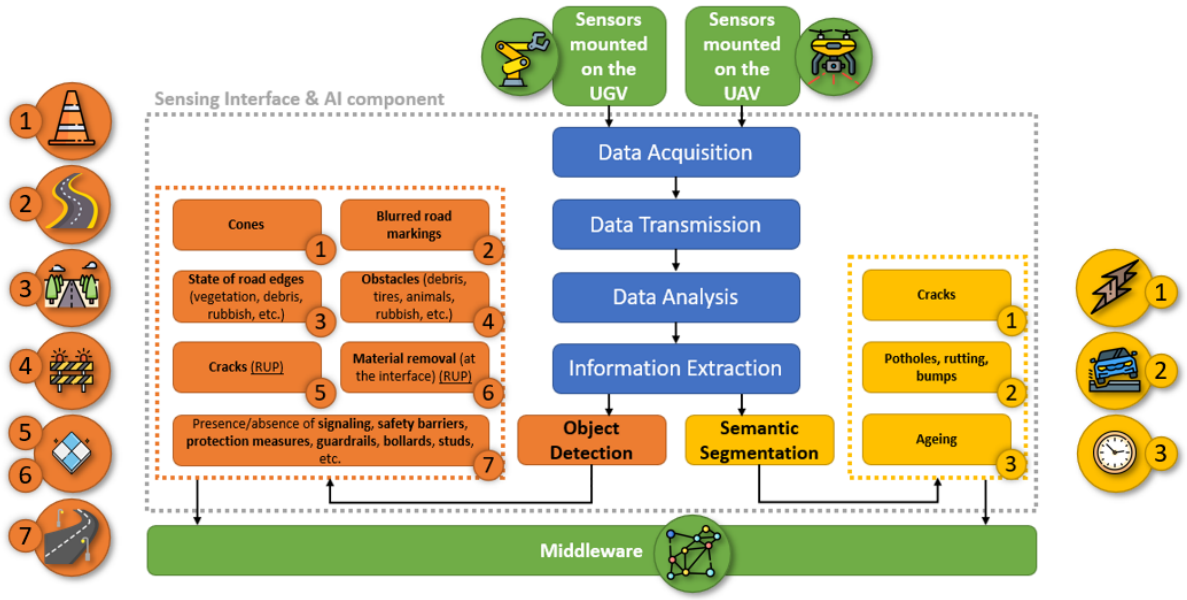


Figure 22: Sensing interface and AI component of the HERON system.

Regardless of the sensor that the data is deriving from (e.g., sensors installed on the UGV or an aerial UAV), two different approaches were considered: (i) a multi-class and multi-label detection scheme and (ii) a binary semantic segmentation task. In the former case, as will be presented in Sections 4.4-4.6, the AI component evaluates the situation of the road and in parallel localizes the RoIs (e.g., localization of cracks, potholes, blurred road markings, and traffic cones) in order the navigation and maintenance process to be effectively planned. In the latter case, as will be demonstrated in Section 4.7, the AI component precisely localizes the defect (e.g., segmentation of crack faces), in order to determine and understand the exact shape of the defected region. Thereby, more sophisticated and challenging sensor-based manipulation strategies and robotic tasks will be able to be performed by the HERON platform, such as visual servoing and motion estimation.

4.3 3D information extraction

Image detection/segmentation alone does not provide all the information needed for the robot to complete the physical tasks it is meant to. It detects the required targets but only in the 2D frame of an image, instead of the 3D world frame of the robot. To tackle this challenge, we will be using a stereoscopic camera, to provide us with depth alongside 2D information. Consecutively, through this information the 3D location of the target can be deduced, relative to the camera. Then, this position can be translated into a relative position to the robot's world frame, which is what is needed to proceed with its manipulation tasks.

Stereo view could also be providing us with the required information to calculate the volume of cracks, potholes, or cones if needed, to further guide the physical tasks. For example, the filling procedure might need the volume information of the pothole to fill, to calculate the material needed to completely fill the gap. If this is the case, 3D features such as volume will be calculated.

4.4 Real-time traffic cone detection

Great changes have taken place in intelligent technology such as object detection in road networks. Despite the technological progress, the demands for driving safety, efficiency, and automated maintenance systems have also increased significantly. There are crucial challenges such as the capability to cope with temporary and sudden circumstances such as accidents and road construction. Among the numerous objects, traffic cones need to be recognized since they present spatio-temporal visual appearance periodicity and are constantly replaced and moved in the road network.

The present section outlines a deep learning approach to effectively recognizing traffic cones in roadwork images collected from multiple sources. This application was implemented with YOLOv5 algorithm which is widely used for object detection problems [16]. We created a dataset of RGB roadwork images that were annotated by engineer experts within the framework of the HERON project. The traffic cone identification task can be addressed as an on-road object detection problem. The aim, therefore, is to broaden current studies of object detection issues and adapt them to the requirements of contemporary road network issues.

Within the HERON project, the aforementioned implementation can contribute to traffic road efficiency and safety development, while in parallel supporting the pre-and post- intervention phase including visual inspections and dispensing and removing traffic cones in an automated and controlled manner. Consequently, the HERON UGV by utilizing the state-of-the-art object detection YOLOv5 algorithm (see Section 4.1.2) will be able to carry out the dispatching and the removal process of the traffic cones effectively in an automated manner, thus avoiding accidental risks for the personnel and make the maintenance more secure, in particular when the weather conditions are adverse.

The literature presents various noteworthy attempts at studies that use road images for object detection methods with deep learning techniques, in order to confront road issues [22]. Object detection methods can apply to different aspects of the above-mentioned issues. The work of [39] presents a single shot detection and classification of road users based on the real-time object detection system YOLO. This method is applied to the pre-processed radar range-Doppler-angle power spectrum. The study of [24] suggests an on-road object detection using SSD which is a detection mechanism based on a deep neural network. In [26] is proposed a novel deep learning anchor-free approach based on CenterNet for road object detection. The paper of [37] focuses on an object detection system called YOLO in order to enhance autonomous driving and other types of automation in transportation systems. Object detection is essential for automated driving and vehicle safety systems. For this purpose, the article [17] compares five algorithms to inspect the contents of images, Region-based Fully Convolutional Network (R-FCN), Mask Region-based Convolutional Neural Networks (Mask R-CNN, Single Shot Multi-Box Detector (SSD), RetinaNet and YOLOv4.

Obstacle recognition on road images is another aspect of object detection. The work of [47] implemented an obstacle detection and avoidance driverless car using Convolutional Neural Networks. In the paper of [40] a deep learning system, using Faster Region-based convolutional neural network was employed for the detection and classification of on-road obstacles such as vehicles, pedestrians, and animals. Tsung-Ming Hsu et al. presented a deep learning model to mimic driving behaviors by learning the dynamic information of the vehicle along with image information in order to improve the performance of a self-driving vehicle. For the

implementation of the model, they placed traffic cones on the road to collect the scene of avoiding obstacles [19].

Little work has been presented in the literature on cone detection with deep learning techniques. The work of [54] utilized a machine vision system with two monochrome cameras and two color cameras in order to recognize the color and position of traffic cones. Another approach is the study of [4], which presents an overview of object detection methods and used sensors and datasets in an autonomous driving application. [49] focuses on the detection of a construction barrel, which includes a construction cone, a looper cone, a barricade, and four types of signs, via a collection of road images. Ankit Dhall et al. presented an accurate traffic cone detection and estimation of their position in the 3D world in real-time [12] presents an implementation of a robust autonomous driving algorithm using the Viola-Jones object detection method for traffic cones recognition. The study of [2] proposes a lightweight neural network to perform cone detection from a racing car in order to research autonomous driving. Finally, the work of [53] presents a deep architecture called ChangeNet for detecting changes between pairs of images and expressing the same semantically. The dataset has 11 different classes of structural changes including traffic cones on road.

4.4.1 Object detection model

The presented system of this section utilizes the roadwork image dataset which is described in detail in 4.4.2 to identify traffic cones. Each RGB image was properly fed into the YOLOv5 algorithm which was presented and analyzed in Section 4.1.2. In a nutshell, in the following sections, we present and evaluate a YOLOv5 algorithm for traffic cone recognition over a multisource roadwork image dataset. The utilized technique uses a deep learning framework, identifying traffic cones as an object detection scenario. The model was able to achieve high scores and successfully managed the identification task. The architecture of YOLOv5 is illustrated in Figure 19.

4.4.2 Dataset description

To train and evaluate the deep learning object detector, a dataset that contains RGB images was collected and manually annotated using labelImg [52], which is a graphical image annotation tool. labelImg is written in Python and uses Qt for its graphical interface. The produced annotations (see Figure 23) are saved as .txt files that store the information of the annotated bounding boxes.

(a) RGB image

```
0 0.372656 0.093814 0.054688 0.183505
0 0.705469 0.116495 0.092188 0.220619
0 0.800000 0.183505 0.087500 0.280412
0 0.856250 0.213402 0.090625 0.307216
0 0.897656 0.387629 0.176563 0.453608
0 0.347656 0.650515 0.329688 0.698969
```

(b) YOLO annotation format

Figure 23: Each RGB image (a) has a corresponding .txt file with the bounding box information (b) of the traffic cones (ID, x, y, w, h).

In particular, for each RGB image (see Figure 23a) a corresponding text file was generated (see Figure 23b) that contains a number of rows equal to the number of the bounding boxes (i.e., traffic cones) in the specific image. As one can observe in Figure 23b, each row consists of five numbers: (i) An integer number, starting at 0, that represents the class ID, which therefore in our case always equals 0, since the cone detection task is a single class problem; (ii) the horizontal coordinate x of the central pixel of the bounding box; (iii) the vertical coordinate y

of the central pixel of the bounding box; (iv) the width w of the bounding box and (v) the height h of the bounding box. It is noted that the central position of the bounding box (ii-iii), as well its dimensions (iv-v) are real numbers on a scale of 0 to 1, and, therefore, represent the relative location and size of the bounding box with respect to the whole image.

The dataset contains RGB data from heterogeneous sources and sensors (e.g., DSLR cameras, smartphones, UAVs). Furthermore, the images vary in terms of illumination conditions (e.g., overexposure, underexposure), environmental landscapes (e.g., highways, bridges, cities, countrysides), and weather conditions (e.g., cold, hot, sunny, windy, cloudy, rainy, and snowy). In parallel, several images include various types of occlusions, thus making the traffic cone detection task more challenging.

The total number of RGB images in the dataset is 540 with various resolutions ranging from 114×170 to 2,100×1,400. It is underlined that the total number of traffic cones in the entire dataset is 947. Representative samples of the dataset are demonstrated in Figure 24. From the images of the whole dataset, 92.5% were used for training the deep model, and 7.5% for testing its effectiveness. Among the training data, 80% of them were used for training and the remaining 20% for validation. The traffic cone detection dataset is made available online at: https://github.com/ikatsamenis/Cone-Detection/ (accessed date 31 May 2022).

Figure 24: Indicative images from the traffic cone detection dataset.

4.4.3 Experimental setup - Model training

Hence, for the training process, we utilized 500 images, 400 of which were included in the train set and 100 in the validation set. It is noted that the training data should include images with non-labeled objects (i.e., empty .txt files) and in particular, the negative samples without bounded boxes should be equal to the positive images with objects [8]. To this end, 50% of the data of both train and validation sets (i.e., 200 and 50 images respectively) are negative samples, while the rest contain at least one traffic cone. Lastly, it is underlined that to further generalize the learning process, we augmented the training data by horizontally flipping the corresponding images, thus increasing the train set size from 400 to 800.

The YOLO object detector was trained and evaluated using an NVIDIA Tesla K80 GPU with 12 GB of memory, provided by Google Colab. We trained the network, using batches of size 32, for 200 epochs, and set the input image resolution to 448×448 pixels. This work is based on the YOLOv5 small model in order to reduce the computational cost of the detection task. Towards this direction, the network takes up less than 15 MB of storage and thus can be easily embedded in smartphone applications and various low-memory digital devices or systems, including drones and microcontrollers.

4.4.4 Evaluation metrics

The Intersection over Union (IoU) metric was employed in evaluating the performance of the proposed method. IoU is the most popular evaluation metric used in the object detection benchmarks [45]. In order to apply IoU, ground-truth bounding boxes and predicted bounding boxes from our model are needed. This metric is used to evaluate how close the predicted bounding boxes are to the ground-truth bounding boxes. The greater the region of overlap, the greater the IoU, and therefore the detection accuracy as shown in Figure 25. Consequently, IoU is a number from 0 to 1 that specifies the size of the overlapping area between prediction and ground truth.

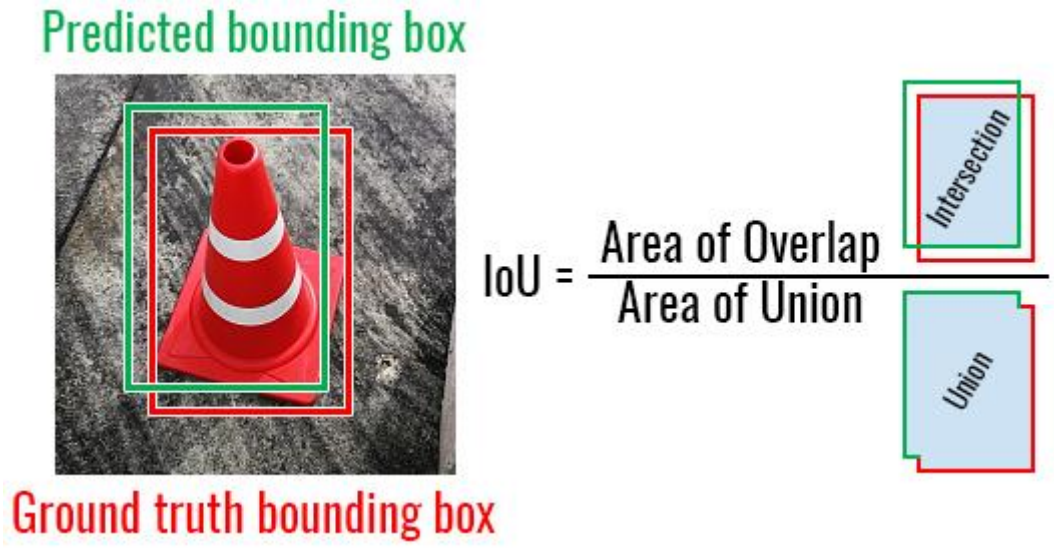


Figure 25: Calculation of the IoU metric. The predicted bounding box is depicted in green color and the ground truth in red.

4.4.5 Experimental validation

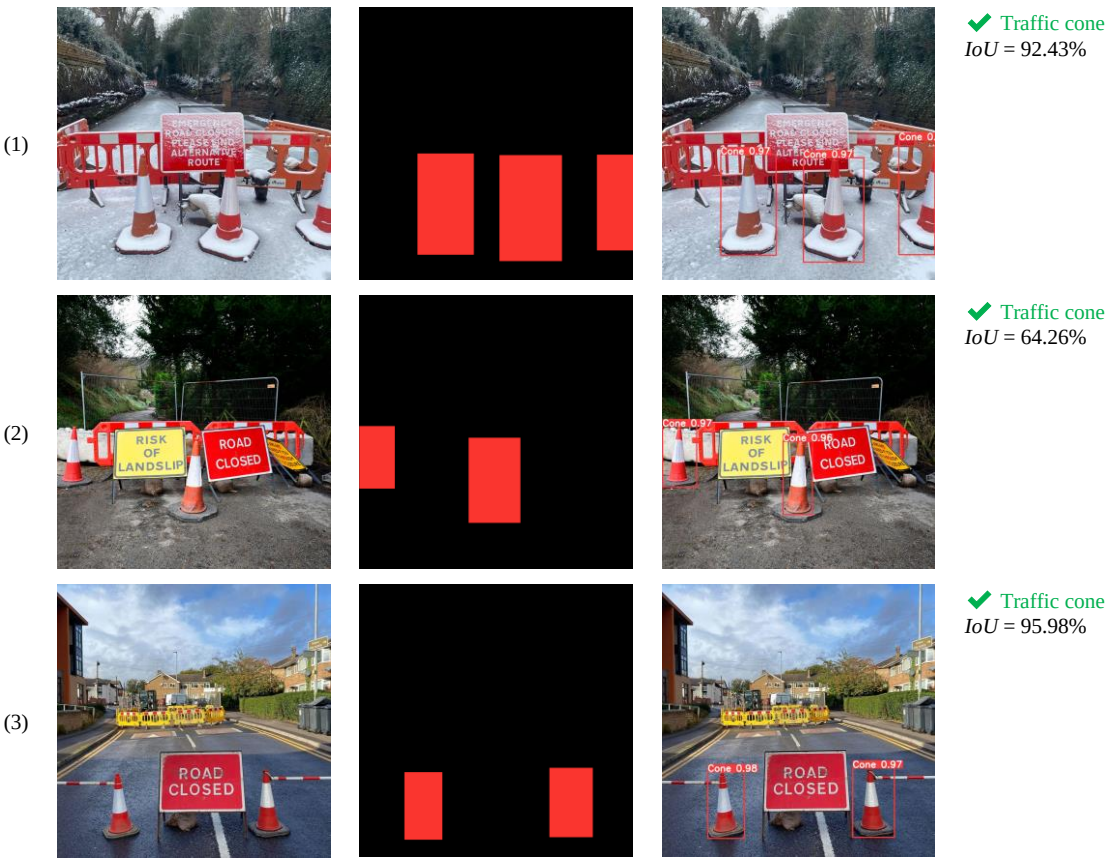
The proposed algorithm reached an excellent average IoU score of 91.31%±5.42% with a confidence level of 95% over the data of the test set. Moreover, the network demonstrated an average prediction time of 0.065±0.029 seconds per image.

The experimental results using the YOLOv5 architecture are shown in the next two subsections (see Sections 4.4.6 and 4.4.7). In particular, on the one hand, Figure 26 presents the detection capabilities of the proposed network in RGB images with traffic cones. On the other hand, Figure 27 depicts the performance of the deep learning model on road images without traffic cones in order to evaluate it in terms of misidentifications that lead to false-positive detections. In both these figures, the first column corresponds to the original RGB images followed by their ground truth bounding boxes in the second column. The third column illustrates the predicted bounding boxes with their corresponding confidence scores. Finally, the last column demonstrates the performance of the model and the IoU score of the respective input image.

4.4.6 Evaluation of the object detector on road images with traffic cones

In the specific section and in particular, in Figure 26 we demonstrate the detection capabilities of the proposed YOLOv5 architecture in the automated traffic cone detection task. More specifically, Figure 26 presents 20 RGB images of the test set that depict at least one traffic cone. It is noted that the images are unseen data during the training process of the deep network. As one can see in the aforementioned figure, and in particular in the third and fourth columns, the model showed excellent identification and localization performance of the traffic cones, even in challenging images that include extreme weather events [e.g., see Figure 26 (1), (17)], low light conditions [e.g., see Figure 26 (11)], and occlusions [e.g., see Figure 26 (15), (20)].

It is however emphasized that in very rare cases the model failed to detect (false negative) one of the traffic cones of the image [e.g., the cone on the left in Figure 26 (6)]. In parallel, in very rare cases the network misclassified an object (false positive) as a traffic cone [e.g., part of the safety barrier in Figure 26 (9)]. Nevertheless, it is underlined that the input data of the HERON AI system is consecutive RGB frames of a video sequence, and, therefore, even if the detection fails for the current frame, it is highly likely that it will succeed in the next ones.



(4)



✔ Traffic cone
IoU = 94.89%

(5)



✔ Traffic cone
IoU = 94.35%

(6)



✔ Traffic cone
IoU = 67.10%

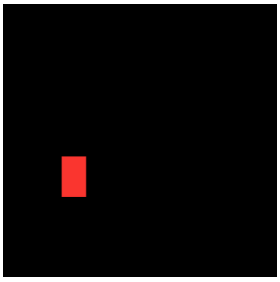
(7)


✔ Traffic cone
IoU = 93.40%

(8)

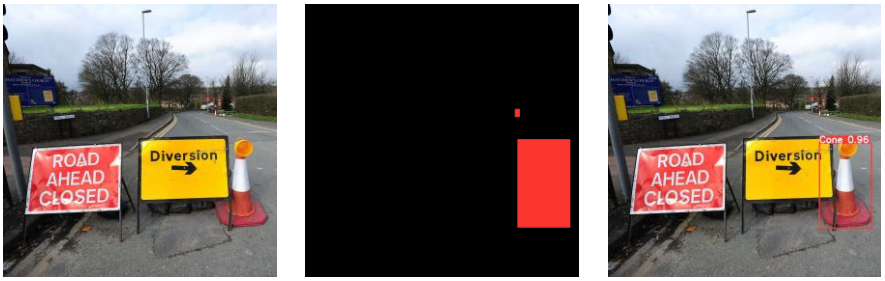

✔ Traffic cone
IoU = 94.71%

(9)




✔ Traffic cone
IoU = 73.45%

(10)

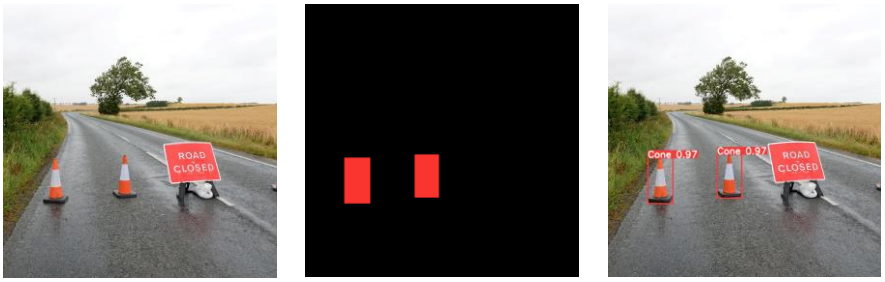


✔ Traffic cone
IoU = 92.49%

(11)

✔ Traffic cone
IoU = 92.24%

(12)



✔ Traffic cone
IoU = 94.95%

(13)

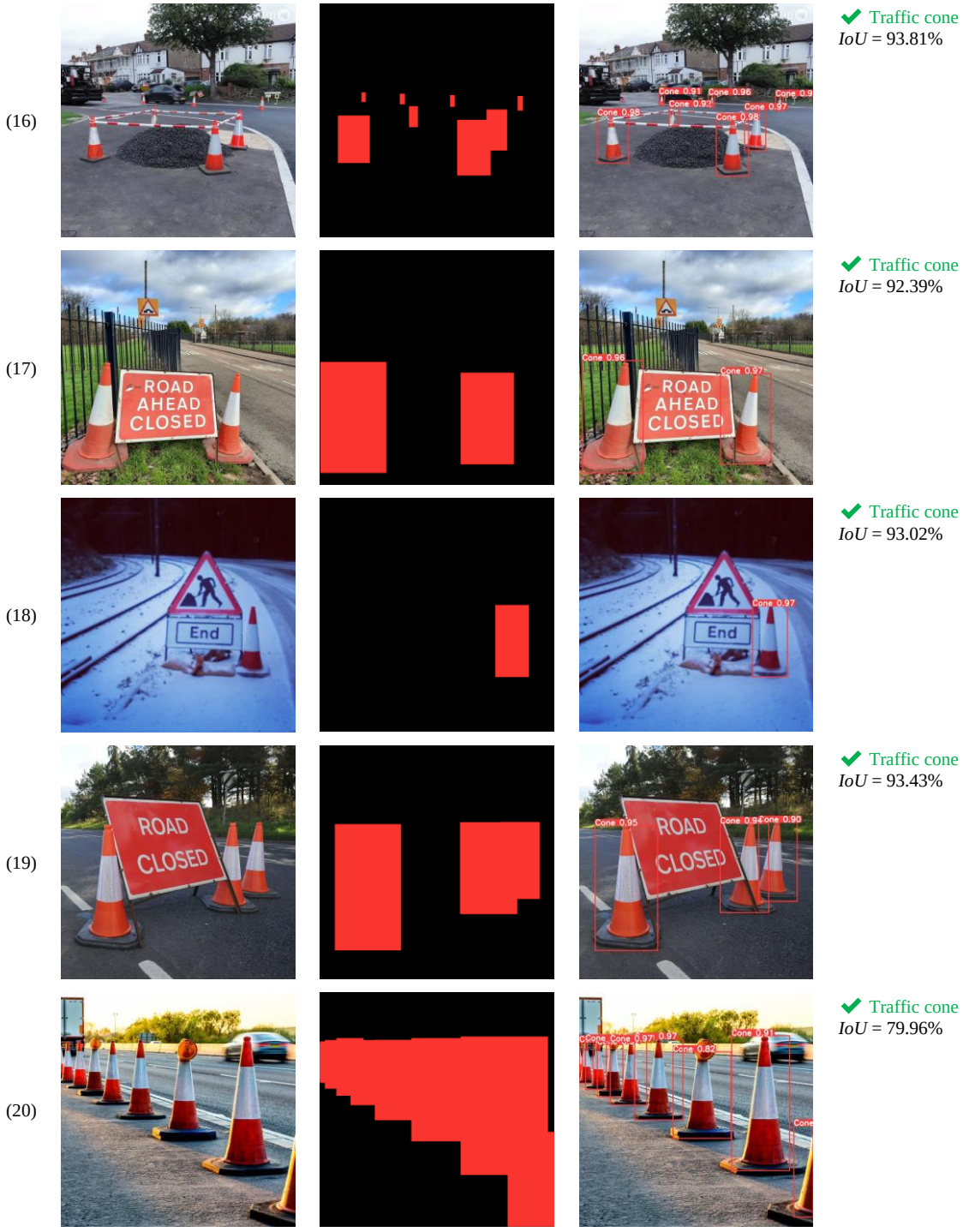
✔ Traffic cone
IoU = 88.93%

(14)

✔ Traffic cone
IoU = 78.83%

(15)

✔ Traffic cone
IoU = 79.89%



(a) Input RGB image (b) Ground truth (c) Object detection output (d) Performance

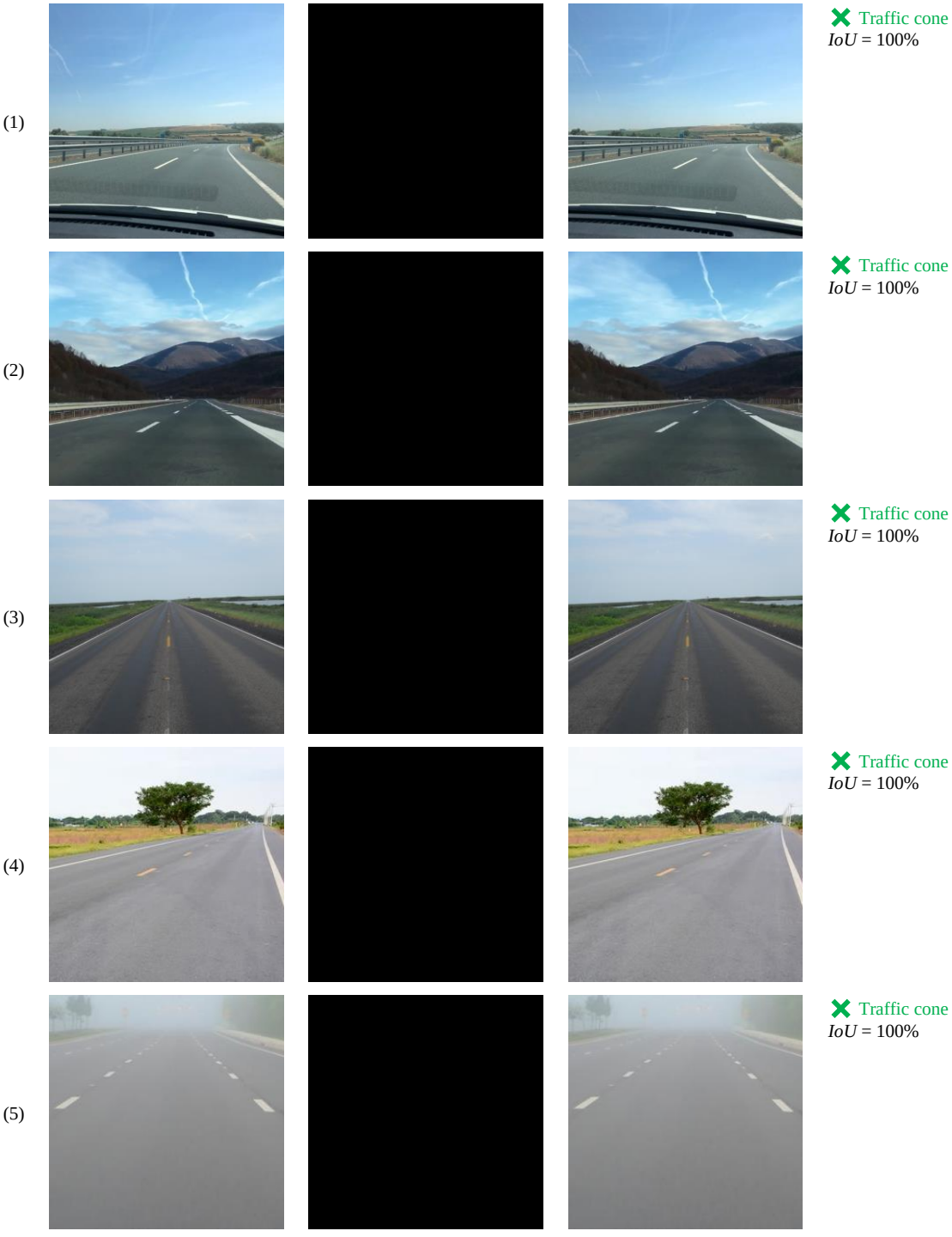
Figure 26: Automated localization of traffic cones (red bounding boxes) on the test set of a custom dataset using small YOLOv5 deep model.

4.4.7 Evaluation of the object detector on road images without traffic cones

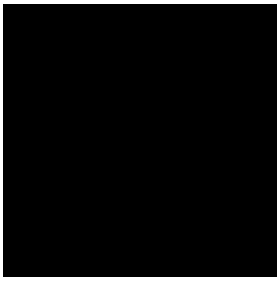
Similar to the previous section, in Figure 27 we demonstrate the identification capabilities of the proposed YOLOv5 network in the automated traffic cone detection task. More specifically, Figure 27 presents 20 RGB images of the test set that depict road infrastructures without traffic cones. It is noted that the images are unseen data during the training procedure of the model. As one can see in the aforementioned figure, and in particular in the third and fourth columns,

the model demonstrated state of the art classification performance, even in challenging images, such as for instance, data that include humidity, as in Figure 27 (5).

It is however noted that in very rare cases the network misclassified an object (false positive) as a traffic cone [e.g., part of the asphalt, and in particular a road marking, in Figure 27 (11)]. Nevertheless, it is highlighted that the input data of the HERON AI system is consecutive RGB frames of a video sequence, and, thus, even if the automated recognition fails for a given frame, it is highly likely that it will succeed in the next ones.

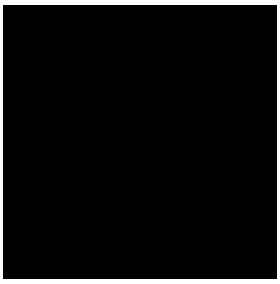


(6)

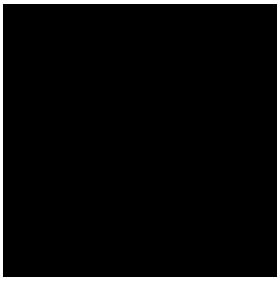

✗ Traffic cone
IoU = 100%

(7)

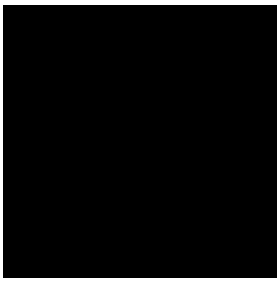

✗ Traffic cone
IoU = 100%

(8)

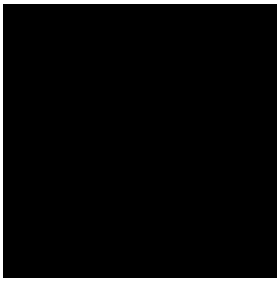

✗ Traffic cone
IoU = 100%

(9)

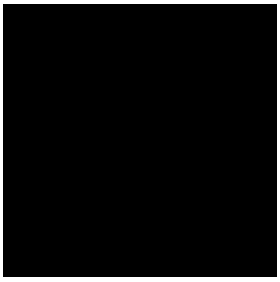

✗ Traffic cone
IoU = 100%

(10)

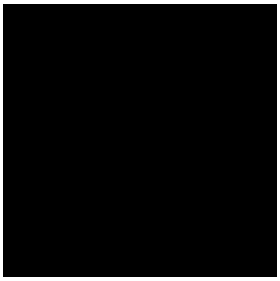

✗ Traffic cone
IoU = 100%

(11)

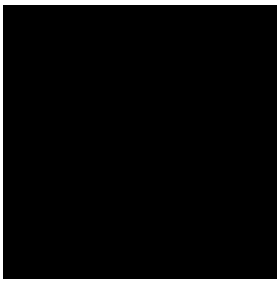

✓ Traffic cone
IoU = 0%

(12)

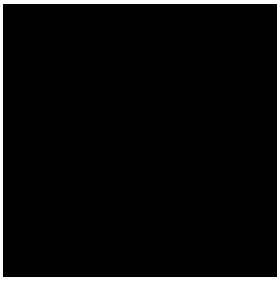

✗ Traffic cone
IoU = 100%

(13)

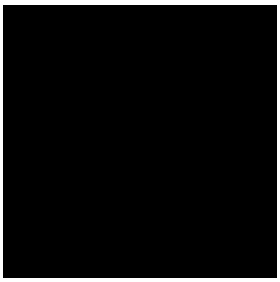

✗ Traffic cone
IoU = 100%

(14)

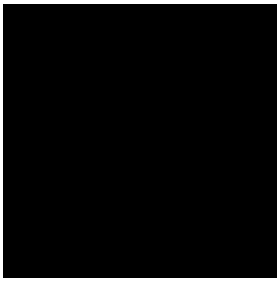

✗ Traffic cone
IoU = 100%

(15)

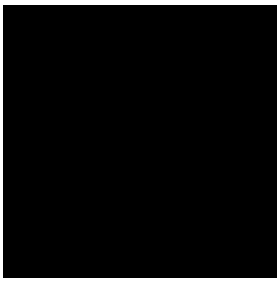

✗ Traffic cone
IoU = 100%

(16)

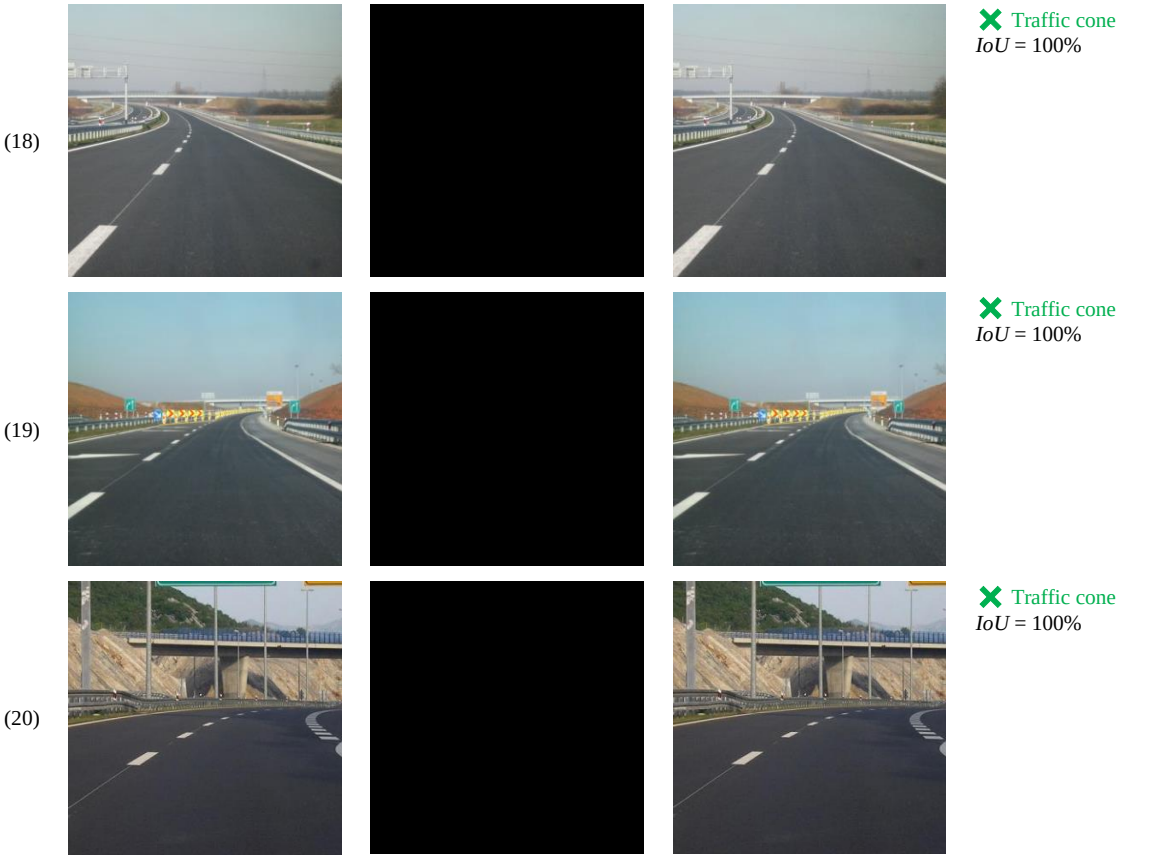




✗ Traffic cone
IoU = 100%

(17)


✗ Traffic cone
IoU = 100%



(a) Input RGB image (b) Ground truth (c) Object detection output (d) Performance

Figure 27: Evaluation of the YOLOv5 object detector on road images without traffic cones.

4.5 Real-time road defect detection

A country's road infrastructure stimulates economic and social development since it provides access to markets, employment, and basic social services, such as education and healthcare. However, many factors, such as weather conditions, geographical location, road age, frequency of usage and more, lead to the road's quality deterioration over time. Therefore, road maintenance plays a crucial role and requires regular monitoring and assessment of road conditions.

Nowadays, efficient road maintenance mainly relies on human visual inspection or high-performance sensors, which is time-consuming and expensive. In response to the abovementioned problem, many methods have been studied, such as the use of laser technology or image processing, in order to efficiently monitor and inspect road infrastructure. To this end, the focus, which is presented in this section, is to automate the detection of three main types of road damages, i.e., potholes, cracks, and blurred markings, by using one of the state-of-the-art deep learning algorithms, YOLOv5, on images captured from various RGB images.

Many deep learning solutions have been proposed in the last years in order to provide a cost-effective road infrastructure monitoring tool. More specifically, studies have been lately focused not only on detecting road defects but also on categorizing the defects into different types, since the differentiation among damage types is crucial for effective road maintenance planning. For example, in the work of [20] a method for pothole detection was developed. In addition, an approach for detecting two types of cracks (i.e., horizontal and vertical) was later proposed [56] and similarly, in another study the damages were classified into three types: vertical, horizontal and crocodile [1]. Later on, a more thorough classification was implemented, in which the damages are classified into eight different categories [33] (as shown on Table 10 of Section 4.5.2). Afterward, a road damage detection for multiple countries was developed [5], which expanded the dataset used in [33], which contained only images from Japan, with images from India and the Czech Republic. The proposed method categorized the damages into four main types: longitudinal/parallel cracks, transverse/perpendicular cracks, alligator/complex cracks, and potholes. The proposed dataset in [5] was made publicly available [6] and formed the basis for the organization of the Global Road Damage Detection Challenge (GRDDC), which led to many innovative solutions that are summarized in [7].

4.5.1 Object detection model

Similar to the previous section, in which the automated traffic cone identification of the HERON system was demonstrated, the state-of-the-art object detection methods were analyzed and compared for the task of road defect detection. Finally, the YOLOv5 model was chosen as the basic framework for the current work, which consists of an end-to-end real-time object detector. The architecture of YOLOv5 is exemplified in Figure 19. Consequently, the focus of this section is to automate the detection of three main types of road damage, i.e., potholes, cracks, and blurred marking, by using one of the state-of-the-art deep learning algorithms, YOLOv5, on images captured from various RGB sensors.

4.5.2 Dataset description

The Road Damage Dataset 2019 [32] was utilized for the model training, which consists of 13,135 road images captured in India, Japan, and the Czech Republic and contains more than 30,000 instances of road damage. The instances for each defect used for training, are shown in Table 9.

Table 9: Number of instances per damage type in the train dataset.

Damage type	Number of instances
Crack	15,235
Pothole	2,259
Blurred road marking	4,901

The data collection involved collecting road images using vehicle-mounted smartphones, using a smartphone application that was designed for this task. The aforementioned image capturing rate was chosen to prevent overlap or leakage during the picture collection when the average speed of the vehicle is approximately 40 km/h.

In the work of [32] eight damage categories were considered in total, based on the Japanese Maintenance Guidebook for Road Pavement [35]. The deterioration is classified into two categories, namely pavement deterioration (D00, D01, D10, D11, D20, D40) and road marking deterioration (D43, D44), as exemplified in Table 10. An additional category has been added (D50), which is not considered damage, in order to prevent the misclassification of a utility hole and a pothole. However, in the current work, all the crack defect types are considered as one, the newly added category was excluded and consequently, the dataset is divided into three main categories: cracks, potholes and blurred markings. The sample images of the training dataset are visualized in Figure 28.

Table 10: Road damage types and definitions considered in the work of [33].

Damage type			**Detail**	**Class name**
Crack	**Linear Crack**	**Longitudinal**	Wheel-marked part	D00
			Construction joint part	D01
		Lateral	Equal interval	D10
			Construction joint part	D11
	Alligator Crack		Partial pavement, overall pavement	D20
Other Damage			Rutting, bump, pothole, separation	D40
			Crosswalk blur	D43
			White line blur	D44
			Utility hole (maintenance hatch)	D50

Figure 28: Sample images from the training dataset for potholes, cracks, and blurred marking defects, captured in Japan (a), India (b), and the Czech Republic (c).

In order to test the model, a custom dataset was created and manually annotated by engineer experts within the framework of the HERON project. In total, 50 RGB images were collected from various areas in Greece and online resources, using typical smartphone cameras, containing images with both road damage and without. Instances from all 3 categories are included, as well as their combination, as illustrated in Sections 4.5.6-4.5.10.

4.5.3 Experimental setup - Model training

The YOLO object detector was trained and evaluated using an NVIDIA Tesla K80 GPU with 12 GB of memory, provided by Google Colab. We trained the network, using batches of size 32, for 200 epochs, and set the input image resolution to 448×448 pixels. This work is based on the YOLOv5 small model in order to reduce the computational cost of the detection task. Towards this direction, the network takes up less than 15 MB of storage and thus can be easily embedded in smartphone applications and various low-memory digital devices or systems, including drones and microcontrollers.

4.5.4 Evaluation metrics

Detection task

As already mentioned in Section 4.4.4 popular similarity measure for object detection problems is the Intersection over Union (IoU) metric, also known as the Jaccard index, which is calculated using the predicted and ground-truth bounding boxes. Evidently, the bigger the overlap between the two bounding boxes, the higher the IoU score and therefore, detection accuracy. For object detection tasks, it is common to calculate precision and recall for a given IoU threshold value, e.g., IoU≥0.5. Therefore, a prediction is regarded as correct only if the IoU exceeds this limit.

Classification task

For the evaluation of the classification accuracy, the precision, recall, and F1-score metrics were chosen.

Precision, also known as positive predictive value (PPV), measures the accuracy of the model's predictions and is calculated as seen in the following expression:

$$PPV = \frac{TP}{TP + FP} \tag{1}$$

where true positives (predicted correctly as positive) are denoted as TP and false positives (predicted incorrectly as positive) as FP. It is noted that Precision, which is the ratio of correct positive outcomes to the total positive outcomes that the network considers, and thus indicates how good a network is when its output is positive. A low precision score implies a high number of false alarms.

Similarly, recall, also known as sensitivity, measures how well the model predicts the total of positives and is calculated as seen in the following expression:

$$TPR = \frac{TP}{TP + FN} \tag{2}$$

where false negatives (incorrectly predicted as negatives) are denoted as FN. It is underlined that recall, which is the percentage of correct positive outcomes to the total of positive cases in the ground truth, and therefore it shows how many of the positive classes the network can correctly predict. A low recall score entails that the classifier has a high number of misses.

Finally, the F1-score is a combination (harmonic mean) of these two last abovementioned metrics and is described as the harmonic mean of the precision and recall. It is calculated as in the following expression:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3}$$

4.5.5 Experimental validation

The results for the detection task are visualized in Figure 29 with a confidence level of 95% over the data of the test set. Regarding the computational complexity, the model needs an average time of 0.087 seconds to detect the road defects on an image. Additionally, a few correct, false and missed model predictions are presented alongside the original image and the ground truth in Sections 4.5.6-4.5.10.

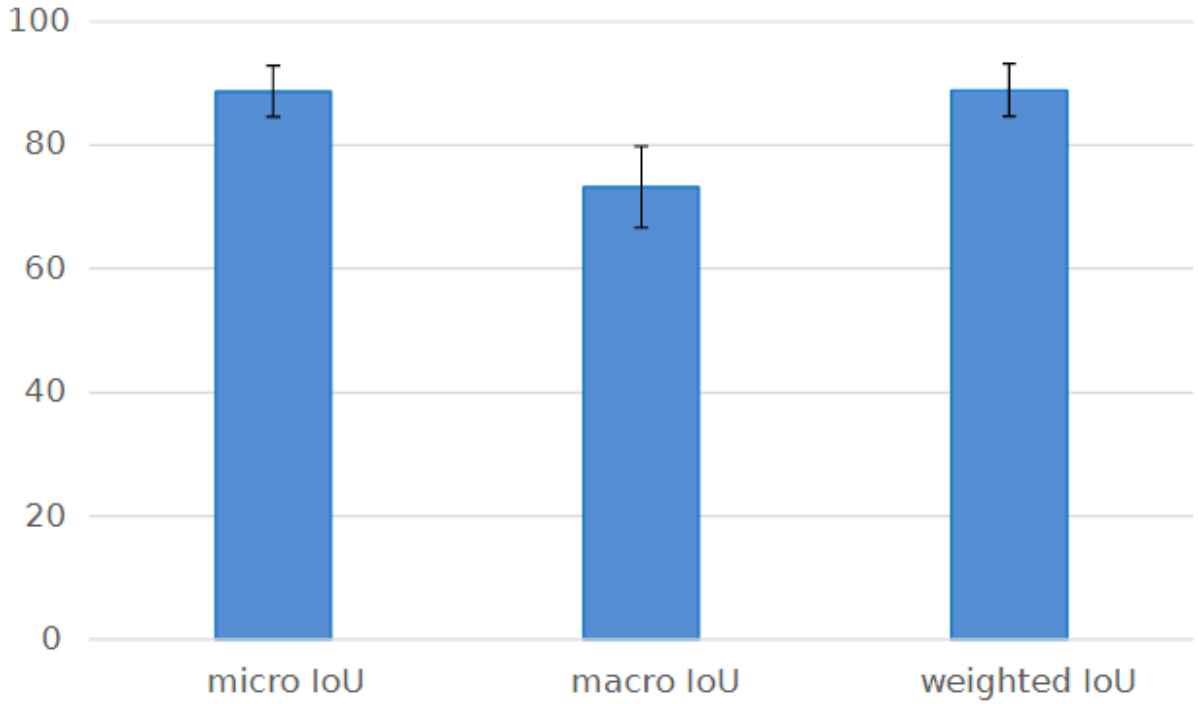


Figure 29: Micro, macro and weighted IoU scores.

The classification results are presented in Figure 30 and Figure 31 using different averaging and scores per class, respectively. It is noted that the micro average uses the global number of TP, FN, and FP and calculates directly the respective performance scores. On the other hand, macro average calculates the metric separated by class but without using weights for the aggregation. Lastly, the weighted average calculates the score for each class independently, but when adding them up it utilizes a weight that depends on the number of true labels in each class.

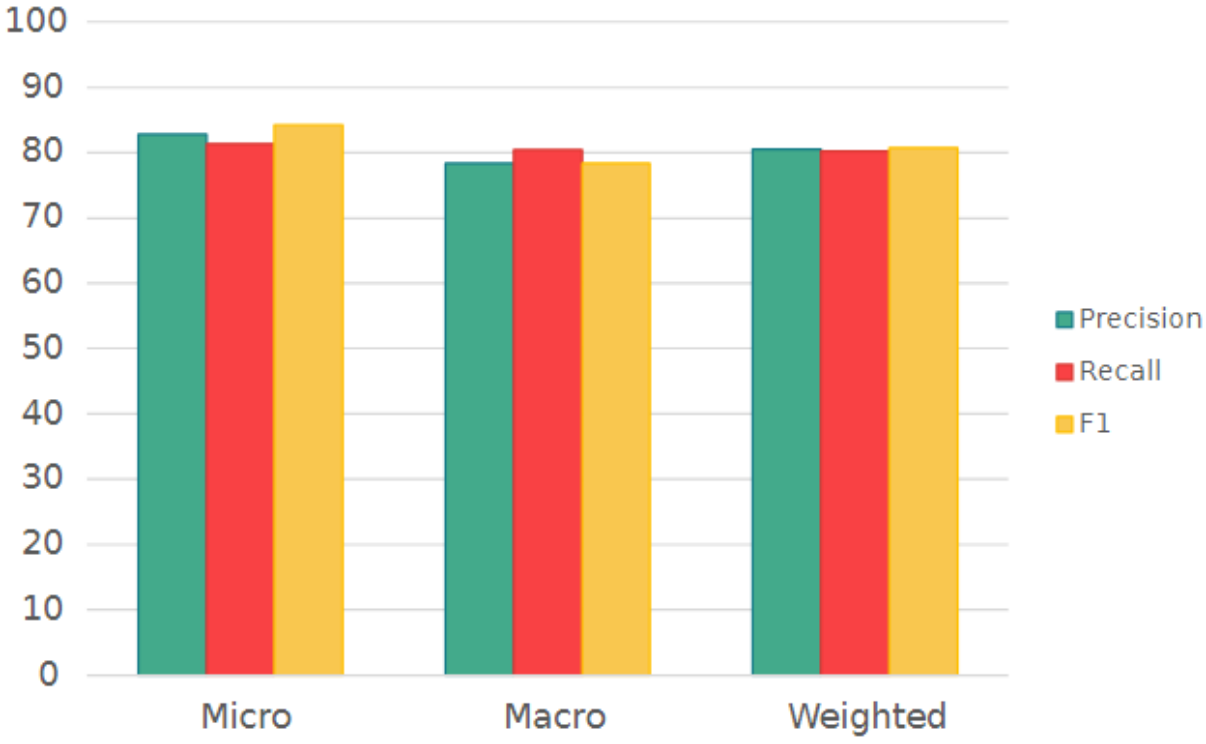


Figure 30: Classification scores calculated with micro, macro, and weighted averaging, respectively.

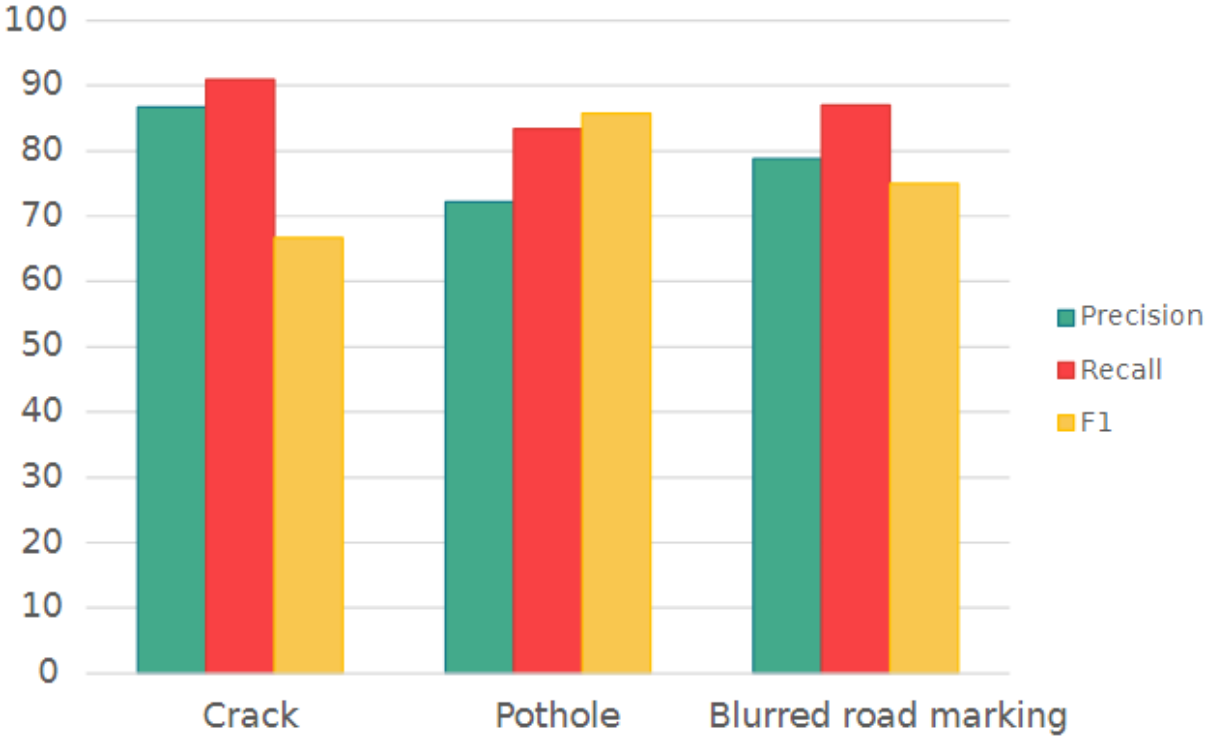


Figure 31: Classification scores calculated per class.

In a nutshell, in this section, an image-based solution for road infrastructure monitoring was presented. In the proposed solution the YOLOv5 detection model is utilized to detect and classify three types of road damages in the processed images. The final model was able to achieve an IoU score up to 88.89% for the detection task and an F1 score up to 80.72% for the classification task with precision and recall scores at 84.26% and 78.38%, respectively.

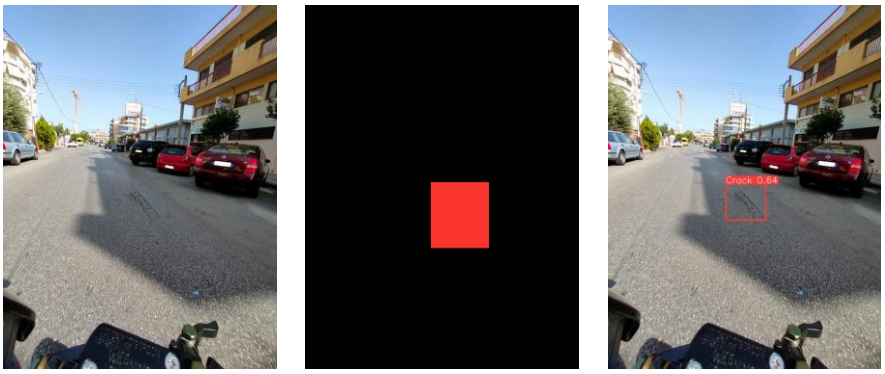
The experimental results using the YOLOv5 architecture are shown in the next five subsections (see Sections 4.5.6-4.5.10). More specifically, (i) Figure 32 presents the detection capabilities of the proposed network in road images with cracks; (ii) Figure 33 illustrates the identification capabilities of the proposed network in road images with potholes; (iii) Figure 34 demonstrates the recognition capabilities of the proposed network in road images with blurred road markings, (iv) Figure 35 shows the localization capabilities of the proposed network in road images with more than one defects (i.e., cracks, potholes, and blurred road markings); and lastly (v) Figure 36 depicts the performance of the deep learning model on road images without defects in order to evaluate it in terms of misidentifications that lead to false-positive detections. In all the aforementioned figures, the first column corresponds to the original RGB images followed by their ground truth bounding boxes in the second column. The third column shows the predicted bounding boxes with their corresponding confidence scores. Finally, the last column demonstrates the performance of the model on the classification task as well as the IoU scores of the respective input image regarding the object detection task.

4.5.6 Evaluation of the object detector on road images with cracks

In the specific section and in particular, in Figure 32 we demonstrate the detection capabilities of the proposed YOLOv5 architecture in the automated crack detection task. More specifically, Figure 32 presents 11 RGB images of the test set that depict at least one crack. It is noted that the images are unseen data during the training process of the deep network. As one can see in the aforementioned figure, and in particular in the third and fourth columns, the model showed satisfactory identification and localization performance of the cracks.

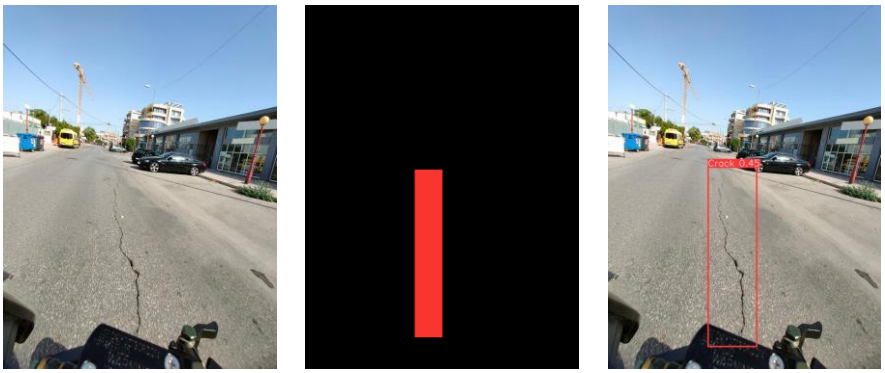
It is however emphasized that in very rare cases the model failed to detect (false negative) a crack defect of the image [e.g., Figure 32 (7)]. In parallel, in very rare cases the network misclassified an object (false positive) as a defect [e.g., Figure 32 (9)]. Nevertheless, it is underlined that the input data of the HERON AI system is consecutive RGB frames of a video sequence, and, therefore, even if the detection fails for the current frame, it is highly likely that it will succeed in the next ones.

(1)



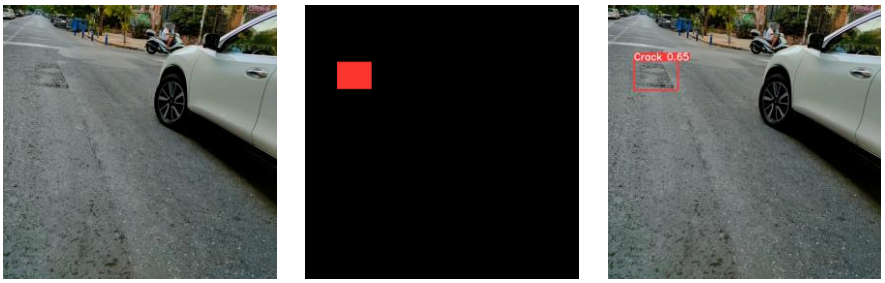
✔ Crack
✘ Pothole
✘ Road marking blur
IoU_m = 94.30%
IoU_M = 63.02%
IoU_w = 94.41%

(2)



✔ Crack
✘ Pothole
✘ Road marking blur
IoU_m = 91.92%
IoU_M = 74.09%
IoU_w = 93.58%

(3)



✔ Crack
✘ Pothole
✘ Road marking blur
IoU_m = 98.99%
IoU_M = 85.36%
IoU_w = 99.13%

(4)

✔ Crack
✘ Pothole
✘ Road marking blur
IoU_m = 55.86%
IoU_M = 48.34%
IoU_w = 52.53%

(5)

✔ Crack
✘ Pothole
✘ Road marking blur
IoU_m = 92.44%
IoU_M = 86.78%
IoU_w = 92.96%

(6)

✔ Crack
✔ Pothole
✘ Road marking blur
IoU_m = 66.81%
IoU_M = 29.67%
IoU_w = 64.75%

(7)

✘ Crack
✘ Pothole
✘ Road marking blur
IoU_m = 47.40%
IoU_M = 32.16%
IoU_w = 41.37%

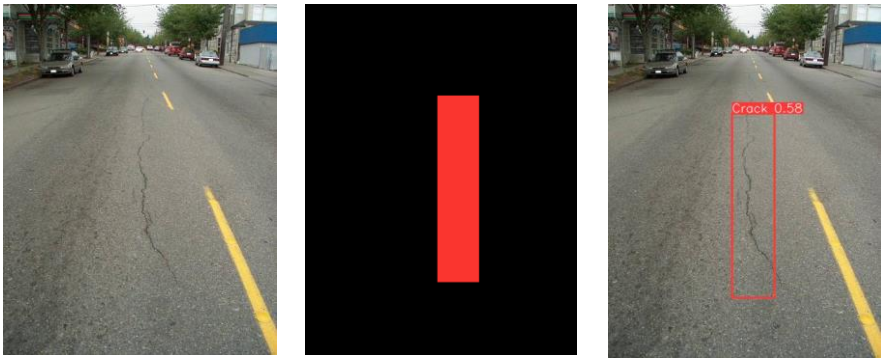
(8)

✔ Crack
✘ Pothole
✘ Road marking blur
IoU_m = 72.69%
IoU_M = 62.88%
IoU_w = 73.37%

(9)

✔ Crack
✘ Pothole
✘ Road marking blur
IoU_m = 66.28%
IoU_M = 48.14%
IoU_w = 66.21%

(10)



✔ Crack
✘ Pothole
✘ Road marking blur
IoU_m = 91.68%
IoU_M = 76.74%
IoU_w = 92.34%

(11)

(a) Input RGB image (b) Ground truth (c) Object detection output (d) Performance

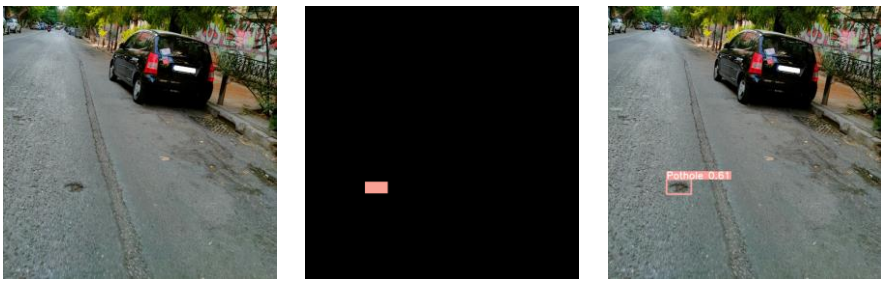
Figure 32: Automated localization of cracks (red bounding boxes) on the test set of a custom dataset using small YOLOv5 deep model trained on the dataset of [6].

4.5.7 Evaluation of the object detector on road images with potholes

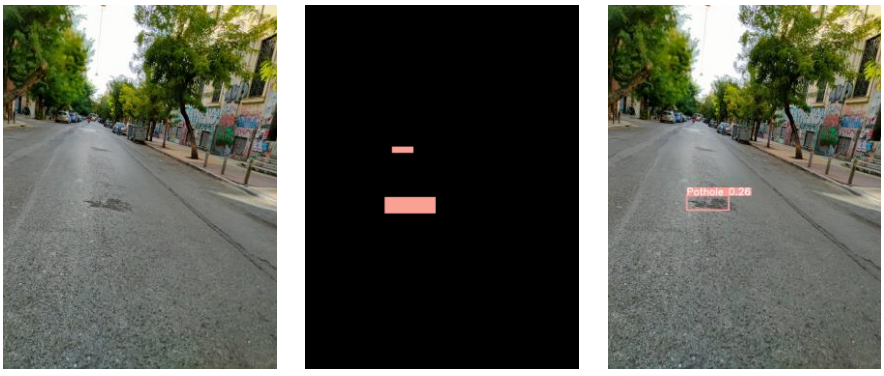
In the specific section and in particular, in Figure 33 we demonstrate the detection capabilities of the proposed YOLOv5 architecture in the automated pothole detection task. More specifically, Figure 33 presents 7 RGB images of the test set that depict at least one pothole. It is noted that the images are unseen data during the training process of the deep network. As one can see in the aforementioned figure, and in particular in the third and fourth columns, the model showed satisfactory identification and localization performance of the potholes.

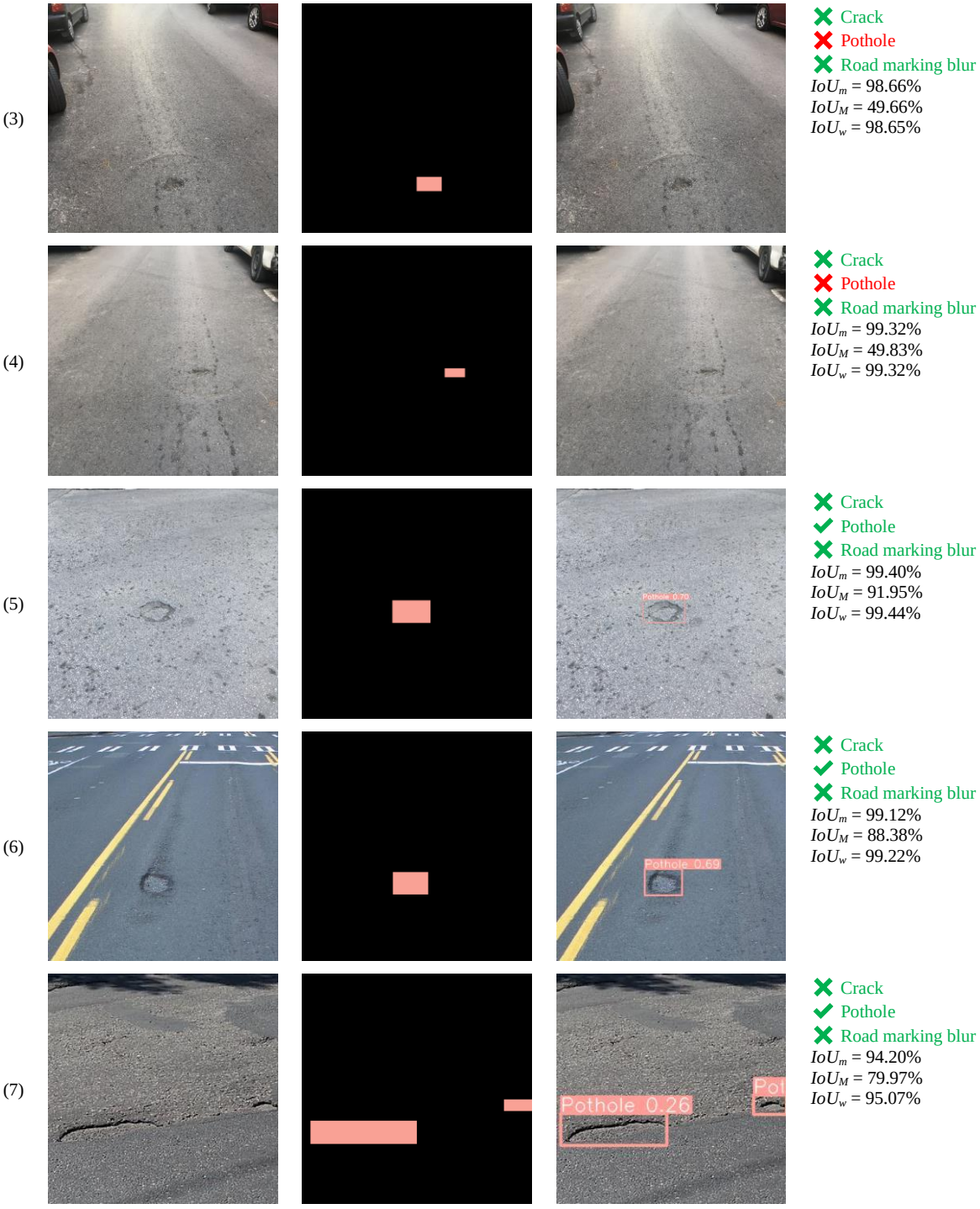
It is however emphasized that in very rare cases the model failed to detect (false negative) a pothole defect of the image [e.g., Figure 33 (3)-(4)]. Nevertheless, it is underlined that the input data of the HERON AI system is consecutive RGB frames of a video sequence, and, therefore, even if the detection fails for the current frame, it is highly likely that it will succeed in the next ones. Lastly, the model, as can be seen in Figure 33, did not misclassify an object as a defect (false positive) in any of the test RGB images.

(1)



(2)





(a) Input RGB image (b) Ground truth (c) Object detection output (d) Performance

Figure 33: Automated localization of potholes (pink bounding boxes) on the test set of a custom dataset using small YOLOv5 deep model trained on the dataset of [6].

4.5.8 Evaluation of the object detector on road images blurred road markings

In the specific section and in particular, in Figure 34 we demonstrate the detection capabilities of the proposed YOLOv5 architecture in the automated blurred road marking detection task. More specifically, Figure 34 presents 5 RGB images of the test set that depict at least one blurred road marking. It is noted that the images are unseen data during the training process of the deep network. As one can see in the aforementioned figure, and in particular in the third

and fourth columns, the model showed satisfactory identification and localization performance of the blurred road markings.

It is however emphasized that in very rare cases the network misclassified an object (false positive) as a defect [e.g., Figure 34 (2)]. Nevertheless, it is underlined that the input data of the HERON AI system is consecutive RGB frames of a video sequence, and, therefore, even if the detection fails for the current frame, it is highly likely that it will succeed in the next ones. Lastly, the model, as can be seen in Figure 34, did not fail to detect (false negative) a blurred road marking in any of the test RGB images.

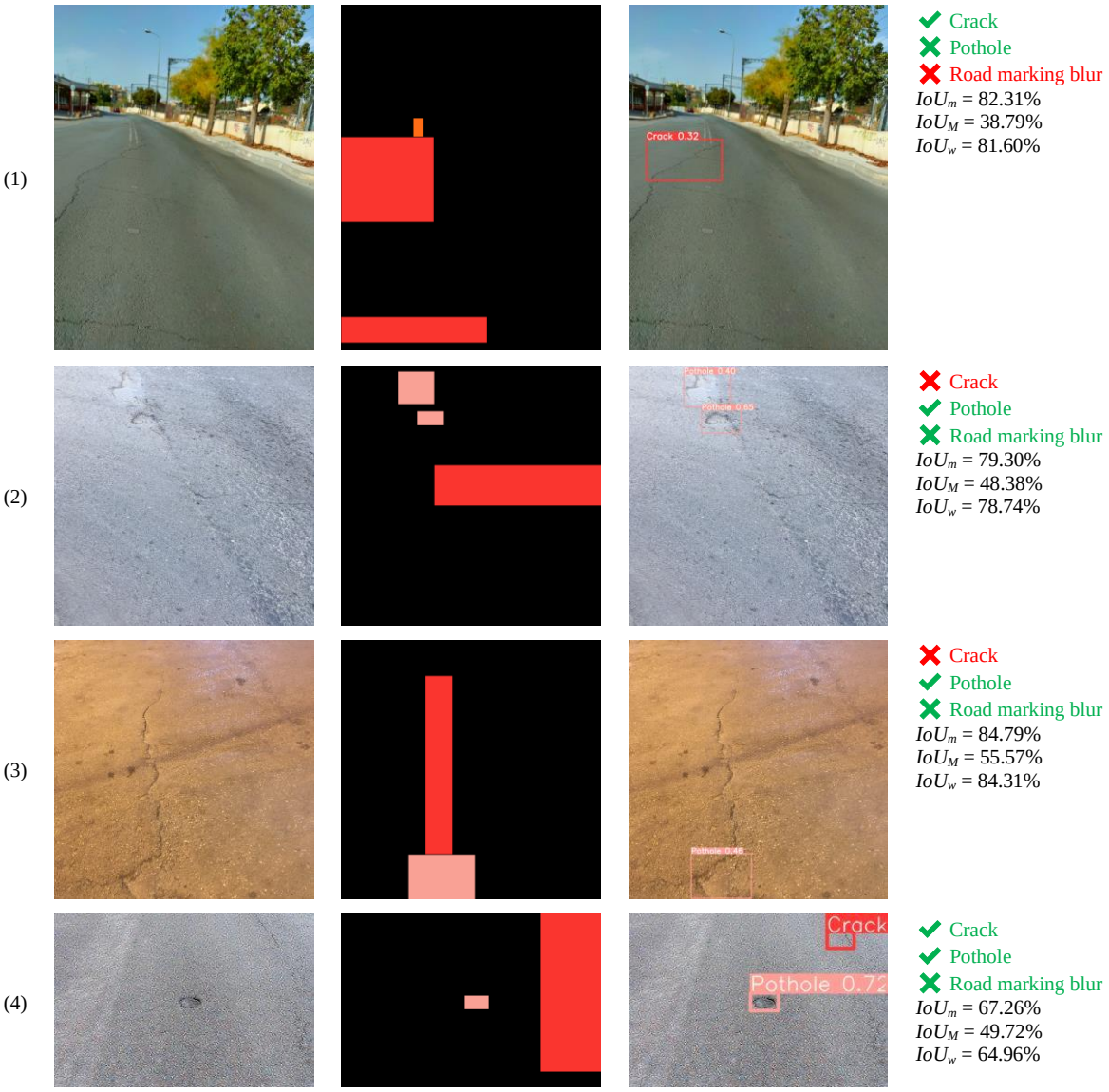
(a) Input RGB image (b) Ground truth (c) Object detection output (d) Performance

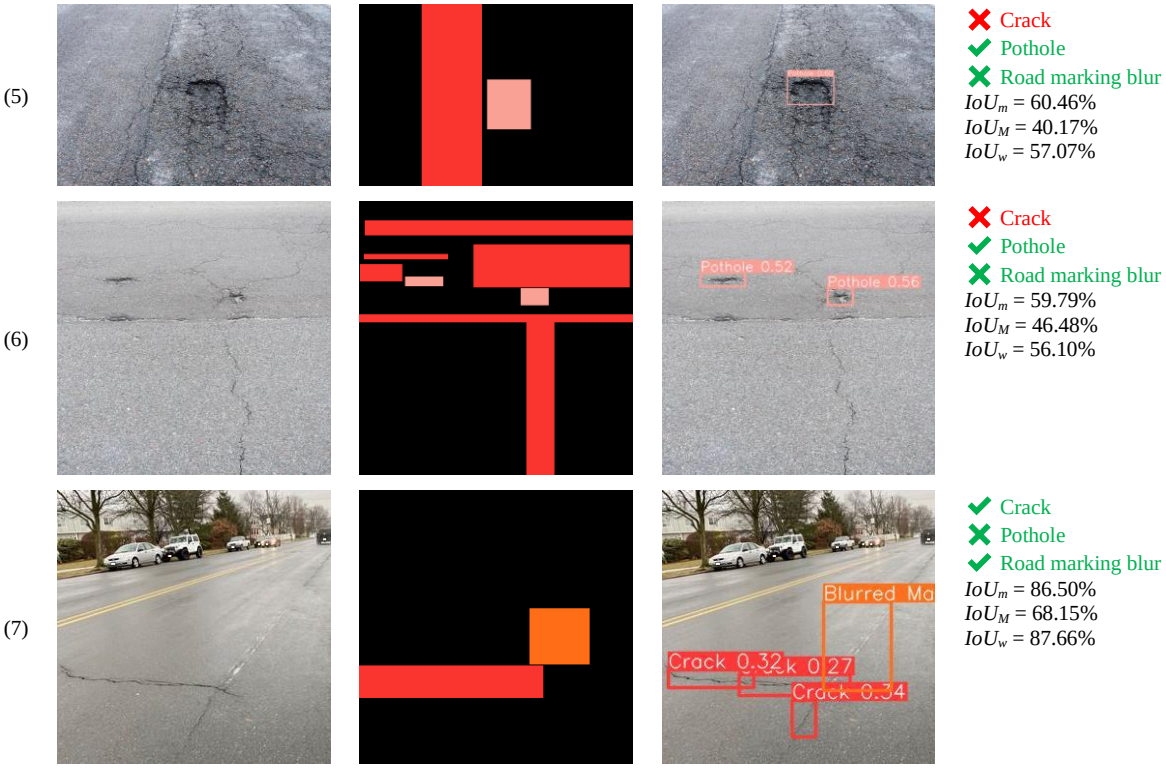
Figure 34: Automated localization of blurred road markings (orange bounding boxes) on the test set of a custom dataset using small YOLOv5 deep model trained on the dataset of [6].

4.5.9 Evaluation of the object detector on road images with more than one defects

In the specific section and in particular, in Figure 35 we demonstrate the detection capabilities of the proposed YOLOv5 architecture in the automated defect (i.e., cracks, potholes, and blurred road markings) detection task. More specifically, Figure 35 presents 7 RGB images of the test set that simultaneously depict more than one category of defects. It is noted that the images are unseen data during the training process of the deep network. As one can see in the aforementioned figure, and in particular in the third and fourth columns, the model showed satisfactory identification and localization performance of the defects.

It is however emphasized that in very rare cases the model failed to detect (false negative) a defect of the image [e.g., Figure 35 (1)-(3),(5)-(6)]. Nevertheless, it is underlined that the input data of the HERON AI system is consecutive RGB frames of a video sequence, and, therefore, even if the detection fails for the current frame, it is highly likely that it will succeed in the next ones. Lastly, the model, as can be seen in Figure 35, did not misclassify an object as a defect (false positive) in any of the test RGB images.





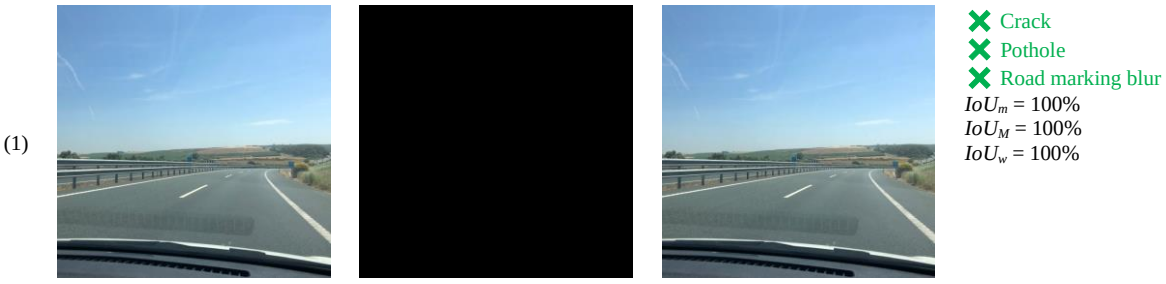
(a) Input RGB image (b) Ground truth (c) Object detection output (d) Performance

Figure 35: Automated localization of (i) cracks (red bounding boxes), (ii) potholes (pink bounding boxes), and (iii) blurred road markings (orange bounding boxes) on challenging images containing more than one road defect using small YOLOv5 deep model trained on the dataset of [6].

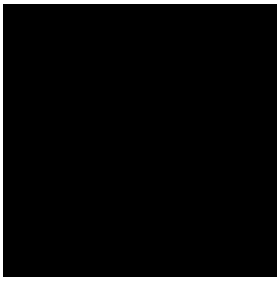
4.5.10 Evaluation of the object detector on road images without road defects

Similar to the previous sections, in Figure 36 we demonstrate the identification capabilities of the proposed YOLOv5 network in the automated defect detection task. More specifically, Figure 36 presents 20 RGB images of the test set that depict road infrastructures without defects. It is noted that the images are unseen data during the training procedure of the model. As one can see in the aforementioned figure, and in particular in the third and fourth columns, the model demonstrated state of the art classification performance, even in challenging images, such as for instance, data that include humidity, as in Figure 36 (5).

It is however noted that in very rare cases the network misclassified an object (false positive) as a road defect [e.g., Figure 36 (2), (8), (13), (15)]. Nevertheless, it is highlighted that the input data of the HERON AI system is consecutive RGB frames of a video sequence, and, thus, even if the automated identification fails for a given frame, it is highly likely that it will succeed in the next ones.

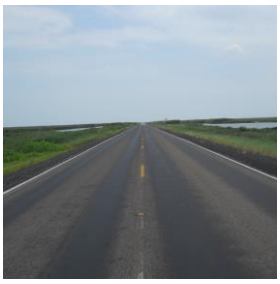
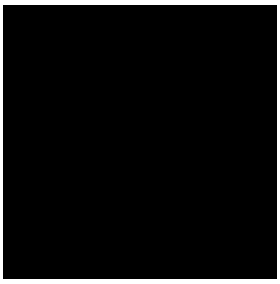


(2)

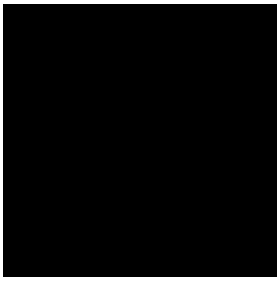

✔ Crack
✘ Pothole
✘ Road marking blur
IoU_m = 97.82%
IoU_M = 49.45%
IoU_w = 98.90%

(3)

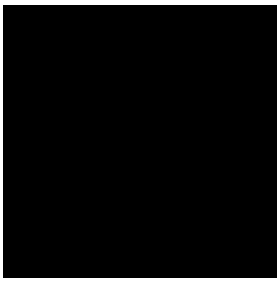

✘ Crack
✘ Pothole
✘ Road marking blur
IoU_m = 100%
IoU_M = 100%
IoU_w = 100%

(4)

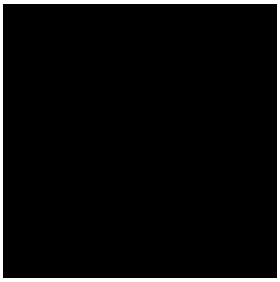

✘ Crack
✘ Pothole
✘ Road marking blur
IoU_m = 100%
IoU_M = 100%
IoU_w = 100%

(5)

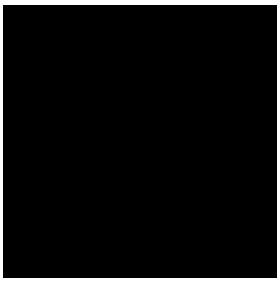

✘ Crack
✘ Pothole
✘ Road marking blur
IoU_m = 100%
IoU_M = 100%
IoU_w = 100%

(6)

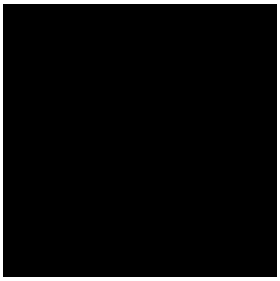

✘ Crack
✘ Pothole
✘ Road marking blur
IoU_m = 100%
IoU_M = 100%
IoU_w = 100%

(7)

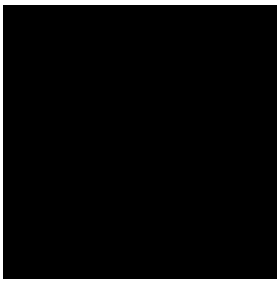

✘ Crack
✘ Pothole
✘ Road marking blur
IoU_m = 100%
IoU_M = 100%
IoU_w = 100%

(8)

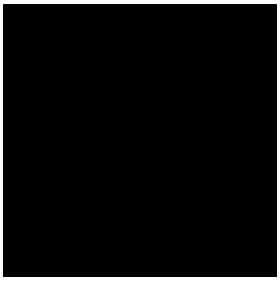

✗ Crack
✗ Pothole
✓ Road marking blur
IoU_m = 96.63%
IoU_M = 49.14%
IoU_w = 98.29%

(9)

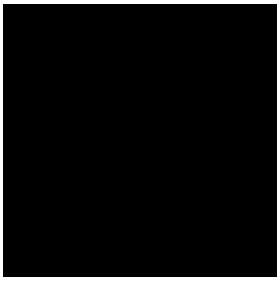

✗ Crack
✗ Pothole
✗ Road marking blur
IoU_m = 100%
IoU_M = 100%
IoU_w = 100%

(10)

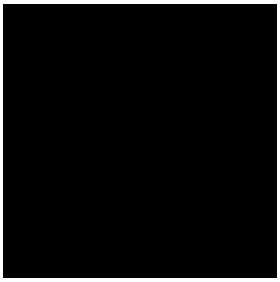

✗ Crack
✗ Pothole
✗ Road marking blur
IoU_m = 100%
IoU_M = 100%
IoU_w = 100%

(11)

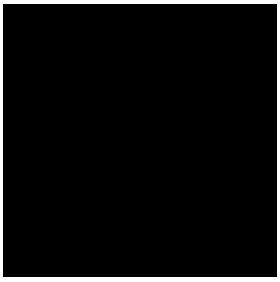

✗ Crack
✗ Pothole
✗ Road marking blur
IoU_m = 100%
IoU_M = 100%
IoU_w = 100%

(12)

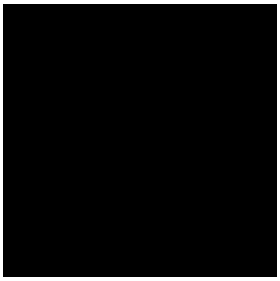

✗ Crack
✗ Pothole
✗ Road marking blur
IoU_m = 100%
IoU_M = 100%
IoU_w = 100%

(13)

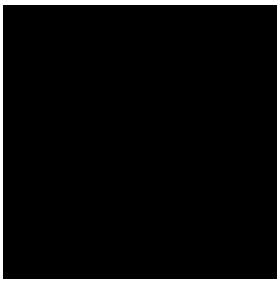

✗ Crack
✗ Pothole
✓ Road marking blur
IoU_m = 95.25%
IoU_M = 48.78%
IoU_w = 97.57%

(14)

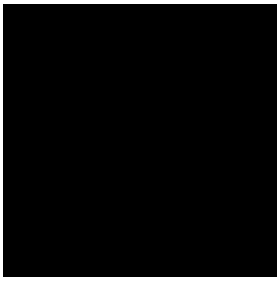

✗ Crack
✗ Pothole
✗ Road marking blur
IoU_m = 100%
IoU_M = 100%
IoU_w = 100%

(15)

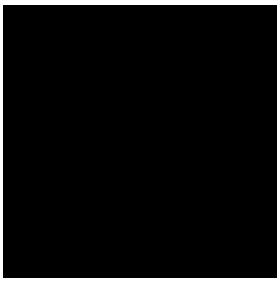

✗ Crack
✗ Pothole
✔ Road marking blur
IoU_m = 59.37%
IoU_M = 37.25%
IoU_w = 74.51%

(16)

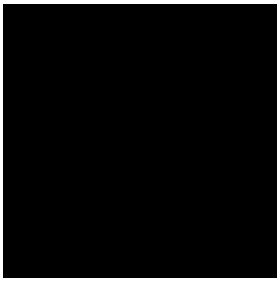

✗ Crack
✗ Pothole
✗ Road marking blur
IoU_m = 100%
IoU_M = 100%
IoU_w = 100%

(17)

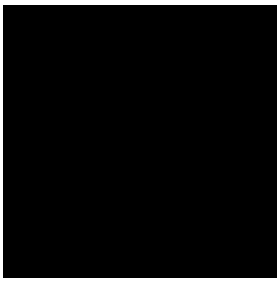

✗ Crack
✗ Pothole
✗ Road marking blur
IoU_m = 100%
IoU_M = 100%
IoU_w = 100%

(18)


✗ Crack
✗ Pothole
✗ Road marking blur
IoU_m = 100%
IoU_M = 100%
IoU_w = 100%

(19)


✗ Crack
✗ Pothole
✗ Road marking blur
IoU_m = 100%
IoU_M = 100%
IoU_w = 100%

(20)

(a) Input RGB image (b) Ground truth (c) Object detection output (d) Performance

Figure 36: Evaluation of the YOLOv5 object detector on road images without defects.

4.6 Real-time road surface monitoring using UAV images

Usually, inspections of civil engineering structures, such as road infrastructures, are carried out by technicians utilizing rope and harness access equipment, in conjunction with construction machineries such as lifts and cranes. These traditional inspection techniques not only pose safety risks, that may lead to worker injuries and accidents, but are also costly and time-consuming. Furthermore, they require heavy machinery that results in hindrances in road and waterway traffic. It is also noted that the identification of the corroded areas is performed using visual methods, potentially leaving damage unnoticed in the inaccessible areas of the structures. Therefore, it is mandatory to adopt innovative inspection methods, through which efficient defect identification is promoted, while in parallel the workers' safety is ensured.

On this basis, unmanned aerial vehicles (UAVs) offer several advantages in processes that involve remote sensing data acquisition. More specifically, by exploiting drone technology we are able to remotely, and therefore safely, collect data from otherwise virtually or physically unreachable areas. Also, we can effectively gather timely and on-demand images [11], by avoiding short-term traffic arrangements, that require time-consuming permits and result in traffic jams, shutdowns, accidents, and CO_2 emissions. Hence, it is underlined that UAVs are emerging as a suitable and cost-effective method for gathering high-quality image data, that encompass key spatial, textural, and chromatic information of the under-inspection structure [3].

HERON addresses existing limitations in maintenance and upgrading by incorporating robot-assistive RI processes that (i) increase automation in the maintenance process, (ii) minimize traffic delays during maintenance, and (iii) improve workers' safety and avoidance of weather hazards. HERON provides effective and faster repairs for RIs and also supports prefabrication strategies that jointly, will reduce the maintenance or upgrading cost and time needed to complete a task, innovates on networking solutions for RI management, exploits drones for inspecting larger areas and adopts novel ML tools which can transform traditional RI to intelligent assets. HERON concepts can stimulate new procedures for managing and operating RI, and leverage previous knowledge on robotics, vision systems, ML, sensing and monitoring systems and automated inspection.

The use of drones will favor the monitoring of RI that are difficult to access, and in the case of road maintenance, they will allow having a current model of defects to plan automated actions for the next day's maintenance tasks, avoiding visual inspection of the personnel (driving vehicles and walking on the road) and therefore, possible accidents. Drone technology will make it possible to reduce the overall cost of these costly interventions. Consequently, the use of aerial drones within the HERON system can provide the bigger picture of the area under maintenance or/and upgrading intervention procedure.

4.6.1 Object detection model

As in the previous sections, in which the automated traffic cone (see Section 4.4) and defect (see Section 4.5) detection of the HERON system were presented, the state-of-the-art object detection frameworks were analyzed, evaluated and compared for the problem of road defect detection from UAV imagery. Again, the YOLOv5 model was chosen as the basic framework for the current work, the architecture of which is illustrated in Figure 19. Thereby, the focus of this section is to automate the identification of two main types of road damage, i.e., potholes

and cracks, by utilizing one of the state-of-the-art computer vision frameworks, YOLOv5, on image data that derive from RGB sensors mounted on UAVs.

4.6.2 Dataset description

In order to train and evaluate the YOLOv5 model the dataset that is presented in the work of [51] was utilized. The data (see Figure 37) was created in order to represent the situation of the Spanish roads and automate the detection of two main types of road damage, i.e., potholes and cracks. The dataset utilized for the evaluation of the results of the specific scientific article [51] has been published at https://github.com/luisaugustos/Pothole-Recognition.

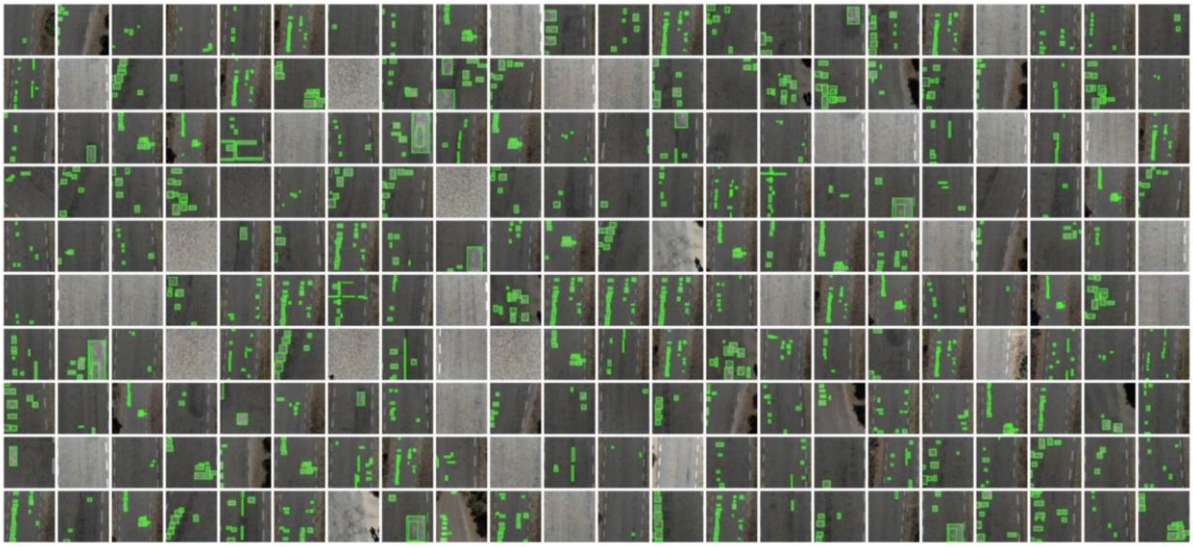


Figure 37: Sample images from the dataset [51] that contain UAV images for crack and pothole recognition.

In particular, initially, it contained 568 labeled road images, with a resolution of 3840×2160 pixels, from RGB sensors mounted on a UAV. After the pre-processing process, the total number of labeled images in the dataset was 1,362 images. More specifically, the following pre-processing process was applied to each RGB image:

- Auto-orientation of pixel data (with EXIF-orientation stripping)
- Resize to 1200×900 [Fill (with center crop)]

Furthermore, in order to generalize the detection capabilities of the trained model, the following augmentation process was applied in order to create three versions of each source UAV image:

- 50% probability of horizontal flip
- 50% probability of vertical flip
- Random rotation of between -15 and +15 degrees
- Salt and pepper noise was applied to 5 percent of the pixels

Thereby, after the preprocessing procedure among the 1,362 UAV images, 70% were used for training (1,191 images), 20% for validation (114 images), and 10% for testing (57 images) the detection capabilities of the trained deep model.

4.6.3 Experimental setup - Model training

The YOLO object detector was trained and evaluated using an NVIDIA Tesla K80 GPU with 12 GB of memory, provided by Google Colab. We trained the network, using batches of size 32, for 200 epochs, and set the input image resolution to 448×448 pixels. This work is based on the YOLOv5 small model in order to reduce the computational cost of the detection task. Towards this direction, the network takes up less than 15 MB of storage and thus can be easily embedded in smartphone applications and various low-memory digital devices or systems, including drones and microcontrollers.

4.6.4 Evaluation metrics

Regarding the detection task, similarly to Sections 4.4.4 and 4.5.4, we will utilize the Intersection over Union (IoU), which is a popular evaluation metric used to measure the accuracy of an object detector on a particular dataset. In parallel, regarding the classification task of the road defects on a given UAV image the performance of the implemented architecture is evaluated in terms of three metrics as follows: (i) precision [see eq. (1) in Section 4.5.4], (ii) recall [see eq. (2) in Section 4.5.4], and (iii) F1-score [see eq. (3) in Section 4.5.4].

4.6.5 Experimental validation

The performance of the object detection task is illustrated in Figure 38 with a confidence level of 95% over the data of the test set. Regarding the computational complexity, the model needs an average time of 0.059 seconds to identify the road defects on a UAV image. In parallel, the classification capabilities in terms of the performance metrics that were demonstrated in Section 4.6.4 are shown in Figure 39 as well as Figure 40.

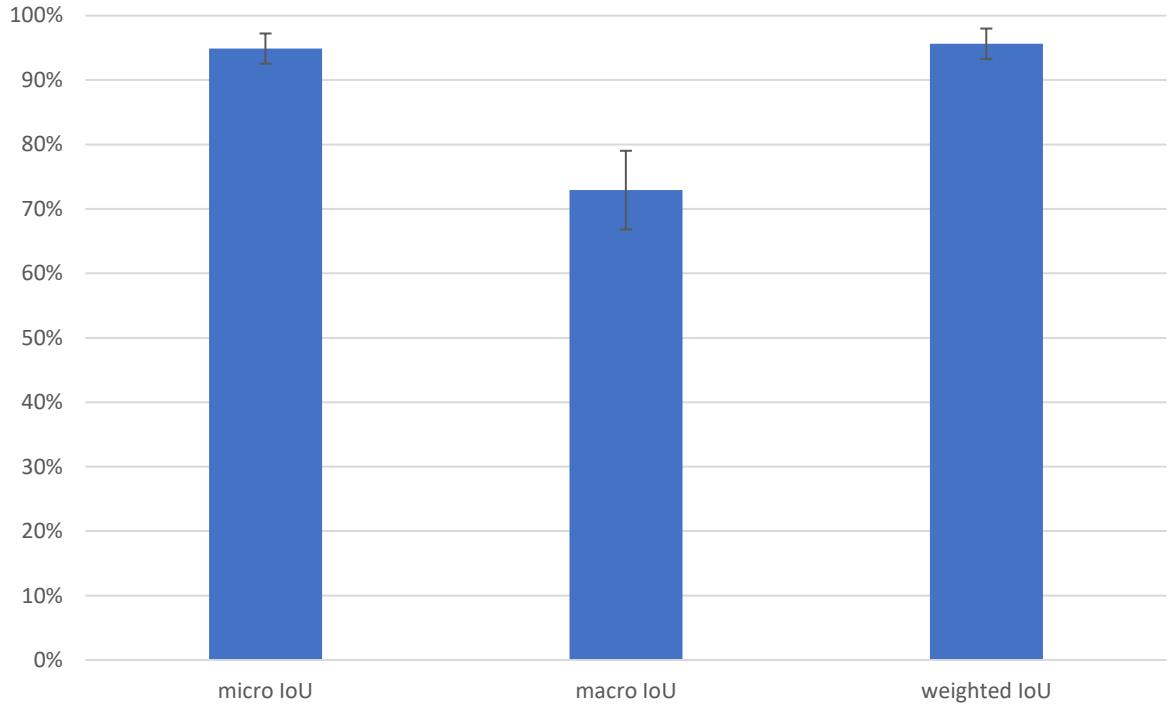


Figure 38: Micro, macro and weighted IoU scores.

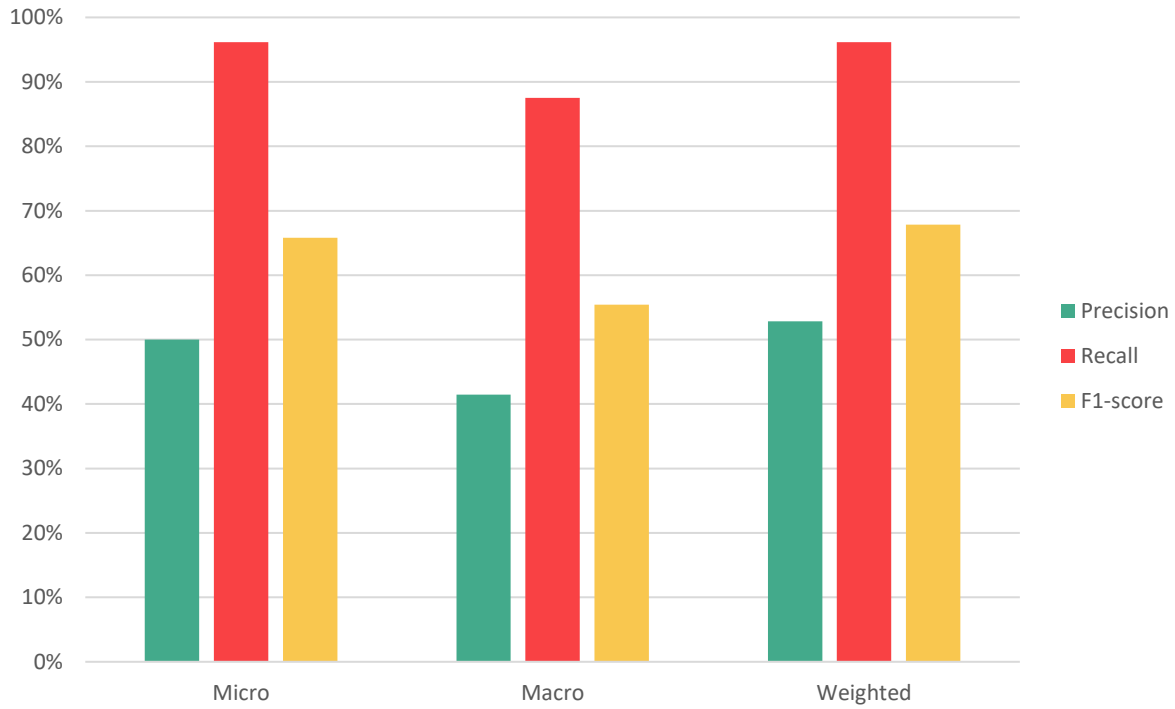


Figure 39: Classification scores calculated with micro, macro, and weighted averaging, respectively.

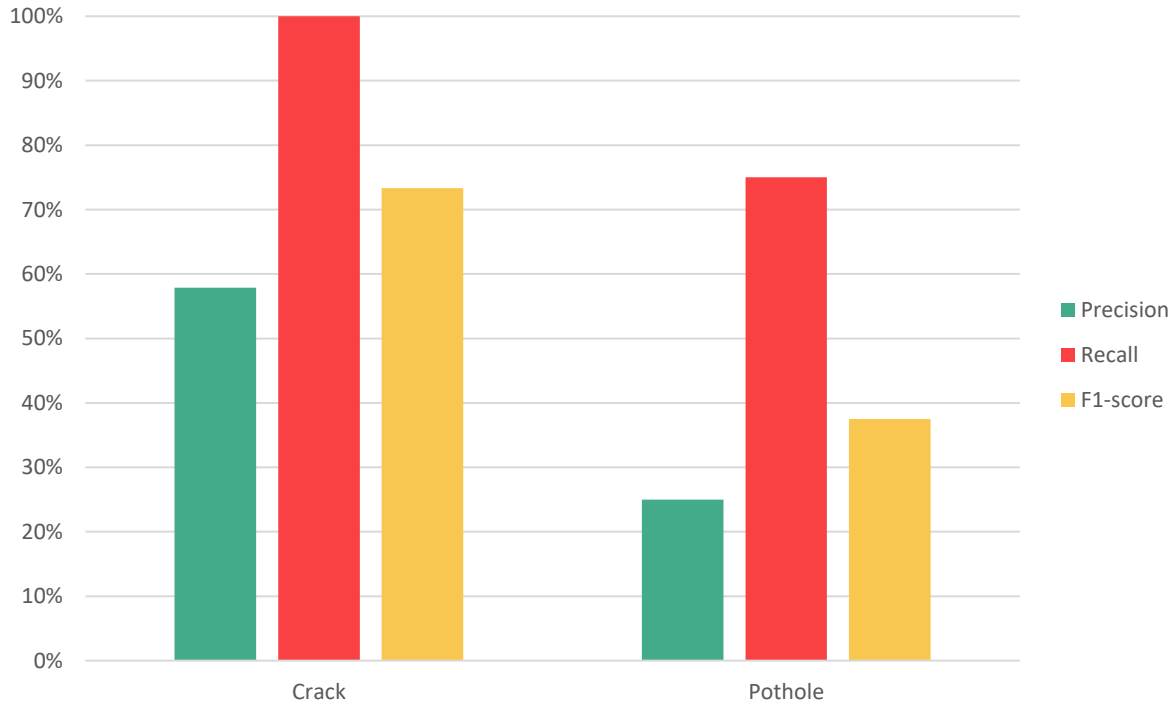


Figure 40: Classification scores calculated per class.

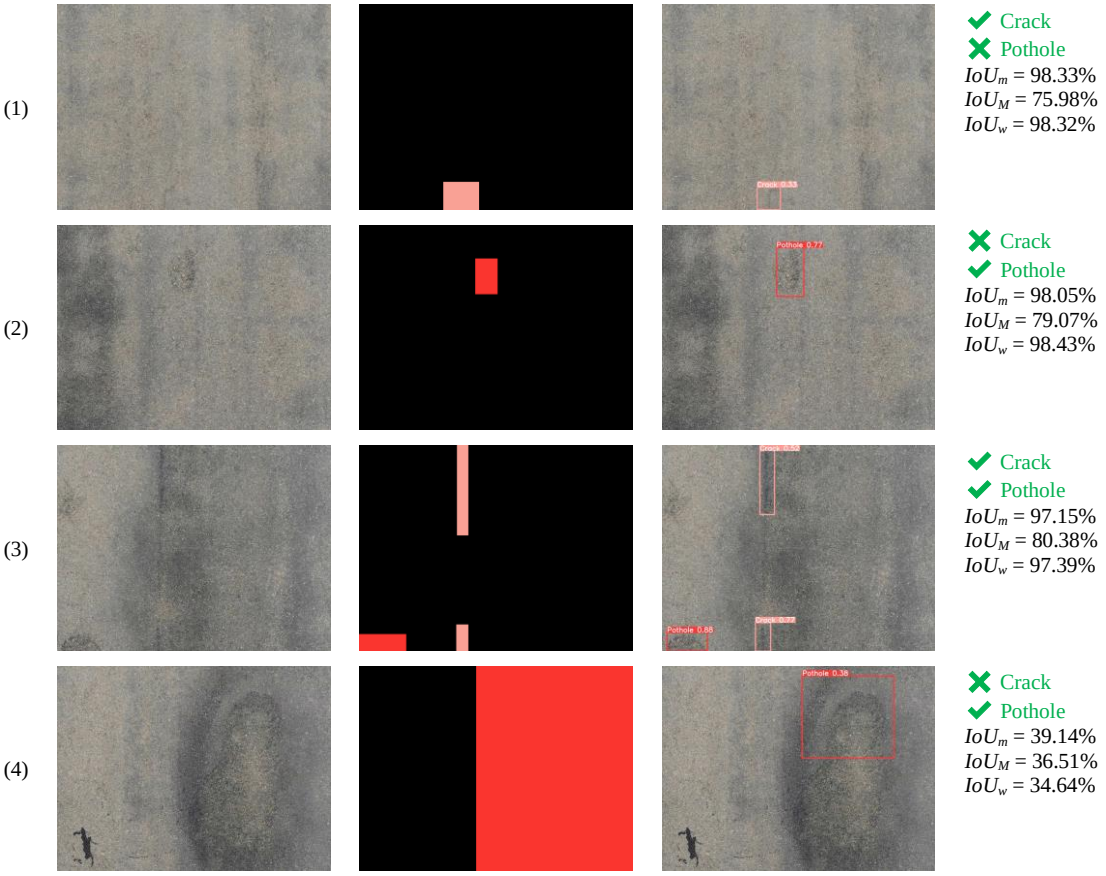
Consequently, in this section, a computer vision framework, which utilizes the YOLOv5 detector and drone images, was demonstrated. The proposed system can classify and localize two classes of road defects (i.e., cracks and potholes), in the processed UAV imagery. The final network was able to demonstrate an IoU score up to 95.64% for the detection task and an F1-score up to 67.82% for the classification task with precision and recall scores of 52.83% and 96.15%, respectively.

4.6.6 Evaluation of the object detector on UAV images with cracks and potholes

In this section, we present the experimental results that the YOLOv5 model demonstrated during the evaluation process. More specifically, in Figure 41 one can observe the automated identification capabilities of the proposed YOLOv5 architecture in the automated crack and pothole detection task from UAV images. The aforementioned figure shows 25 indicative drone images of the test set and in particular, the first column corresponds to the original RGB drone images followed by their ground truth bounding boxes in the second column. The third column shows the predicted bounding boxes with their corresponding confidence scores. Finally, the last column demonstrates the performance of the model on the classification task as well as the IoU scores of the respective input image regarding the defect detection task.

To effectively explore the performance of the model, the test images can contain: (i) only cracks [e.g., Figure 41 (1)], (ii) only potholes [e.g., Figure 41 (2)], (iii) both cracks and potholes [e.g., Figure 41 (3)], (iv) healthy asphalt surface without degradation [e.g., Figure 41 (15)]. It is noted that the images are unseen data during the training process of the deep model. As one can see in the aforementioned figure, and in particular in the third and fourth columns, the model showed satisfactory recognition and localization performance of the cracks and potholes.

It is however noted that in rare cases the network failed to identify (false negative) a defect of the drone image [e.g., Figure 41 (24)-(25)]. In parallel, in rare cases, the model misclassified an object (false positive) as a defect [e.g., Figure 41 (23)]. Nevertheless, it is emphasized that the input data of the HERON AI system is consecutive RGB frames of a video sequence, and, therefore, even if the detection fails for the current frame, it is highly likely that it will succeed in the next ones.



(5)

✗ Crack
✓ Pothole
IoU_m = 99.66%
IoU_M = 95.51%
IoU_w = 99.68%

(6)

✓ Crack
✓ Pothole
IoU_m = 81.77%
IoU_M = 64.32%
IoU_w = 82.03%

(7)

✗ Crack
✓ Pothole
IoU_m = 92.57%
IoU_M = 86.11%
IoU_w = 93.11%

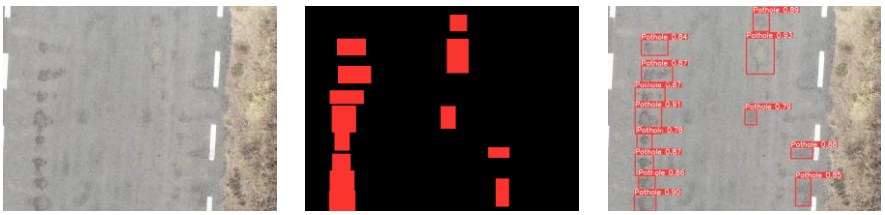
(8)

✗ Crack
✓ Pothole
IoU_m = 98.07%
IoU_M = 84.95%
IoU_w = 98.31%

(9)

✗ Crack
✓ Pothole
IoU_m = 96.31%
IoU_M = 87.54%
IoU_w = 96.47%

(10)



✗ Crack
✓ Pothole
IoU_m = 95.19%
IoU_M = 87.96%
IoU_w = 95.35%

(11)

✗ Crack
✓ Pothole
IoU_m = 95.80%
IoU_M = 79.86%
IoU_w = 95.90%

(12)

✓ Crack
✓ Pothole
IoU_m = 91.05%
IoU_M = 62.29%
IoU_w = 93.42%

(13)

✗ Crack
✓ Pothole
IoU_m = 98.28%
IoU_M = 90.20%
IoU_w = 98.34%

(14)

✗ Crack
✓ Pothole
IoU_m = 98.27%
IoU_M = 88.39%
IoU_w = 98.39%

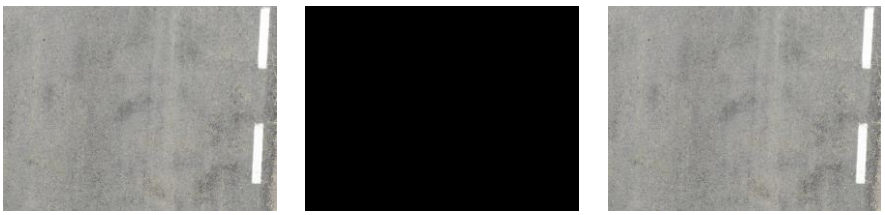
(15)

✗ Crack
✗ Pothole
IoU_m = 100%
IoU_M = 100%
IoU_w = 100%

(16)

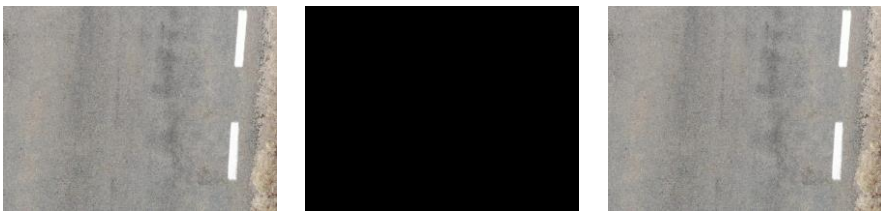
✗ Crack
✗ Pothole
IoU_m = 100%
IoU_M = 100%
IoU_w = 100%

(17)



✗ Crack
✗ Pothole
IoU_m = 100%
IoU_M = 100%
IoU_w = 100%

(18)

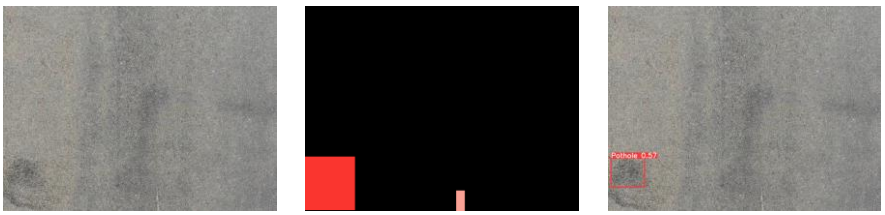


✗ Crack
✗ Pothole
IoU_m = 100%
IoU_M = 100%
IoU_w = 100%

(19)

✗ Crack
✗ Pothole
IoU_m = 100%
IoU_M = 100%
IoU_w = 100%

(20)



✓ Crack
✓ Pothole
IoU_m = 93.53%
IoU_M = 44.37%
IoU_w = 93.43%

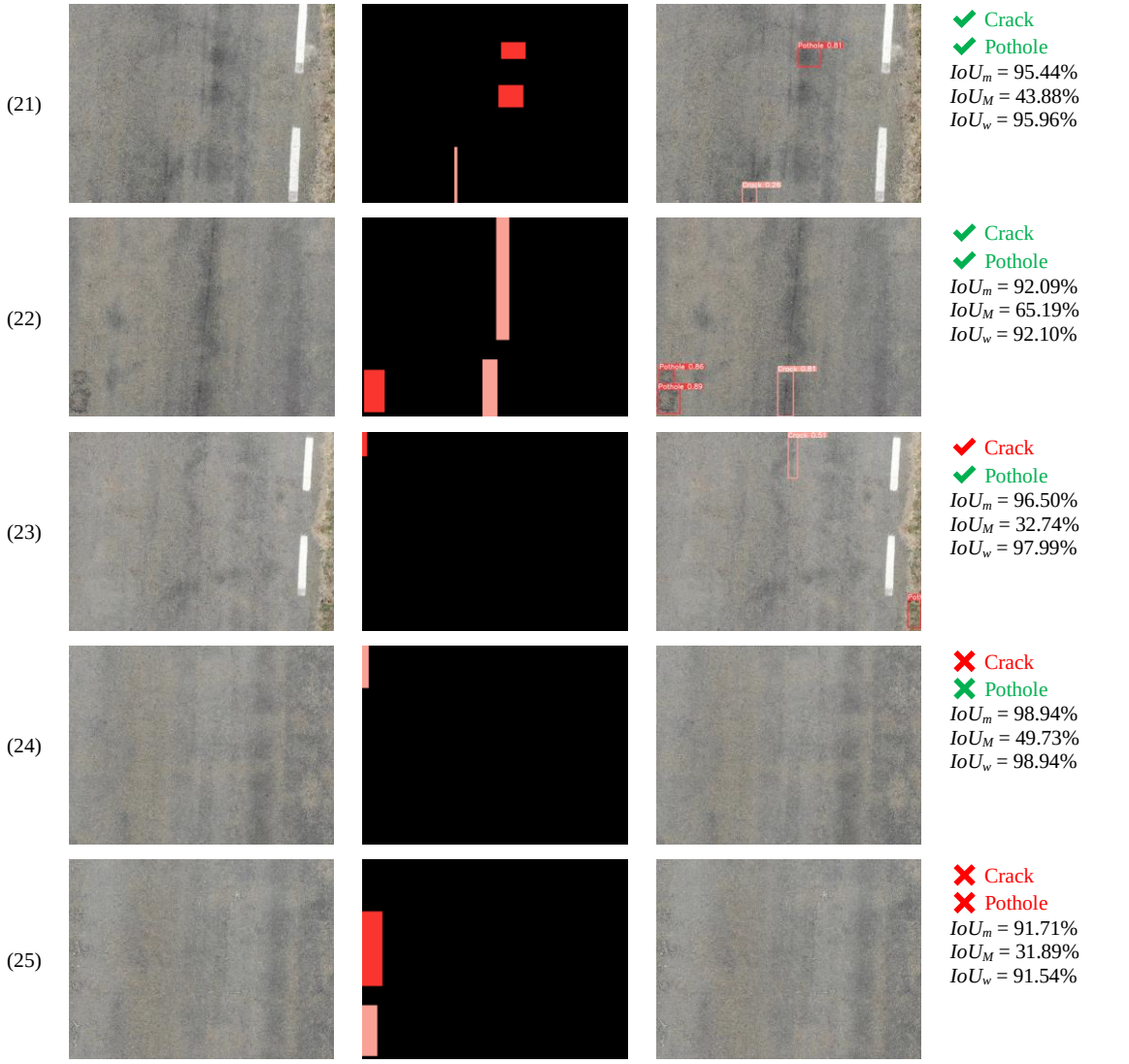


Figure 41: Automated localization of (i) cracks (pink bounding boxes) and (ii) potholes (red bounding boxes) on UAV images using small YOLOv5 deep model trained and tested on the dataset of [50].

4.7 Pixel-level crack semantic segmentation

As presented in the previous sections, object detection tasks provide a rough localization of the target objects or defects, and they are the most important because of the instantiation they achieve (distinguishing between different instances of targets). However, object detection does not provide precise localization at a pixel level, which is required in some cases, such as crack detection. Thus, by applying semantic segmentation, the precise location and direction of cracks can be acquired and consecutively be provided as guidance to the robot for repairing them.

Many state-of-the-art deep learning models exist in both object detection and semantic segmentation, and even in various combinations of them, such as Mask-RCNN [18]. A model such as this though, cannot be utilized due to a lack of training data in the scenario of crack detection. Thus, separate models for detection and segmentation will be used because of the separate relevant training data.

4.7.1 Dataset description

The dataset that we deemed to be useful for the solution of the crack detection problem is "Cracks and Potholes in Road Images Dataset" by Passos et al [38]. This is a publicly available dataset, and it was developed using images made available by the Brazilian National Department of Transport Infrastructure. It contains images of defects (cracks, potholes) in asphalted roads in Brazil, and it was made in order to be used for a study on the detection of cracks and potholes in asphalted roads, using texture descriptors and machine learning algorithms such as Support Vector Machine, K-Nearest Neighbors and Multi-Layer Perceptron Neural Network. The contained images are from highways in the states of Espírito Santo, Rio Grande do Sul, and the Federal District. They were selected manually, following criteria such as not showing signs of vehicles and people, as well as not having image defects. This work consists of 2,235 samples of roads where each produces 1 image and 3 masks that delimit the vehicle's path and crack and pothole defects (see Figure 42).

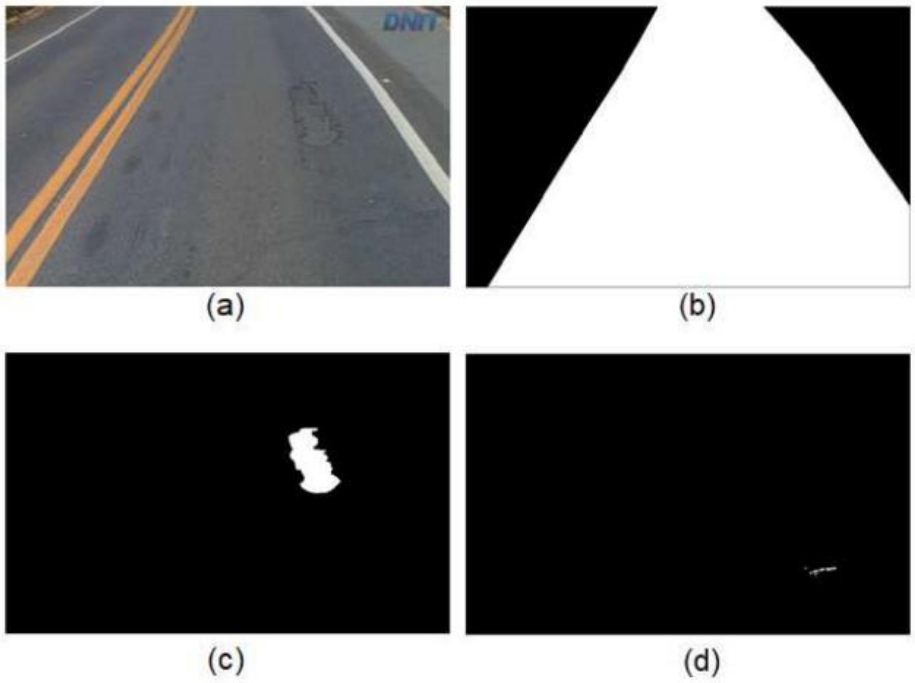


Figure 42: Example of the original image (a) and the masks corresponding to the road region (b), pothole (c), and crack (d).

The dataset images were extracted from videos captured by an HD camera of 1280x729 resolution and 16:9 aspect ratio, mounted on a Highway Diagnostic Vehicle (HDV). The camera is installed on the highest part of the vehicle, facing the front and with an inclination closer to orthogonality. Thus, the visibility of the pavement is 15 meters. This camera captures images with a minimum resolution of 4 megapixels, every 5 meters away. The setup can be seen in the following figure below (see Figure 43).

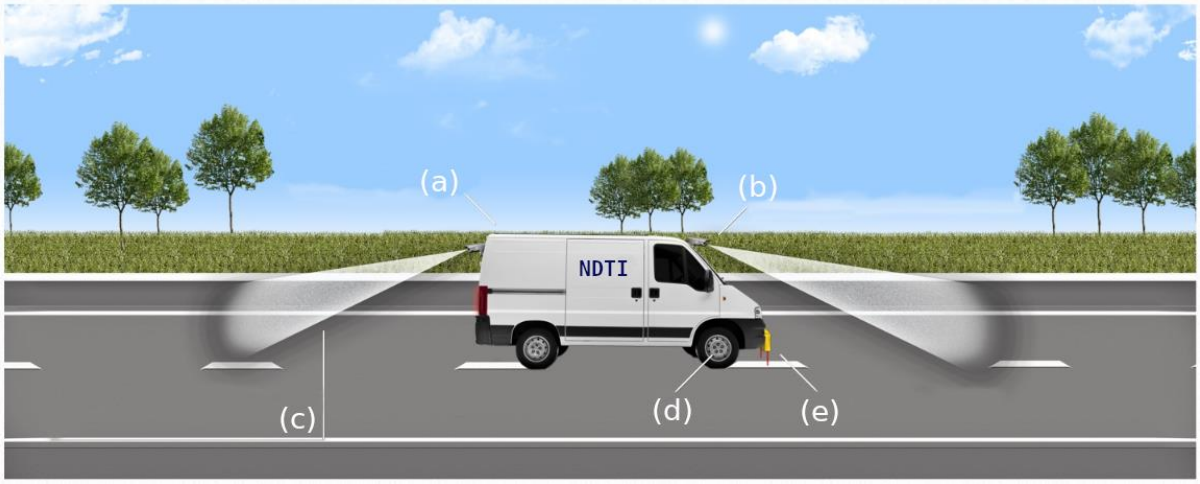


Figure 43: Representation of the HDV used by NDTI: (a) satellite tracking system (b) high-resolution camera (c) recording cameras (d), precision odometer and (e) laser sensors.

To check the feasibility of the dataset and its capacity for properly training a neural network, at the time of writing only the crack annotations have been used. In further work, if the segmentation of potholes is deemed useful, they will be included in the training.

4.7.2 Semantic segmentation model

For the segmentation task with the pre-mentioned dataset, U-net architecture has been chosen as described in Section 4.1.3. Though not the most advanced architecture currently available, it is a baseline of architectures that fit this specific task and thus, is suitable for the first step, if not achieving the required goal.

U-net can be effective in situations where the segmentation annotation of some classes is sparse and the training images are few, which is exactly the case with this dataset. Because annotation happens on pixel level and cracks occupy only a few hundred pixels per image, the crack and non-crack class ratio is less than 99.9%. This is still a heavily unbalanced dataset, thus image augmentations and modifications of the base architecture have been applied, which will be explained below.

To begin with, the model that we have trained consists of a Fully Connected Network (FCN) as the base model (head), and a Unet-s5-d16 as the backbone (encoder-decoder). The input image size is 1024×640 pixels, and it goes through various augmentations. These include random resizing, random crop of 256×256 pixels, random image flip, photometric distortions and normalization. As Loss function, Focal Loss have been used in combination with weighted Dice Loss with class weights of 0.01 and 0.99 for non-cracks and cracks classes respectively.

4.7.3 Evaluation metrics

To evaluate properly the performance of the training, two metrics has been used. Sørensen–Dice coefficient (F1 score) and Accuracy.

Pixel Accuracy is a simple metric, but it gives an easy evaluation of the results. It is defined as the percent of pixels in your image that is classified correctly. It does not always reflect the real performance of the training as it fails to include some edge cases. Thus, other more robust metrics are used too.

Precision and recall (see Figure 44) are other two metrics that are widely used in computer vision detection and segmentation tasks. Precision can be defined as the fraction of correctly identified pixels out of all the pixels of the respective class. Recall, on the other hand, can be defined as the percentage of pixels that belong to a specific class and have been correctly retrieved.

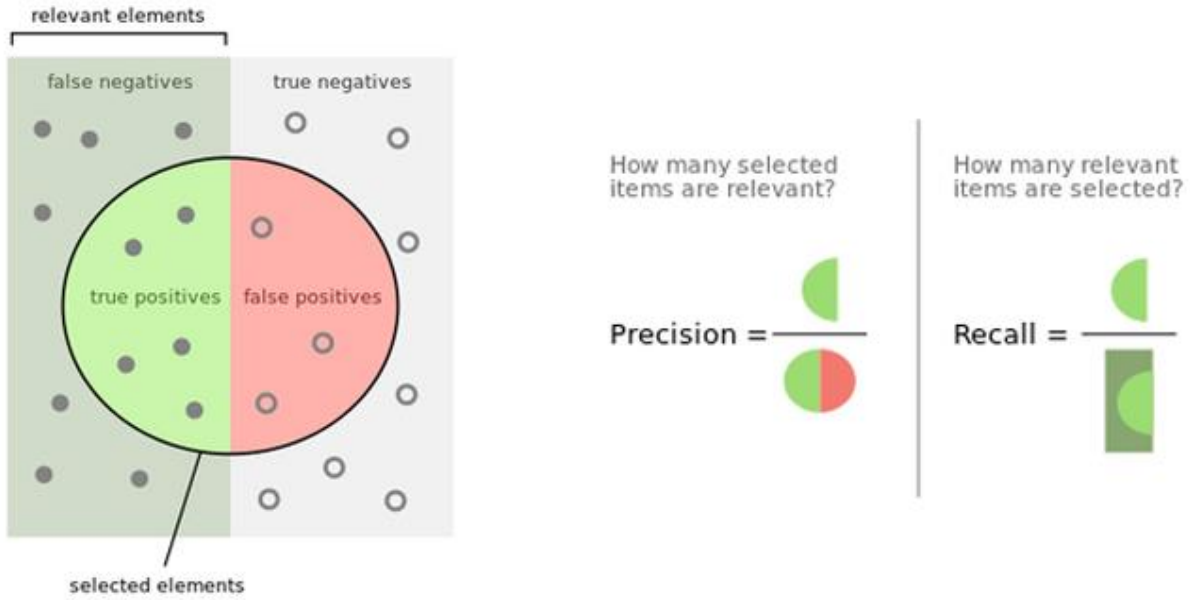


Figure 44: Precision and recall.

The dice coefficient is defined as twice the area of overlap divided by the total number of pixels in both images (see Figure 45). It can also be defined as the harmonic mean of the precision and the recall metrics. It is a popular metric used in segmentation tasks alongside the IoU metric (see Figure 25).

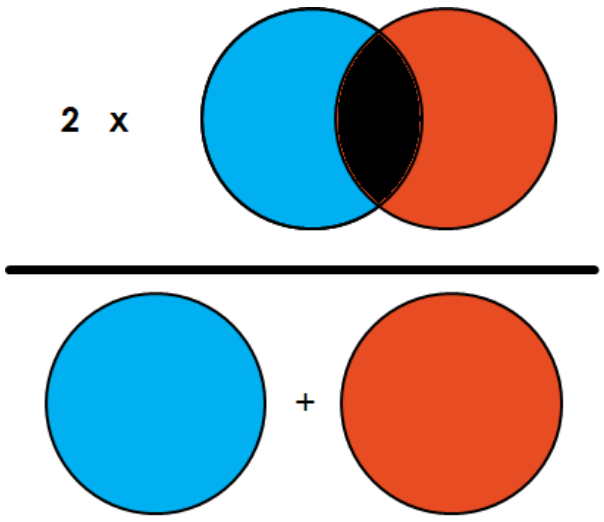


Figure 45: Illustration of Dice Coefficient. 2×Overlap/Total number of pixels.

4.7.4 Model training and results

This setup has been trained for 200 epochs, with a gradient descent optimizer, at a learning rate of 0.01, with a momentum of 0.9, and a weight decay of 0.0005. The dataset has been split into 80% training set and 20% test set.

The overall performance of the training can be observed in Table 11. The averaging happens between the two classes, "cracks" and "background". Although the average values are promising for this kind of task, do not reflect the actual target of the task, which is only crack segmentation. Table 12 shows the values only relevant to the "crack" class, which can give us better insights for correction.

Table 11: Average metrics of the current training setup on the test dataset.

Average accuracy	Average F1-score	Average Precision	Average Recall
97.51%	61.35%	58.13%	69.85%

Table 12: Average metrics for each class on the test dataset.

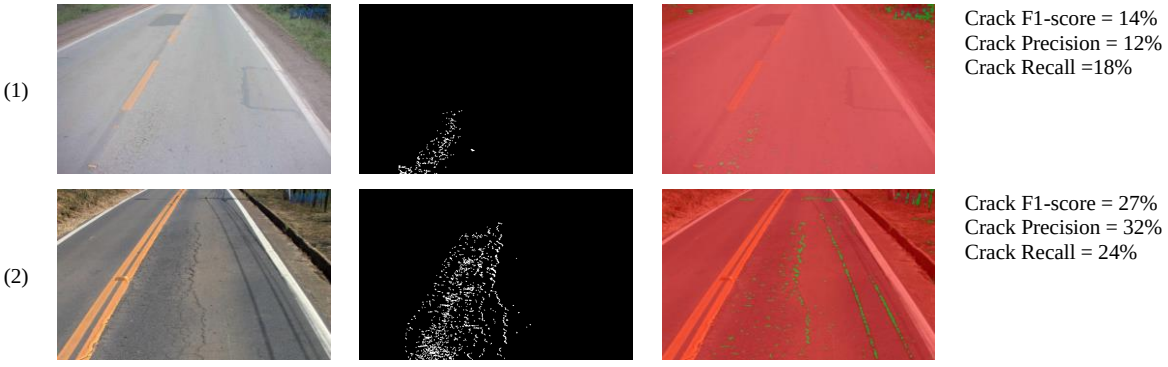
Class	F1-score (Dice)	Precision	Recall
Background	98.73%	99.44%	98.04%
Crack	23.97%	16.83%	41.65%

4.7.5 Evaluation of the U-Net model

In this subsection, and in particular in Figure 46, a small sample of the test set will be shown along with its ground truth mask and the prediction of the trained crack segmentation model. The individual metric results will also be quoted for each picture.

Noticeable is the fact that the metric measures greatly vary between these examples (see Figure 46). There are cases where the recall rate reaches 90% (sample 6), meaning that most of the cracks have been found, while others fail to go above 7% (sample 4). This seems to happen when the annotations are very fine and few in number. These cases can be improved in future work by changing the type of the neural network and probably using a deeper one that can comprehend finer details.

An additional problematic situation that drops the average precision score, is where large groups of pixels are wrongly identified as cracks, for example, shadows; as seen in the 9th sample. This probably happens due to unoptimized hyperparameters or due to the current configuration of the loss functions. Further changes and configurations as well as new image augmentations will be tested out to improve the results, eliminate the identified problems, and reach the required target metrics.



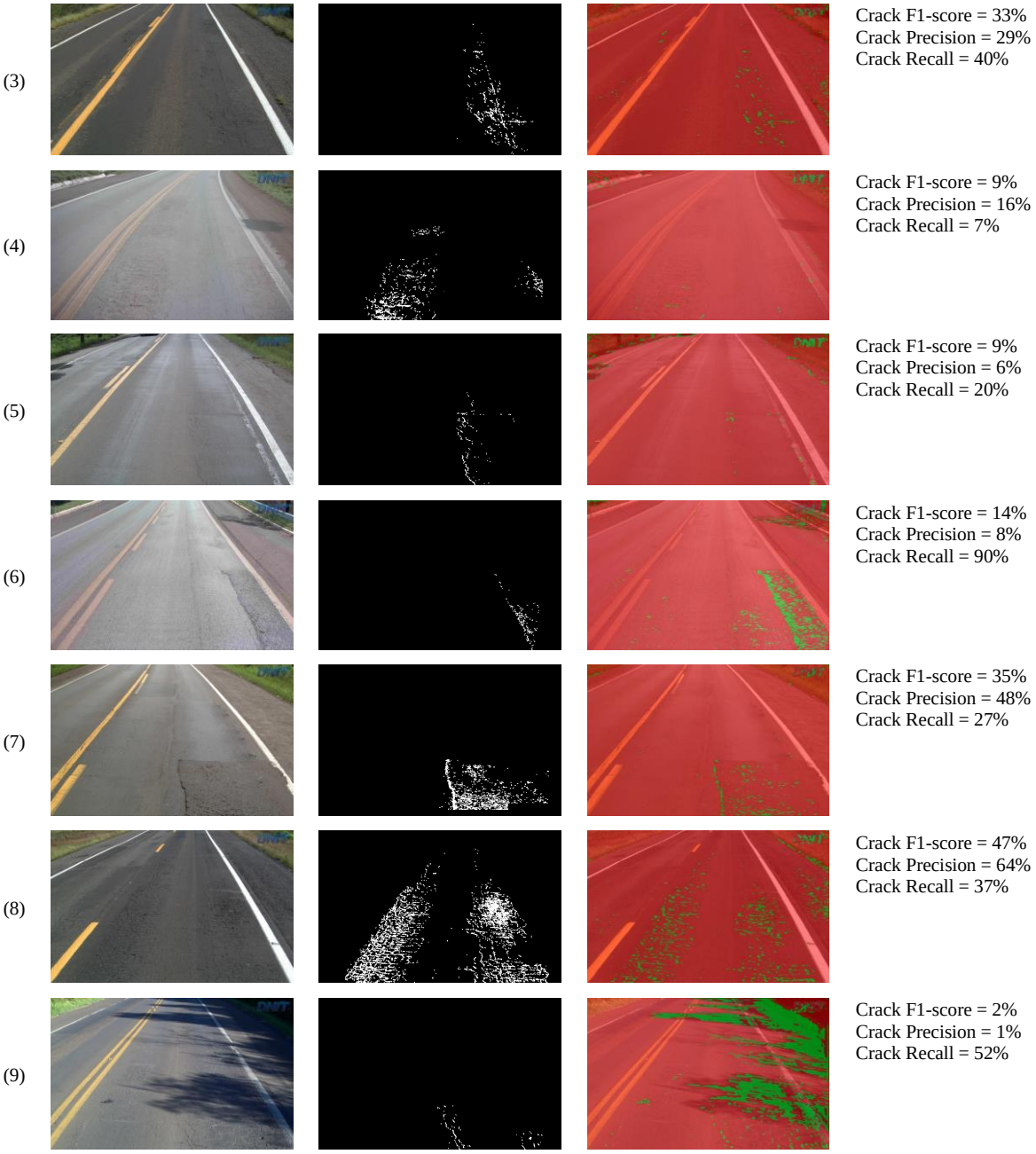


Figure 46: Evaluation of the U-Net model on road images with crack defects.

5 Conclusions

This deliverable report, namely "AI - driven image segmentation and feature extraction", demonstrates insights related to the automated continuous monitoring of the road infrastructure. This report is a compilation of the work that was completed in the framework of task 3.1 "AI-driven image segmentation and feature extraction". In particular, the work focused on specific deep machine learning toolkits that are fed with optical data coming from various sensors (e.g., RGB, stereo cameras) and have been developed for feature representation of the various HERON maintenance and upgrading tasks, such as for instance, potholes, blurred road markings, and cone detection as well as crack localization and segmentation.

Developed methodologies rely on optical data deriving from fixed optical sensors (e.g., RGB, stereo cameras), mounted on a maintenance robotic vehicle and/or aerial inspection drones. It is noted that the UGV is utilized for the detailed inspection of the road surface as well as the robotic maintenance interventions, whereas the UAV could monitor the general whole procedure and give the "big-picture" of the intervened area by proving additional insights regarding the state of the road corridor.

In this report, various state-of-the-art computer vision approaches have been developed in order to efficiently address the automated inspection of the road infrastructures. The employed schemes involve image classification (for the identification of the deterioration category), object detection (for the localization of the corresponding defect), and semantic segmentation (e.g., pixel-wise classification of crack defects) in order to provide a detailed and effective automated assessment of the road infrastructure state.

References

[1] Akarsu, B., KARAKÖSE, M., PARLAK, K., Erhan, A. K. I. N., & SARIMADEN, A. (2016). A fast and adaptive road defect detection approach using computer vision with real time implementation. International Journal of Applied Mathematics Electronics and Computers, (Special Issue-1), 290-295.

[2] Albaráñez Martínez, J., Llopis-Ibor, L., Hernández-García, S., Pineda de Luelmo, S., & Hernández-Ferrándiz, D. (2022). A Case of Study on Traffic Cone Detection for Autonomous Racing on a Jetson Platform. In Iberian Conference on Pattern Recognition and Image Analysis (pp. 629-641). Springer, Cham.

[3] Ammour, N., Alhichri, H., Bazi, Y., Benjdira, B., Alajlan, N., & Zuair, M. (2017). Deep learning approach for car detection in UAV imagery. Remote Sensing, 9(4), 312.

[4] Arnold, E., Al-Jarrah, O. Y., Dianati, M., Fallah, S., Oxtoby, D., & Mouzakitis, A. (2019). A survey on 3d object detection methods for autonomous driving applications. IEEE Transactions on Intelligent Transportation Systems, 20(10), 3782-3795.

[5] Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., Mraz, A., Kashiyama, T., & Sekimoto, Y. (2020). Transfer learning-based road damage detection for multiple countries. arXiv preprint arXiv:2008.13101.

[6] Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., Omata, H., Kashiyama, T., Seto, T., Mraz, A., & Sekimoto, Y. (2021), "RDD2020: An Image Dataset for Smartphone-based Road Damage Detection and Classification", Mendeley Data, V1, doi: 10.17632/5ty2wb6gvg.1

[7] Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., Omata, H., Kashiyama, T., & Sekimoto, Y. (2020, December). Global road damage detection: State-of-the-art solutions. In 2020 IEEE International Conference on Big Data (Big Data) (pp. 5533-5539). IEEE.
[8] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.
[9] Boureau, Y. L., Ponce, J., & LeCun, Y. (2010). A theoretical analysis of feature pooling in visual recognition. In Proceedings of the 27th international conference on machine learning (ICML-10) (pp. 111-118).
[10] Cao, W., Liu, Q., & He, Z. (2020). Review of pavement defect detection methods. Ieee Access, 8, 14531-14544.
[11] Colomina, I., & Molina, P. (2014). Unmanned aerial systems for photogrammetry and remote sensing: A review. ISPRS Journal of photogrammetry and remote sensing, 92, 79-97.
[12] Dhall, A., Dai, D., & Van Gool, L. (2019, June). Real-time 3D traffic cone detection for autonomous driving. In 2019 IEEE Intelligent Vehicles Symposium (IV) (pp. 494-501). IEEE.
[13] Escobar-Wolf, R., Oommen, T., Brooks, C. N., Dobson, R. J., & Ahlborn, T. M. (2018). Unmanned aerial vehicle (UAV)-based assessment of concrete bridge deck delamination using thermal and visible camera sensors: A preliminary analysis. Research in Nondestructive Evaluation, 29(4), 183-198.
[14] Ge, Z., Liu, S., Wang, F., Li, Z., & Sun, J. (2021). Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430.
[15] 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).
[16] Glenn Jocher, Liu Changyu, Adam Hogan, Lijun Yu 于力军, changyu98, Prashant Rai, & Trevor Sullivan. (2020). ultralytics/yolov5: Initial Release (v1.0). Zenodo. https://doi.org/10.5281/zenodo.3908560.
[17] Haris, M., & Glowacz, A. (2021). Road object detection: a comparative study of deep learning-based algorithms. Electronics, 10(16), 1932.
[18] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).
[19] Hsu, T. M., Wang, C. H., & Chen, Y. R. (2018, November). End-to-end deep learning for autonomous longitudinal and lateral control based on vehicle dynamics. In Proceedings of the 2018 International Conference on Artificial Intelligence and Virtual Reality (pp. 111-114).
[20] Jo, Y., & Ryu, S. (2015). Pothole detection system using a black-box camera. Sensors, 15(11), 29316-29331.
[21] Kasperzyk, C., Kim, M. K., & Brilakis, I. (2017). Automated re-prefabrication system for buildings using robotics. Automation in Construction, 83, 184-195.
[22] Katsamenis, I., Doulamis, N., Doulamis, A., Protopapadakis, E., & Voulodimos, A. (2022). Simultaneous Precise Localization and Classification of metal rust defects for robotic-driven maintenance and prefabrication using residual attention U-Net. Automation in Construction, 137, 104182.
[23] Khayatazad, M., De Pue, L., & De Waele, W. (2020). Detection of corrosion on steel structures using automated image processing. Developments in the Built Environment, 3, 100022.
[24] Kim, H., Lee, Y., Yim, B., Park, E., & Kim, H. (2016, October). On-road object detection using deep neural network. In 2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia) (pp. 1-4). IEEE.

[25] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.
[26] Li, G., Xie, H., Yan, W., Chang, Y., & Qu, X. (2020). Detection of road objects with small appearance in images for autonomous driving in various traffic situations using a deep learning based approach. IEEE Access, 8, 211164-211172.
[27] Li, Z., Tian, X., Liu, X., Liu, Y., & Shi, X. (2022). A Two-Stage Industrial Defect Detection Framework Based on Improved-YOLOv5 and Optimized-Inception-ResnetV2 Models. Applied Sciences, 12(2), 834.
[28] Liao, K. W., & Lee, Y. T. (2016). Detection of rust defects on steel bridge coatings via digital image recognition. Automation in Construction, 71, 294-306.
[29] Livens, S., Scheunders, P., Van de Wouwer, G., Van Dyck, D., Smets, H., Winkelmans, J., & Bogaerts, W. (1996). A texture analysis approach to corrosion image classification. Microscopy microanalysis microstructures, 7(2), 143-152.
[30] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).
[31] Loupos, K., Doulamis, A. D., Stentoumis, C., Protopapadakis, E., Makantasis, K., Doulamis, N. D., ... & Singh, P. (2018). Autonomous robotic system for tunnel structural inspection and assessment. International Journal of Intelligent Robotics and Applications, 2(1), pp. 43-66.
[32] Maeda, H., Kashiyama, T., Sekimoto, Y., Seto, T., & Omata, H. (2021). Generative adversarial network for road damage detection. Computer-Aided Civil and Infrastructure Engineering, 36(1), 47-60.
[33] Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., & Omata, H. (2018). Road damage detection and classification using deep neural networks with smartphone images. Computer-Aided Civil and Infrastructure Engineering, 33(12), 1127-1141.
[34] Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., & Omata, H. (2018). Road damage detection and classification using deep neural networks with smartphone images. Computer-Aided Civil and Infrastructure Engineering, 33(12), 1127-1141.
[35] Maintenance guidebook for road pavement 2013 edition. Tech. rep., http://www.road.or.jp/english/publication/index.html.
[36] Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2015). Is object localization for free?-weakly-supervised learning with convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 685-694).
[37] Pandey, A., Puri, M., & Varde, A. (2018, November). Object detection with neural models, deep learning and common sense to aid smart mobility. In 2018 IEEE 30th international conference on tools with artificial intelligence (ICTAI) (pp. 859-863). IEEE.
[38] Passos, Bianka Tallita; Cassaniga, Mateus; Fernandes, Anita M. R. ; Medeiros, Kátya B. ; Comunello, Eros (2020), “Cracks and Potholes in Road Images”, Mendeley Data, V4, doi: 10.17632/t576ydh9v8.4
[39] Pérez, R., Schubert, F., Rasshofer, R., & Biebl, E. (2019, September). Deep learning radar object detection and classification for urban automotive scenarios. In 2019 Kleinheubach Conference (pp. 1-4). IEEE.
[40] Prabhakar, G., Kailath, B., Natarajan, S., & Kumar, R. (2017, July). Obstacle detection and classification using deep learning for tracking in high-speed autonomous driving. In 2017 IEEE region 10 symposium (TENSYMP) (pp. 1-6). IEEE.
[41] Protopapadakis, E., Katsamenis, I., & Doulamis, A. (2020, June). Multi-label deep learning models for continuous monitoring of road infrastructures. In Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments, pp. 1-7.

[42] Protopapadakis, E., Makantasis, K., Kopsiaftis, G., Doulamis, N., & Amditis, A. (2016, February). Crack Identification Via User Feedback, Convolutional Neural Networks and Laser Scanners for Tunnel Infrastructures. In VISIGRAPP (4: VISAPP) (pp. 725-734).
[43] Protopapadakis, E., Voulodimos, A., Doulamis, A., Doulamis, N., & Stathaki, T. (2019). Automatic crack detection for tunnel inspection using deep learning and heuristic image post-processing. Applied intelligence, 49(7), 2793-2806.
[44] 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 (pp. 779-788).
[45] Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 658-666).
[46] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.
[47] Sanil, N., Rakesh, V., Mallapur, R., & Ahmed, M. R. (2020, January). Deep learning techniques for obstacle detection and avoidance in driverless cars. In 2020 International Conference on Artificial Intelligence and Signal Processing (AISP) (pp. 1-4). IEEE.
[48] Scherer, D., Müller, A., & Behnke, S. (2010, September). Evaluation of pooling operations in convolutional architectures for object recognition. In International conference on artificial neural networks (pp. 92-101). Springer, Berlin, Heidelberg.
[49] SEO, S., CHEN, D., KIM, K., KANG, K., Doo, D., Chae, M., & Park, H. K. (2022). Temporary Traffic Control Device Detection for Road Construction Projects using Deep Learning Application. In Construction Research Congress (CRC), Arlington VA.
[50] Silva, L. A., Sanchez San Blas, H., Peral García, D., Sales Mendes, A., & Villarubia González, G. (2020). An architectural multi-agent system for a pavement monitoring system with pothole recognition in UAV images. Sensors, 20(21), 6205.
[51] Silva, L. A., Sanchez San Blas, H., Peral García, D., Sales Mendes, A., & Villarubia González, G. (2020). An architectural multi-agent system for a pavement monitoring system with pothole recognition in UAV images. Sensors, 20(21), 6205.
[52] Tzutalin/LabelImg. Free Software: MIT License. Available online: https://github.com/tzutalin/labelImg (accessed on 05 May 2022).
[53] Varghese, A., Gubbi, J., Ramaswamy, A., & Balamuralidhar, P. (2018). ChangeNet: A deep learning architecture for visual change detection. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops (pp. 0-0).
[54] Wang, L., Sun, P., Xie, M., Ma, S., Li, B., Shi, Y., & Su, Q. (2020). Advanced driver-assistance system (ADAS) for intelligent transportation based on the recognition of traffic cones. Advances in Civil Engineering, 2020.
[55] Xu, X., Zhang, X., & Zhang, T. (2022). Lite-YOLOv5: A Lightweight Deep Learning Detector for On-Board Ship Detection in Large-Scene Sentinel-1 SAR Images. Remote Sensing, 14(4), 1018.
[56] Zhang, L., Yang, F., Zhang, Y. D., & Zhu, Y. J. (2016, September). Road crack detection using deep convolutional neural network. In 2016 IEEE international conference on image processing (ICIP) (pp. 3708-3712). IEEE.