xR4DRAMA

Extended Reality For DisasteR management And Media planning

H2020-952133

D3.8

Outdoors localization algorithms & tools v2

Dissemination level:	Public
Contractual date of delivery:	Month 27, 31.01.2023
Actual date of delivery:	Month 28, 03.02.2023
Work package:	WP3 - Analysis and fusion of multi-modal data
Task:	T3.2 Visual analysis
Type:	Demonstrator
Approval Status:	Final version
Version:	1.0
Number of pages:	33
Filename:	D3.8_xR4Drama_outdoorslocv2_20230203_v1.0.pdf

Abstract

This deliverable describes the updated versions of exterior localisation algorithms & tools.

co-funded by the European Union

History

Version	Date	Reason	Revised by
V0.1	14.12.2022	Table of contents	Theodora Pistola
V0.2	21.12.2022	Input related to PST	Elissavet Batziou
V0.3	20.01.2023	1st version of the deliverable	Theodora Pistola
V0.4	27.01.2023	2nd version of the deliverable for internal review by Up2metric	Theodora Pistola
V1.0	03.02.2023	Final version	Theodora Pistola

Author list

Organization	Name	Contact Information
CERTH	Elissavet Batziou	batziou.el@iti.gr
CERTH	Konstantinos Chatzistavros	konschat@iti.gr
CERTH	Sotiris Diplaris	diplaris@iti.gr
CERTH	Nefeli Georgakopoulou	nefeli.valeria@iti.gr
CERTH	Theodora Pistola	tpistola@iti.gr

Executive Summary

This deliverable elaborates on the modules related to *Task 3.2* (*T3.2– "Visual analysis"*) of the xR4DRAMA project and the appropriate approaches, components, and resources that were adopted to fulfil the respective functionalities that were described in the Description of Actions (DoA) and later on documented by the users throughout the compiled user requirements *(D6.1, D6.2, D6.3)*. Specifically, the document reports on the updated techniques for outdoors localisation algorithms and tools that were deployed during the final phase of the project's lifetime for the implementation of the final xR4DRAMA system (M27). A description of the analysis requirements for visual components is provided, while for each module an overview of the State-of-the-Art (SoA) and a comparison to other approaches is documented. Finally, the evaluation approaches and results are explained and demonstrated.

More specifically, the modules that are described in further detail are the following:

I. **Photorealistic Style Transfer (PST) v2**, which generates new images with the same content and the style of a selected image that can be transferred to make them look like they are in different lighting, time of day or weather. Its goal is to provide enhanced input images to object detection and localisation algorithm and to the newly added river overtopping detection module to get more accurate results.
II. **Building and Object Localisation (BOL) v2**, which is responsible to detect, recognise and localise buildings and the desired objects or elements that might exist in the acquired xR4DRAMA image and video samples.
III. **River Overtopping Detection (ROD)**, which is a new module in the xR4DRAMA visual analysis pipeline. It analyses input from a static camera installed at the Angeli bridge at Vicenza, Italy, estimates the water level and provides information about whether there is an overtopping or not.

It is worth to note, that the performance of the above modules have been extensively evaluated and compared to the first experimental results, proving the correctness of the actions followed during the second period.

Abbreviations and Acronyms

AdaIN	Adaptive Instance Normalization
AR	Augmented Reality
BOL	Building and Object Localisation
DoA	Description of Actions
EmC	Emergency Classification
GAN	Generative Adversarial Network
KB	Knowledge Base
MFA	Multi-layer Feature Aggregation
POI	Point of Interest
PSNR	Peak Signal-to-Noise Ratio
PST	Photorealistic Style Transfer
ROD	River Overtopping Detection
SoA	State-of-the-Art
SR	Scene Recognition
SSIM	Structural Similarity Index Measure
UMFA	U-Net and multi-layer feature aggregation

Table of Contents

List of figures

List of tables

1 INTRODUCTION

In xR4DRAMA, the scope of *Task 3.2 (T3.2 - "Visual analysis")* is to localise the exteriors of buildings, the surrounding environment, as well as objects and other valuable assets needed for media production and situation monitoring.

An overview of the xR4DRAMA visual analysis pipeline is presented in *Figure 1*. Visual analysis receives visual input (images or videos) from the data collection module, the citizen app, the authoring tool and, particularly for the River Overtopping Detection (ROD) module, from static surveillance camera. The extracted subsequences are used to assist the reconstruction of the 3D-model of an unknown area, including the detected objects of interest (T4.4). Moreover, the meaningful semantic information that emerges from the image or video analysis enriches xR4DRAMA's Knowledge Base (KB) (T3.5).

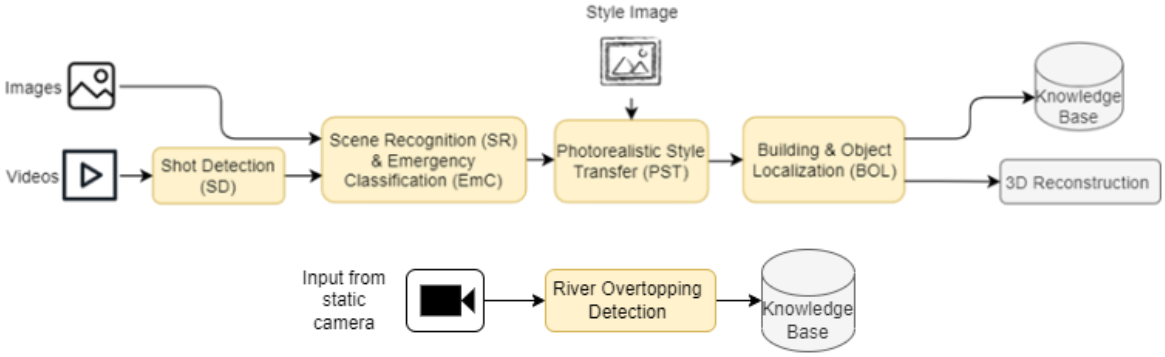


Figure 1: The xR4DRAMA visual analysis pipeline

During the second phase of the xR4DRAMA project (M14-M30), T3.2 contributed to the final xR4DRAMA system (MS4) by deploying and integrating the final versions of Photorealistic Style Transfer (PST), Building and Object localisation (BOL) and River Overtopping Detection (ROD). The updated timeline of T3.2 is depicted in *Figure 2*.

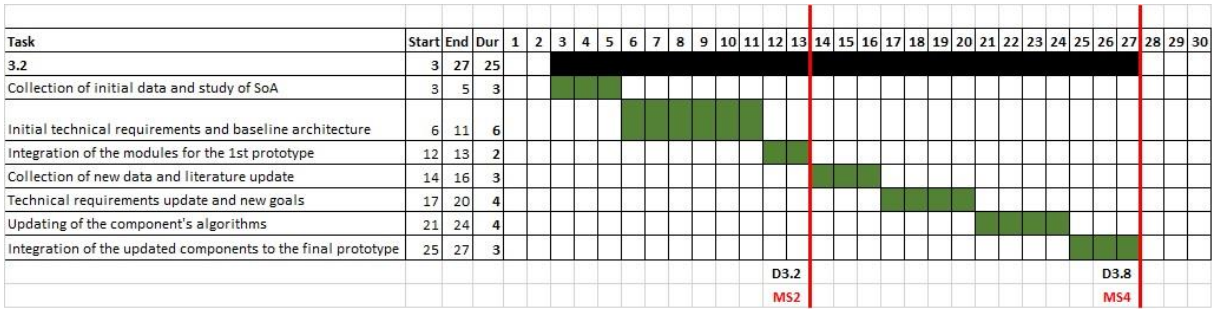


Figure 2: Updated timeline of T3.2

1.1 Objectives

The objectives of T3.2 for the 2nd period of the project (M14-M30) are aligned with the main goals that were described in the DoA, in the IA1.2, and the user requirements reports and summarized as follows:

- Updated study on the literature that exists for photorealistic style transfer, building and object localization and river overtopping detection.

- Development of updated image semantic segmentation algorithms for the localisation of building and their surroundings, objects and other elements of interest.
- Development of a proper river overtopping detection algorithm using input from a static camera installed by the river.
- Enhancement of visual analysis output for the 3D reconstruction module (T4.4) (e.g., extraction of better masks to distinguish between foreground and background pixels within a video frame or image, diminish unwanted information like people and vehicles).
- Integration of the final versions of the visual analysis modules to the xR4DRAMA system.

1.2 Results towards the foreseen objectives of the xR4DRAMA project

xR4DRAMA has fulfilled the foreseen objectives of the project by completing the development of the final versions of photorealistic style transfer and building and object localisation, while adding a river overtopping detection module in the visual analysis pipeline. Specifically, in the 2nd period of the project, the following actions took place:

a) Re-study of the literature related to photorealistic style transfer and building and object localization, in order to update the corresponding modules. Review of literature on water level estimation using visual input to develop the river overtopping detection algorithm.
b) Gathering of additional visual annotated material and datasets to enhance the classification and segmentation models that have been developed.
c) Acceleration of the computational efficiency of the visual analysis modules by redesigning and compressing the corresponding deep learning architectures.
d) Development and deployment of a river overtopping algorithm for the analysis of visual input from a static camera.

1.3 Outline

The outline of this deliverable is as follows. *Sections 2* and *3* respectively contain a brief presentation of the relevant user requirements for the analysis of the visual content and a description of the relevant SoA studies in the scientific fields of computer vision, deep learning and segmentation. Details about the conducted experiments (datasets, settings, results) and the evaluation of the visual analysis modules are then described in *Section 4*, while *Section 5* concludes this deliverable.

2 OUTDOORS LOCALISATION REQUIREMENTS

The xR4DRAMA user requirements have been reported in D6.1 “Pilot use cases and initial user requirements” and D6.2 “Final user requirements”, while they were refined in D6.3 “Evaluation of the 1st prototype and updated user requirements”. *Table 1* presents the nine user requirements reported in D6.2 that are related to the visual analysis components. All, except PUC1-08, are associated with Scene Recognition (SR), Emergency Classification (EmC) and Building and Object Localization (BOL). To fulfil the users’ requirement for the extraction of information related to river embankment's overtopping (PUC1-08), a new module has been developed and integrated in the visual analysis pipeline during the 2nd period of the xR4DRAMA project. The module is the River Overtopping Detection (ROD), which analyses visual input from a static camera installed next to the river and determines if there is any river overtopping or not.

Further details on how the rest user requirements (G-01, G-02, G-03, G04, PUC2-01, PUC2- 02, PUC1-07 and PUC1-09) are covered by the visual analysis modules are reported in D3.2 “Outdoors localization algorithms & tools v1”.

User Requirement (UR)	Category	Name	Description	Priority (1=high, 4=low)
G-01	Accessibility	Transportation	quality and type of road (highway, street, path), distance to railway station and airport, public transport	3
G-02	Geography, Surroundings	Buildings, Monuments	the shape, look and size of buildings, the purpose of buildings	1
G-03	Geography, Surroundings	Landmarks	indication of high voltage lines, windmills and other landmarks	1
G04	Geography, Surroundings	Roads, Railroads	indication of roads, highways, railroads	1
PUC1-07	Flood risk management	Flooded elements	Information on flooded elements (e.g., cars and people inside the river)	1

PUC1- 08	Flood risk management	River embankment's overtopping and/or breaking	Information related river embankments overtopping or breaking	1
PUC1- 09	Flood risk management	Elements at risk	Information on the presence of elements at risk and the degree of emergency	1
PUC2-01	Environmental factors	Noise pollution	identification of possible sources like busy roads or highways, crowds of people, factories, airports, railway stations, railway tracks	1
PUC2- 02	Environmental factors	Light Pollution	identification of possible sources like streetlights, ads etc.	2

Table 1: Relevant user requirements reported in D6.2 for the visual analysis components

3 RELEVANT WORK

In the period M14-M27, a second study of the available techniques and datasets on photorealistic style transfer, building and object localisation and river overtopping detection was performed. Regarding the shot detection, scene recognition and emergency classification we kept the previous versions that were described in D3.2 - "Outdoors localization algorithms & tools v1" (Section 3 for relevant works and Section 4 for modules' description).

3.1 Photorealistic style transfer

In the age of technology and media, numerous pictures are taken every day, but only a small portion of them is taken in ideal lighting circumstances. When there is no other sensor involved, such as several cameras with varying specifications, low brightness, low contrast, a narrow grey range, colour distortion, and severe noise, it is difficult to improve image acquisition in low-lighting conditions. Maximizing the information from the input image on the patterns that are available and extracting the colour information that is concealed behind the low luminance values are the key challenges in low-light image enhancement. Traditional methods of histogram equalization (Fu Y. et al., 2022) and Retinex theory (Li, 2018) have been used to deal with this problem, while neural network approaches either involve Generative Adversarial Networks or Convolutional Neural Networks architectures (Jiang Y. et al., 2021).

Lee et al. (Lee C. et al., 2013) utilise the large grey-level variations and the statistical characteristics of adjacent pixels to alter the local level of brightness. Histogram equalization method creates aesthetically appealing images with limited resources, but with significant distortion because local pixel correlations are not taken into account.

Retinex based methods for image enhancement have more precisely calculated illumination and reflectance using the created weighted variation model (Fu X. et al., 2016). By calculating the highest intensity of each RGB pixel channel, the authors in (Guo X. et al., 2016) produced a coarse illumination map, which they subsequently improved using a structure prior. Despite techniques that modify the distribution of an image's histogram or rely on potentially inaccurate physical models, Retinex theory has been revisited in the work of (Wei . et al., 2018). The authors proposed Retinex-Net, which divides the input images into reflectance and illumination using a neural network, and that adjusts the lighting using an encoder-decoder network in a separate layer. Contrary to using neural networks and Retinex-based techniques to enhance images, these techniques could result in fuzzy artifacts that lower the quality of the resulting image.

There are numerous methods for image enhancement (Fu Y. et al., 2022) that have been introduced in the deep learning era. In (Wang W. et al., 2018) and (Guo X. et al., 2016) the authors propose a GLobal illumination-Aware and Detail-preserving Network (GLADNet). The proposed network's architecture is divided into two phases. For the global illumination prediction, the image is first downsampled to a fixed size and passes through an encoder-decoder network. The second phase is a reconstruction stage, which helps to restore the detail lost during the rescaling process. Moreover, in (Guo C. et al., 2020) the authors propose the Zero-DCE that creates high-order tonal curves from a low-light input image, which are then used for adjusting the input range pixel-by-pixel to produce the improved image. Furthermore, an unsupervised low-light image enhancement method namely Le-GAN is

introduced in (Fu Y. et al., 2022), which includes an illumination-aware attention module and an identity invariant loss. Finally, in (Jiang Y. et al., 2021) the authors propose EnlightenGAN, which is an unsupervised GAN-based model on the low-light image enhancement. EnlightenGAN utilises a one-way GAN and a global–local discriminator structure. Although these methods improve the input's overall brightness, they result in the creation of overexposed regions.

Contrary to the aforementioned approaches, the method that we developed in the context of xR4DRAMA (Batziou, E. et al., 2023) adopts a modified U-Net based architecture with dense blocks and wavelet pooling layers. The low frequency (LL) component is involved in encoding, while the high frequency components (LH, HL, HH) are utilised in the decoding of the input image. As a result, the wavelet pooling transformation used in the proposed approach allows for the preservation of the structure and texture information.

3.2 Building and object localisation

Following the literature review recorded in D3.2, a brief description of the most popular annotated datasets of 2D images related to the problem of building and object localisation follows:

Cityscapes[1] is a large database that focuses on semantic understanding of urban street scenes. It includes a variety of stereo video sequences recorded in street scenes from 50 cities, with high quality pixel-level annotation of 5000 frames, in addition to a set of 20000 weakly annotated frames. It includes semantic and dense pixel annotations of 30 classes, grouped into 8 categories (flat surfaces, humans, vehicles, constructions, objects, nature, sky, and void). In *Figure 3*, some examples of images and their corresponding annotations are shown.

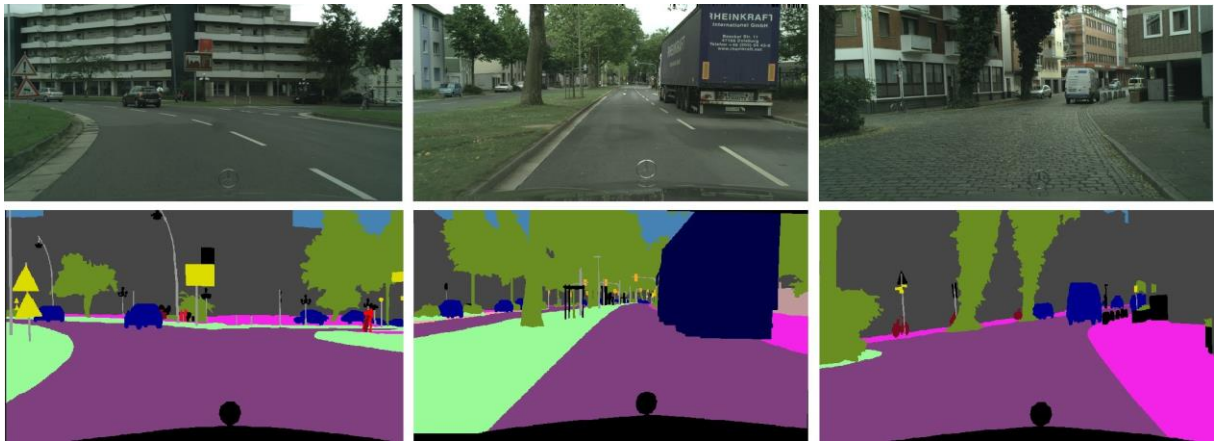


Figure 3: Examples of the CityScapes dataset (1st row training images and 2nd row their corresponding labels)

ADE20k (Zhou B. et al., 2017) contains 20210 photos for training, a validation set of 2000 images, and a testing set of 3000 images. All of the images have extensive object annotations. The parts of many things are also annotated. Additional details on each object, including its properties and whether it is occluded or cropped, are provided. While the part annotations are not exhaustive over the images in the training set, they are over the images in the

[1] https://www.cityscapes-dataset.com/dataset-overview/

validation set. The dataset's annotations continue to expand. Images come from the LabelMe (Torralba A. et al., 2010), SUN datasets (Xiao J. et al., 2010), and Places (Zhou B. et al., 2014) and were selected to cover the 900 scene categories defined in the SUN database. Images were annotated by a single expert worker using the LabelMe interface. This dataset contains both outdoors and indoors images. *Figure 4* displays some examples of the photos and annotations from the ADE20K dataset.

Figure 4: Images in the ADE20K dataset are densely annotated in details with objects and parts. The first row shows the sample images, while the second row shows the annotation of objects and stuff. *Source:* (Zhou B. et al., 2017)

Mapillary Vistas[2] is a diverse street-level imagery dataset with pixel-accurate and instance-specific human annotations for understanding street scenes around the world. It includes 25000 high-resolution images with a variety of weather, season, time of day, camera and viewpoint, as well as annotations of 124 semantic object categories. Examples of the Mapillary Vistas dataset can be seen in *Figure 5*.

[2] https://www.mapillary.com/dataset/vistas

Figure 5: Examples of the Mapillary Vistas dataset (1st row training images and 2nd row their corresponding labels)

PASCAL Visual Object Classes *(VOC)*[3] is a highly popular dataset in computer vision, with annotated images available for 5 tasks (classification, segmentation, detection, action recognition, and person layout). For the segmentation task, there are 21 labelled object classes and pixels are labelled as background if they do not belong to any of these classes. The dataset is divided into two sets, training and validation, with 1464 and 1449 images, respectively, and a private test set for the actual challenge.

Microsoft Common Objects in Context (MS COCO)[4] is a large-scale object detection, segmentation, and captioning dataset. COCO includes images of complex everyday scenes, containing common objects in their natural contexts. This dataset contains photos of 91 object types, with a total of 2.5 million labelled instances in 328K images.

YouTube-Objects[5] contains videos collected from YouTube, which include objects from ten PASCAL VOC classes (airplane, bird, boat, car, cat, cow, dog, horse, motorbike, and train). The original dataset did not contain pixel-wise annotations (as it was originally developed for object detection, with weak annotations). However, Jain and Grauman (Jain, S. D. and Grauman K., 2014) manually annotated a subset of 126 sequences, and then extracted a subset of frames to further generate semantic labels. In total, there are about 10167 annotated 480x360 pixel frames available in this dataset.

CamVid[6] is another scene understanding database (with a focus on road/driving scenes) which was originally captured as five video sequences via camera mounted on the dashboard of a car. A total of 701 frames were provided by sampling from the sequences. These frames were manually annotated into 32 classes.

[3] http://host.robots.ox.ac.uk/pascal/VOC/

[4] https://cocodataset.org/#home

[5] https://data.vision.ee.ethz.ch/cvl/youtube-objects/

[6] http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/

KITTI[7] is one of the most popular datasets for autonomous driving, containing videos of traffic scenarios, recorded with a variety of sensor modalities (including high-resolution RGB, grayscale stereo cameras, and 3D laser scanners). The original dataset does not contain ground truth for semantic segmentation, but researchers have manually annotated parts of the dataset, like (Alvarez, J. et al., 2012) generated ground truth for 323 images from the road detection challenge with 3 classes, namely "road", "vertical", and "sky".

Other datasets for image segmentation purposes include PASCAL Context[8], Semantic Boundaries Dataset (SBD)[9], PASCAL Part[10], Berkeley Segmentation Dataset (BSD)[11], Stanford Background[12], SiftFlow (Liu C. et al., 2009) and SYNTHIA (Ros G. et al., 2016).

3.3 River overtopping detection

The estimation of rivers' water level is a crucial task for detecting flood situations. However, the lack of available data usually makes this task more difficult. Various technologies exist for the estimation and measurement of the rivers' water levels, including ground-based and remote-sensing techniques, like river-gauges, analysis of satellite and airborne images, and images taken by unmanned aerial systems (UAS). In the context of T3.2, we focus on the analysis of visual input (images, videos) from static surveillance cameras installed next to the river.

Relevant studies have suggested the usage of extensive networks of river camera footage to estimate the water levels. Traditional methods exploit edge direction algorithms (Udomsiri S. and Iwahashi M., 2008), (Park S. et al., 2009), (Kwak, J. Y. et al., 2011) (horizontal edge (Yu J. and Hahn H., 2010), (Sakhardande P. et al., 2016), vertical edge detection), pixel difference calculations (Kwak, J. Y. et al., 2011) or optical flow[13] algorithms.

In a recent study (Vandaele R. et al., 2021), the development of a tool for the study of floods using river camera images is presented. This method is based on the application of a transfer learning methodology to deep semantic segmentation networks, in order to repurpose them for the segmentation of water. Once the image is segmented, a proper algorithm directly estimates the water level from some classified landmarks algorithm.

[7] https://www.cvlibs.net/datasets/kitti/

[8] https://www.cs.stanford.edu/~roozbeh/pascal-context/

[9] http://home.bharathh.info/pubs/codes/SBD/download.html

[10] http://roozbehm.info/pascal-parts/pascal-parts.html

[11] https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/

[12] https://www.kaggle.com/datasets/balraj98/stanford-background-dataset

[13] https://www.theimagingsource.com/media/blog/archive/20110908/

4 EXPERIMENTS AND EVALUATION

4.1 Photorealistic style transfer v2

A model should successfully increase the brightness of an image, while recovering the structural information of the image in order to achieve photorealism. The suggested U-Net-based network is paired with wavelet transforms and Adaptive Instance Normalization (AdaIN) to overcome this problem, as it is illustrated in *Figure 6*. More specifically, wavelet pooling and unpooling are used to recover images, while also protecting the content's information for the transfer network. To create the enhanced images, the frameworks input both low-light and normal-light images. The quality of feature transferring is thus improved, and connections are skipped during the transferring process, using dense blocks. The associated characteristics of various levels from the encoding process are added to the enhanced features as part of the image reconstruction process with the goal of creating a natural stylization effect.

The suggested network is comprised of the following three sub-components: encoder, enhancer, and decoder after a U-Net modification. In order to maintain the structural integrity of an image, the encoder and decoder formulate the symmetric structure of a U-Net, as it was first introduced by Ronneberger et al. (Ronneberger O. et al., 2015). Convolutional and pooling layers in the encoder also require a downsampling mechanism, whereas an equivalent upsampling mechanism is present in the decoder. Encoder and decoder modules are connected via a skip connection in the U-Net architecture. The Photorealistic Style Transfer Network named UMFA (Rao D. et al., 2021) adds multi-layer feature aggregation (MFA) and AdaIN blocks (Huang, X. and Belongie, S., 2017) to the already-existing skip connections in a module called enhancer. In order to capture smooth surface and texture information, the enhancer module and the encoder both incorporate a Haar wavelet-based pooling layer in place of the traditional max-pooling layer.

The encoder of the framework is presented in the blue box of *Figure 6* and it has convolutional layers and downsampling dense blocks. Two convolutional layers, a dense block, and a pooling layer based on Haar wavelets are all included in downsampling dense blocks. Three densely connected convolutional layers result in a dense block. The pooling layer's Haar wavelet transforms implement downsampling operations to decrease the feature size in half while maintaining the same dimensions as a max pooling layer. The encoder enables processing of high-resolution images while maintaining multi-scale image information.

The decoder, which is shown in the orange box in *Figure 6*, has four upsampling blocks to create a U-Net structure in addition to the encoder previously mentioned. An upsampling layer, three convolutional layers, and a concatenation operation make up each block. The concatenation operation receives the corresponding encoder feature from the enhancer module.

The enhancer module's structure is shown in *Figure 6* purple box. Multi-scale features are transmitted from the encoder to the decoder using this module. The network can incorporate spatial information from different scales, while preserving the fine details of the input image thanks to feature aggregation. For all pairs of layers in the encoder, features are added from one layer to the next, and then they are transformed using AdaIN to improve the input image. The pooling layer that is based on the Haar wavelet transformation is succinctly explained below.

Haar wavelet pooling has four kernels,$\{LL^T, LH^T, HL^T, HH^T\}$, where the low (L) and high (H) pass filters are: $L^T = \frac{1}{\sqrt{2}}[1 \quad 1]$, $H^T = \frac{1}{\sqrt{2}}[-1 \quad 1]$. Each of the four channels produced by the Haar wavelet-based pooling is represented by the output of each kernel, which is labelled as LL, LH, HL, and HH, respectively. The high-pass filters (LH, HL, and HH) capture edge-like information of various directions, but the low-pass filter (LL) contains smooth surface and texture. Each max-pooling of UMFA method (Rao D. et al, 2021) is replaced with the wavelet pooling layer. The high frequency components (LH, HL, HH) are skipped to the decoder directly while the low frequency kernel (LL) is passed to the next layer of the encoder. The original signal can be exactly recreated by wavelet unpooling using wavelet pooling. Wavelet unpooling performs a component-wise transposed-convolution, followed by a summation, to completely recover the original signal. One of the simplest wavelet transformations, the Haar wavelet, divides the original signal into channels that record various components.

The total loss function follows the approach of (Rao D. et al., 2021), and it is computed as a predefined weighted sum of enhancement loss, content loss and structured similarity loss functions.

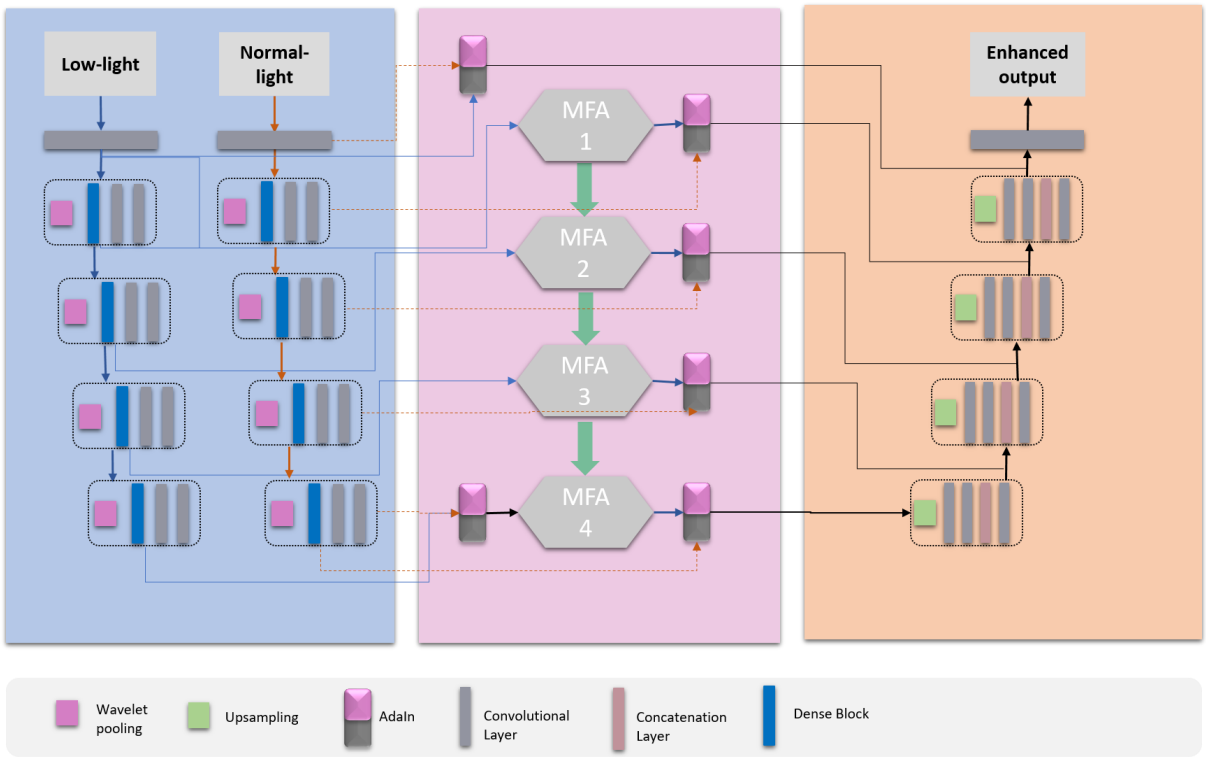


Figure 6: Image enhancement framework based on U-Net architecture and wavepooling layers

4.1.1 Dataset description

Two benchmark datasets namely LOL (Wei . et al., 2018) and SYN (Wei W. et al., 2018) respectively were used to compare the performance of the enhanced and low-light images effectively. LOL dataset has 500 image pairs, where 485 pairs of them are selected randomly for training and 15 pairs for testing. SYN dataset has 1,000 image pairs, and where 950 pairs of them are randomly selected for training and 50 pairs for testing.

4.1.2 Settings

The model was fed with images in both normal and low light to start the training process. For both datasets, the weights for total loss were set as $\alpha = 0.5$, $\beta = \gamma = (1 - \alpha) * 0.5$ following the default values of (Rao D. et al., 2021). Similarly, the number of epochs is set to 150 for the LOL dataset and 100 for the SYN dataset. The batch size is set 4 for both datasets. All input images are 256×256 pixels. Adam optimizer is used for training and the optimal learning rate is 0.0001.

4.1.3 Results

In this section, the qualitative and quantitative validation of the resulting results in comparison to other relevant works is presented. The presented approach denoted in *Table 2*, *Figure 7* and *Figure 8* as "Ours".

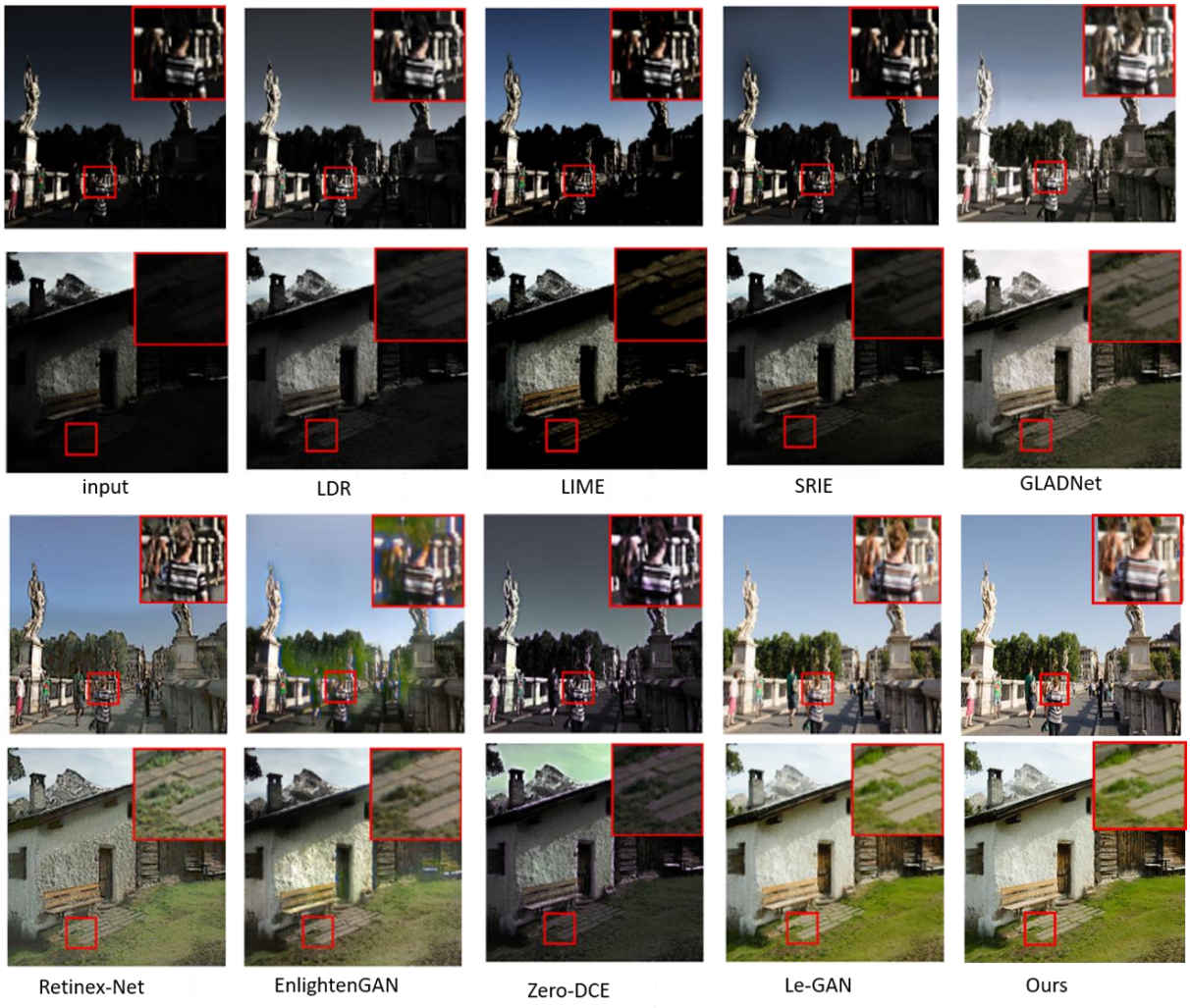


Figure 7: Qualitative comparison of the presented framework with SoA on SYN dataset

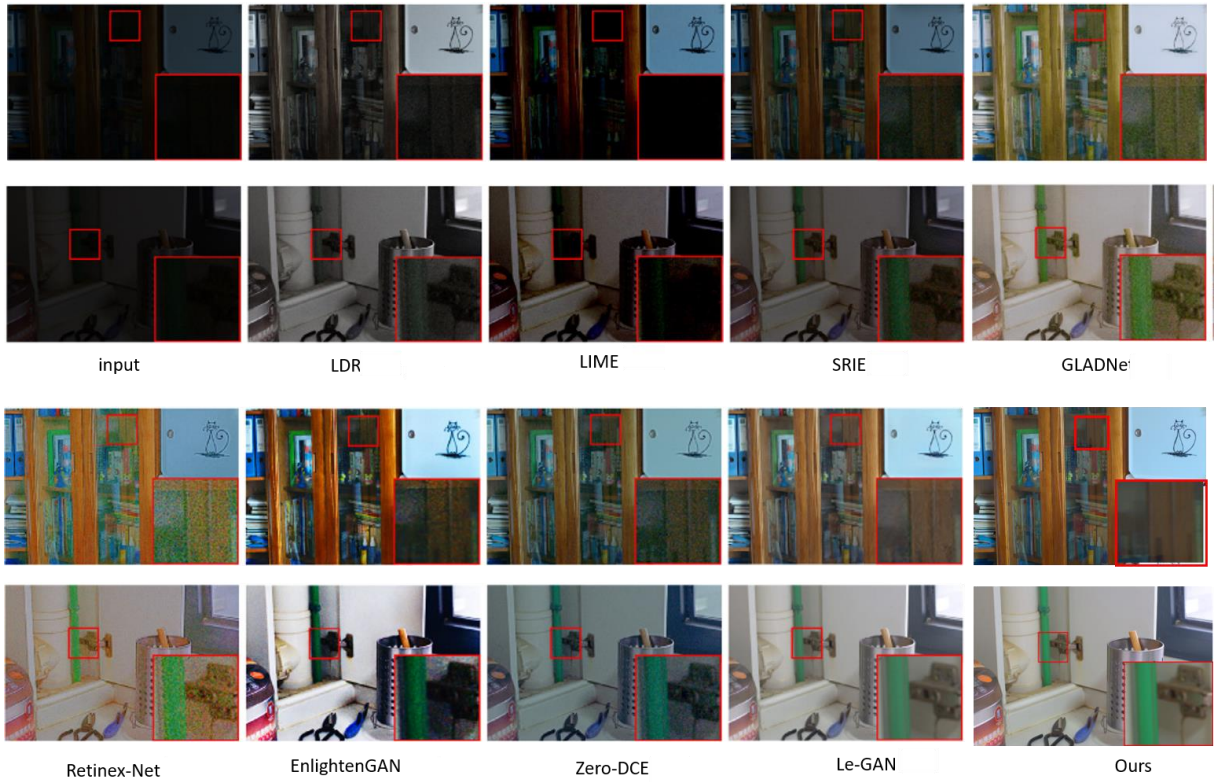


Figure 8: Qualitative comparison of the proposed framework with SoA on LOL dataset

Methods	LOL		SYN		Times	
	PSNR	SSIM	PSNR	SSIM	training	testing
LIME (Guo X. et al., 2016)	15.484	0.634	14.318	0.554	-	-
LDR (Lee C. et al., 2013)	15.484	0.634	13.187	0.591	-	-
SRIE (Fu X. et al., 2016)	17.440	0.649	14.478	0.639	-	-
GLADNet (Wang W. et al., 2018)	20.314	0.739	16.761	0.797	-	-
Retinex-Net (Wei W. et al., 2018)	17.780	0.425	16.286	0.779	-	-
EnlightenGAN (Jiang Y. et al., 2021)	18.850	0.736	16.073	0.827	-	-
Zero-DCE (Guo C. et al., 2020)	10.770	0.426	15.600	0.796	-	-

Le-GAN (YFu Y. et al., 2022)	22.449	0.886	24.014	0.899	25.6h	8.43ms
Ours	21.266	0.784	19.212	0.716	10.7h	5.27ms

Table 2: Quantitative results on LOL and SYN datasets

The qualitative comparison of the presented framework with SoA methods is illustrated in *Figure 7* and *Figure 8*. The hidden information in the low-light input image on the left is observed to be somewhat recovered by the GLADNet (Wang W. et al., 2018), Retinex-Net (Wei, 2018), EnlightenGAN (Jiang, 2021), and Zero-DCE (Guo C. et al., 2020). Although the brightness of the image is increased by these techniques, the colour saturation of the final product is less than that of the suggested way. Moreover, they suffer from noise and colour bias. On the LOL dataset, in particular, it is seen that several bright parts of the results produced by the compared algorithms are over-exposed in the enhanced outputs. In contrast, the suggested method produces normal-light images that are remarkably realistic and performs well across all datasets with almost no artifacts.

Moreover, a quantitative comparison is presented in *Table 2*, where our approach is compared with the aforementioned methods for LOL and SYN datasets respectively. The Peak Signal-to-Noise Ratio (PSNR) scores correspond to the average value of the complete test set of enhanced images. PSNR values demonstrate that, compared to images created using all other methods, those created using the suggested methodology and the Le-GAN method achieve closer approximation to the original normal-light images. The training and testing times show that the suggested approach is substantially more effective than Le-GAN and does not require the additional computational resources mentioned in *Table 2.* More specifically, the training time of Le-GAN method is 25.6 h, 2.5 times higher the training time of the presented approach and they use 3 NVIDIA 3090ti GPUs. The training time of our method is 10.7h on the NVIDIA GeForce RTX 2060 SUPER. Moreover, Le-GANs execution time on a 600 × 400 image is about 8.43ms, 1.5 times up to the execution time of the proposed approach. In Le-GAN, the method requires 3*24GB (standard Memory of 3 NVIDIA 3090ti) contrary to 8GB of ours. In terms of SSIM value, the presented framework's outputs on the LOL dataset are on top with two higher scores, and on the SYN dataset, the results are comparable to those of other SoA techniques.

4.2 Building and object localisation v2

The Building and Object Localisation (BOL) module consists of two different deep learning models:

i) An image semantic segmentation model that mainly focuses on the localisation of buildings and urban elements.

ii) An image instance segmentation model that we use to localise and count the number of people, animals and vehicles, in case the EmC module detects flood in the analysed image or video.

The information extracted from the above models is sent to the xR4DRAMA's Knowledge Base (KB) for further use. The masks generated by the semantic segmentation model are exploited to pre-process the video frames before the 3D reconstruction process (T4.4).

Throughout the xR4DRAMA project, we experimented with different image semantic and instance segmentation models for the localisation of buildings and their surroundings, as well as objects of interest.

Concerning the localisation of buildings and their surroundings, initially, we had trained the DeepLabV3+ architecture on the CityScapes dataset, supporting 19 labels, as reported in D3.2. Afterwards, we deployed and tested in the context of xR4DRAMA the semantic segmentation model of the PixelLib[14] framework, which is also based on the DeepLabV3+ architecture and was trained on the ADE20K dataset. As a final modification of this module, wanting to focus on the labels that really cover the xR4DRAMA's needs, we created a mixed dataset using images both from CityScapes and the Mapillary Vistas datasets keeping only selected labels. Regarding the model for the detection of people, animals and vehicles, we use the instance semantic segmentation model of the PixelLib framework. More information is provided in the following subsections.

4.2.1 **Dataset description**

In this subsection, we provide information about the datasets that we used throughout the 2nd period of the visual analysis modules' development.

For the image semantic segmentation model, in the 1st period of XR4DRAMA we used the CityScapes dataset that supported 19 classes (presented in D3.2). At the beginning of the 2nd period of the project, we utilised the PixelLib image segmentation model that was pre-trained on the ADE20K dataset and supported 150 classes. For the final BOL version we decided to focus only on selected classes that cover the project's needs and we created a dataset that contains images both from the CityScapes and the Mapillary Vistas datasets. Specifically, we used all the images of the CityScapes dataset and a part of the Mapillary Vistas[15]. We merged the common labels and we finally kept the 25 labels shown in *Table 3.*

You can find a brief description of the aforementioned datasets in Subsection 3.2 of this document.

Road	Sidewalk	Parking	Rail track	Building
Wall	Fence	Bridge	Tunnel	Traffic light
Traffic sign	Vegetation	Sky	Terrain	Person
Rider	Car	Truck	Bus	Caravan
Train	Motorcycle	Bicycle	Bench	Background

Table 3: The 25 classes supported by the final xR4DRAMA's semantic segmentation model (BOL)

[14] https://pixellib.readthedocs.io/en/latest/index.html

[15] The Mapillary Vistas dataset was used in the context of a synergy with CALLISTO. Mapillary Vistas dataset is included in the Callisto Dataset Collection.

	Train Images	Validation Images
CityScapes dataset	2975	500
Mapillary Vistas dataset	3000	500
xR4DRAMA's mixed dataset	5975	1000

Table 4: Number of images that consist the xR4DRAMA's mixed dataset for semantic segmentation

The image instance segmentation model of the BOL module was pre-trained on the MS COCO dataset, which support 80 classes in total. For the xR4DRAMA's needs, we only keep 12 classes that are shown in *Table 5* and are separated in 3 categories: i) people, ii) vehicles and iii) animals.

Person	**Vehicles**	Car	**Animals**	Dog
		Motorcycle		Cat
		Bicycle		Horse
		Bus		Cow
		Truck		Sheep
		Train		

Table 5: The 12 classes supported by the final xR4DRAMA's instance segmentation model (BOL) are divided in 3 major categories

4.2.2 **Settings**

The PixelLib image semantic segmentation model was based on a DeepLabV3+ architecture with Xception (Chollet F., 2017) as a backbone and pre-trained on the ADE20K dataset. We used it in a Tensorflow framework.

The final xR4DRAMA image semantic segmentation model has also a DeepLabV3+ architecture[16] with a pre-trained on ImageNet[17] ResNet-50 as a backbone. It was trained on the mixed dataset that we described in the previous subsection. The training of the model was done in Keras[18] and Tensorflow 2.4[19] using an NVIDIA GeForce RTX 3090 GPU. Concerning the training parameters, we used an Adam optimiser, a learning rate of 0.001, batch size was set at 4 and training took 150 epochs.

The PixelLib image instance segmentation model has a Mask-RCNN architecture and was pre-trained on the MS COCO dataset. We used it in a Tensorflow framework, setting the target classes as shown in *Table 5*.

[16] https://keras.io/examples/vision/deeplabv3_plus/

[17] https://www.image-net.org/

[18] https://keras.io/

[19] https://www.tensorflow.org/

The development of the xR4DRAMA BOL module was made in Python 3.7-3.8. In the case of video analysis, image semantic segmentation is applied for every 6 frames, to reduce the analysis time. The instance segmentation model is used only when a "flood" has been detected by the Emergency Classification (EmC) module.

4.2.3 **Results**

Bellow we can see some results of the BOL module. In *Figure 9*, we can see the extracted masks from the image segmentation model and the processed frame based on these masks, where unwanted information was removed.

Figure 9: The analysed frame, the extracted masks and the final masked frame are shown in this figure. The labels detected are: *'building', 'sky', 'vegetation', 'wall', 'road', rail track, 'sidewalk', terrain'*

In *Figure 10*, we can see the results of BOL from the analysis of an image where "flood" is detected by the EmC module. In *Figure 11*, we can see the corresponding output JSON file and the information displayed in the xR4DRAMA AR app. The extracted information (e.g. flood detection, number of cars in danger) of the visual analysis are used to create a Point of Interest (POI), whenever location is known for the analysed image/video.

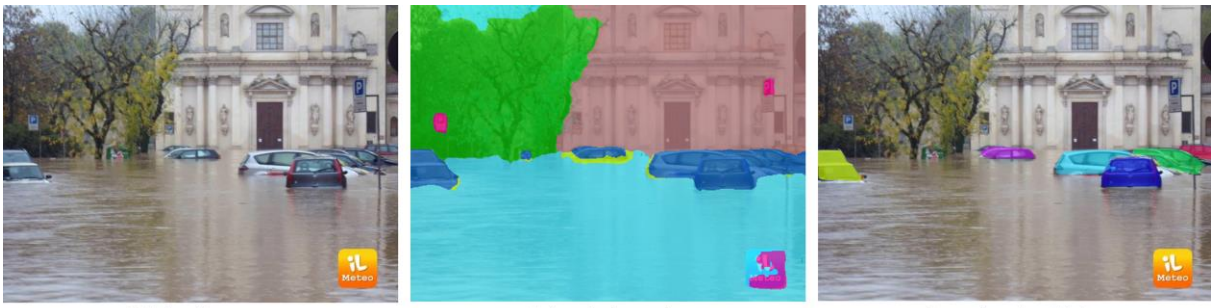
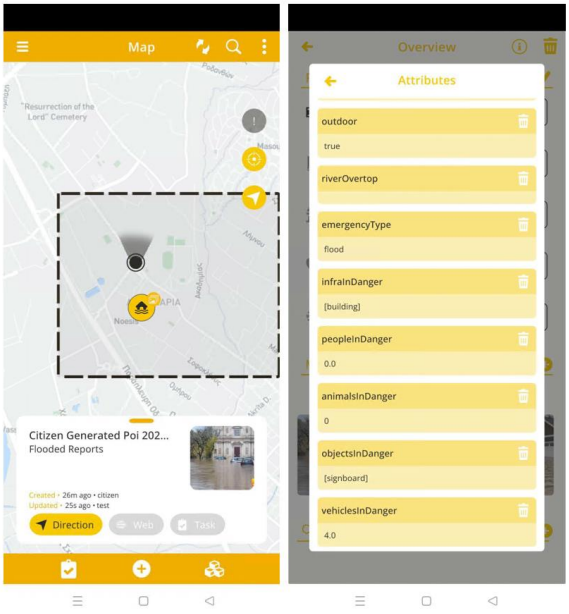


Figure 10: In this figure, we can see the original image that was analysed, the extracted mask and the localised vehicles in case of "flood" detected

```
"objectsFound": [{
        "type": "water",
        "probability": 0.4333648681640625
    }, {
        "type": "building",
        "probability": 0.2800394694010417
    }, {
        "type": "tree",
        "probability": 0.19872029622395837
    }
],
"peopleInDanger": 0,
"vehiclesInDanger": 4,
"riverOvertop": "",
"infraInDanger": ["building"],
"animalsInDanger": 0,
"objectsInDanger": ["signboard"],
"category": "Disaster Management",
"subcategory": "Flooded Reports",
"coordinate": [40.567567547268965, 22.996495214743785],
"area": "canal_urban",
"areaProb": 0.31370000000000003,
"outdoor": true,
"emergencyType": "flood",
"emergencyProb": 0.9996998310089111
```


Information for POI creation/update



Projection of visual analysis information for POI in AR App

Figure 11: Visual analysis output JSON (left) and visualisation of the extracted information on the AR app (right) for the image depicted in *Figure 10*

In Figure 12, we can see the results of the initial and the final BOL module for the pre-processing of the video frames sent to the 3D reconstruction module.

Figure 12: The left column shows results of processed images/frames using the 1st version of the BOL module and the right column shows the corresponding results of the 2nd version

4.3 River overtopping detection

River overtopping detection (ROD) is a new addition to the xR4DRAMA's visual analysis pipeline. Its aim is to perform visual analysis on videos from static surveillance cameras installed by the river to monitor the water level and create alerts when the predefined thresholds are exceeded.

4.3.1 Dataset description

The data that we used for the development and evaluation of the ROD module are video streams from the static surveillance camera installed at the Angeli bridge of the Bacchiglione river in Vicenza, Italy. An example input frame can be seen in *Figure 9*.

4.3.2 Settings

Though the developed algorithm has been calibrated for a static camera placed in Bacchiglione river (Angeli Bridge) in Vicenza, Italy, it is easily adaptable to other cameras. The ROD module receives live video from the area directly from the IP of the static camera and creates short videos to be processed. *Figure 13* shows an example of a captured frame that depicts Angeli bridge, a part of the Bacchiglione river, and an old rod (marked inside a red box), placed on the bank of the river, that was used for measuring the water level, before the installation of water level sensors.

Figure 13: Captured frame for the static camera installed in Bacchiglione river (Angeli Bridge). The red box indicates the marker. Some water level indications (1-6m) are marked on the rod

An edge detection algorithm is used for the detection of the marker (rod), which is of known length. After the marker detection, the system calculates the distance in pixels between the marker's highest and lowest detected points (that should mark the surface of the water). The calculated distance corresponds to the length (in pixels) of the visible part of the marker, which is then translated in real length (in meters) by using calibration data. The length of the visible portion is then correlated to the water level. There are three different types of alerts that are generated if the water level exceeds some predefined threshold: "Moderate",

"Severe" and "Extreme". The thresholds for the specific camera have been defined by AWAA to 3.0m, 4.6m and 5.4m respectively.

Apart from the water level estimation, in case of “Severe” or “Extreme” alerts, the ROD module detects people and vehicles that are present on the road/bridge to have a better view of the emergency situation (e.g., how many people or vehicles may be in danger). For the localisation of people and vehicles, we deployed the PointRend (Kirillov A. et al., 2020) model (ResNet50 variant) of the PixelLib framework, as it provides a good combination of accuracy and speed. The model is trained on the COCO dataset.

In order to increase the module’s accuracy, especially for dusk and dawn when lighting conditions are challenging, we utilised the new Photorealistic Style Transfer algorithm (see Section 4.3), developed in the context of the xR4DRAMA project, before the edge detection of the rod.

4.3.3 **Results**

In *Figure 14*, the original frame that was analysed by the ROD module, along with the cropped rod, after the application of Photorealistic Style Transfer (PST), and its edges. Based on the location of the lowest pixel the algorithm estimates the water level (in this example is under 1m).

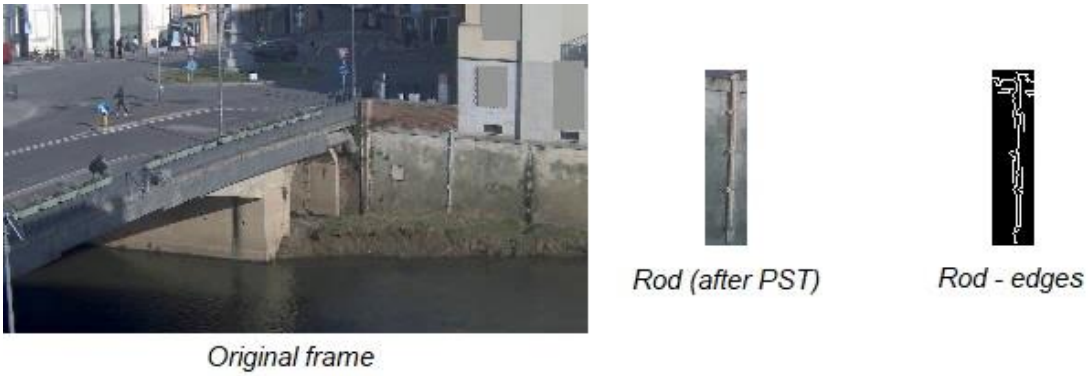


Figure 14: We can see the original frame, the cropped rod after the applying PST and the rod’s edges (water level < 1m)

Figure 15 shows a similar example, where the water level is estimated at 3.7 meters.

Figure 15: We can see the original frame, the cropped rod after the applying PST and the rod’s edges (water level 3.7m)

Figure 16 shows an example of corresponding output in the analysed frame captured by the static camera.

Figure 16: Example of people and vehicles detection and localisation by the ROD module

5 CONCLUSIONS

In this deliverable, we described the research study, the developed components along with the experiments performed, and the final outcomes of T3.2 for the second and last period of the xR4DRAMA project (M14-M30). Taking into consideration the showcased analyses and results, we can easily come to the conclusion that all user requirements presented in D6.1, D6.2 and D6.3 are satisfied, all modules have been deployed and integrated to the xR4DRAMA system and thorough evaluation and testing took place using both benchmark and xR4DRAMA data. As a first step, a thorough search in the related work has been done and the most interesting findings were documented. Then suitable annotated datasets have been researched for our experiments and utilized in order to train efficient and effective models for photorealistic style transfer and building and object localisation. Finally, with the deployment and final integration of the components, the technical requirements were associated to each user requirement, satisfying all these requests.

6 REFERENCES

Alvarez, J. M., Gevers, T., LeCun, Y., & Lopez, A. M. 2012. "Road scene segmentation from a single image" (S. B. Heidelberg, Ed.), Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VII 12, p. 376-389.

Batziou, E., Ioannidis, K., Patras, I., Vrochidis, S., Kompatsiaris, I. 2023. "Low-light image enhancement based on U-Net and Haar", In Proceedings of the 29th International Conference on Multimedia Modeling (MMM 2023). Bergen, Norway.

Chollet, F. 2017. "Xception: Deep learning with depthwise separable convolutions", In Proceedings of the IEEE conference on computer vision and pattern recognition , p. 1251-1258.

Fu, X., Zeng, D., Huang, Y., Zhang, X. P., & Ding, X. 2016. "A weighted variational model for simultaneous reflectance and illumination estimation", In Proceedings of the IEEE conference on computer vision and pattern recognition, p. 2782-2790.

Fu, Y., Hong, Y., Chen, L., You, S. 2022. "LE-GAN: Unsupervised low-light image enhancement network using attention module and identity invariant loss", Knowledge-Based Systems, 240, 108010.

Guo, C., Li, C., Guo, J., Loy, C. C., Hou, J., Kwong, S., Cong, R. 2020. "Zero-reference deep curve estimation for low-light image enhancement", In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, p. 1780-1789.

Guo, X., Li, Y., Ling, H. 2016. "LIME: Low-light image enhancement via illumination map estimation", IEEE Transactions on image processing, 26(2), p. 982-993.

Huang, X. and Belongie, S. 2017. "Arbitrary style transfer in real-time with adaptive instance normalization", In Proceedings of the IEEE international conference on computer vision, p. 1501-1510.

Jain, S. D. and Grauman, K. 2014. "Supervoxel-consistent foreground propagation in video", In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13, p. 656-671. Springer International Publishing.

Jiang, Y., Gong, X., Liu, D., Cheng, Y., Fang, C., Shen, X., Yang J., Zhou P., Wang, Z. 2021. "Enlightengan: Deep light enhancement without paired supervision", *IEEE transactions on image processing*, vol *30*, p. 2340-2349.

Kirillov, A., Wu, Y., He, K., Girshick, R. 2020. "Pointrend: Image segmentation as renderin", In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, p. 9799-9808.

Kwak, J. Y., Ko, B. C., Nam, J. Y. 2011. "Automatic water-level detection using CCD camera and k-mean clustering", J. Kor. Soc. Image Sci. Technol, 17(9), p. 1-8.

Lee, C., Lee, C., Kim, C. S. 2013. "Contrast enhancement based on layered difference representation of 2D histogram", IEEE transactions on image processing, 22(12), p. 5372-5384.

Liu, C., Yuen, J.,Torralba, A. 2009. "Nonparametric scene parsing: Label transfer via dense scene alignment", In 2009 IEEE Conference on Computer Vision and Pattern Recognition, p. 1972-1979. IEEE.

Park, S., Lee, N., Han, Y., Hahn, H. 2009. "The water level detection algorithm using the accumulated histogram with band pass filter", International Journal of Computer and Information Engineering, 3(8), p. 2151-2155.

Rao, D., Wu, X. J., Li, H., Kittler, J., Xu, T. 2021. "UMFA: a photorealistic style transfer method based on U-Net and multi-layer feature aggregation", Journal of Electronic Imaging, 30(5), 053013-053013.

Ronneberger, O., Fischer, P., Brox, T. 2015. "U-net: Convolutional networks for biomedical image segmentation", In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, p. 234-241. Springer International Publishing.

Ros, G., Sellart, L., Materzynska, J., Vazquez, D., & Lopez, A. M. 2016. "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes", In Proceedings of the IEEE conference on computer vision and pattern recognition, p. 3234-3243.

Sakhardande, P., Hanagal, S., Kulkarni, S. 2016. "Design of disaster management system using IoT based interconnected network with smart city monitoring", In 2016 international conference on internet of things and applications (IOTA), p. 185-190. IEEE.

The Imaging Source Europe GmbH Camera Based Water Level Measurement. (n.d.). Retrieved from https: //www.theimagingsource.com/media/blog/archive/20110908/

Torralba, A., Russell, B. C., Yuen, J. 2010. "Labelme: Online image annotation and applications", Proceedings of the IEEE, 98(8), p. 1467-1484.

Udomsiri, S., and Iwahashi, M. 2008. "Design of FIR filter for water level detection", World Academy of Science, Engineering and Technology, vol 48, p. 47-52.

Vandaele, R., Dance, S., Ojha, V. 2021. "Deep learning for the estimation of water-levels using river cameras", Hydrology and Earth System Sciences Discussions 2021, p. 1-29.

Wang, W., Wei, C., Yang, W., Liu, J. 2018. "Gladnet: Low-light enhancement network with global awareness", In 2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018), p. 751-755. IEEE.

Wei, C., Wang, W., Yang, W., Liu, J. 2018. "Deep retinex decomposition for low-light enhancement", *arXiv preprint arXiv:1808.04560*.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., Torralba, A. 2010. "Sun database: Large-scale scene recognition from abbey to zoo", 2010 IEEE computer society conference on computer vision and pattern recognition, p. 3485-3492.

Yu, J. and Hahn, H. 2010. "Remote Detection and Monitoring of a Water Level Using Narrow Band Channel", J. Inf. Sci. Eng., 26(1), p. 71-82.

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A. 2014. "Learning deep features for scene recognition using places database", Advances in neural information processing systems, 27.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A. 2017. "Scene parsing through ade20k dataset", Proceedings of the IEEE conference on computer vision and pattern recognition, p. 633-641.