STREAM
Strategic Development of Flood Management
Project id 10249186

Deliverable 4.3.1 – Automatic video analysis software developed

Table of contents

EXECUTIVE SUMMARY

The work carried out for addressing this project objective started with a deep and detailed analysis of the pilot site's context. All criticalities and general take-aways have been analyzed and described for the subsequent definition of a preliminary solution.

The preliminary solution achieved very good results for what it concerns the night frames (those characterized by higher standardization) but proved to be not sufficient for what it concerned day frames, where variability and complexities connected to sun rays makes it harder to compute the water level in a reliable and accurate way using traditional image processing techniques.

This preliminary attempt allowed the definition of a more advanced solution, which exploits Artificial Intelligence, specifically Deep Neural Networks, for segmenting frames into water/background categories. This segmentation allows for precise and accurate water level computation in both night and day frames. Moreover, this solution has been evaluated in terms of processing speed and cost, in order to understand feasibility of edge implementation of it.

As conclusion of activities within the project scope, the proposed solution, developed and tested on the pilot site, has been implemented into a containerized version. A docker as the one we created, can be easily shared with those in charge for data acquisition and storage, without the need for them of difficult and complex integration of the solution into data management architecture. Specifically, the docker can run easily by taking as input an image file and outputting the computed water level, as a black box solving the problem of water level extraction from images.

The overall results achieved on the pilot site, and the characteristics set for the final solution, makes it adaptable to new sites by means of adaptation in terms of neural network training and site's ortho-rectification parameters, while the main framework is always valid also when we change the application site.

CHAPTER 1. – Context Analysis

Solution Requirements

Automatic monitoring solutions are at the base of developing smart Early Warning (EW) systems for flood hazards, which usually exploit, among relevant input data, the water level of rivers. Commonly adopted technologies for monitoring river flood are pressure transducers, rangefinders, ultrasonic, radar as well as optical sensors. Some of these technologies require frequent calibration, otherwise, the accuracy becomes very low when objects like wooden logs pass underneath, or when the wind causes waves. Moreover, these technologies are prone to measurement errors which could happen especially during dry riverbed and during extreme weather conditions like heavy rainfall, which are those conditions to be controlled more strictly for flood monitoring purposes.

The proposed solution has been created with the following requisites in mind, set together with the entity in charge of disaster management:

a. be fully automatic.
b. be able to detect the water level with an accuracy of ±3 cm.
c. require minimum site-specific customization, except for the initial in-site installation.
d. be able to workday and night.
e. be reliable even during extreme weather conditions.
f. transmit to the central server high-quality data only.
g. be able to work in sites where the gauge is made up of multiple pieces, framed all together by the camera.

Pilot Site Identification

The development of an automatic river flood monitoring solution based on smart video/image processing is strictly connected to the availability of a pilot site, able to provide data for both solution prototyping and testing. To carry out the activities required for sketching a smart solution, a pilot site where both gauge and camera were already installed has been jointly defined with Marche Region Protezione Civile .

The site is particularly interesting also from a flooding point of view, being a site inside the city center of Senigallia, where the Misa River passes (See Figure 1). This river is sadly known for its frequent and dangerous floods.

Figure 1: a snapshot taken by the camera by changing its set-point respect to the one used for flood monitoring purposes.

In the pilot site one RGB camera, with auto-Infra Red for acquisitions during the night, frames the gauge from an optimal perspective, which is almost-front view as can be noted on the two sample frames shown in Figure 2

Figure 2: sample day and night frame acquired by the camera in the pilot site.

Dataset Definition and Organization

The data acquisition system in the pilot site is available since almost a year at the beginning of the project. This eases the development and organization of a pilot dataset, useful for sketching and prototyping the smart automatic image processing solution.

Specifically, we can exploit over sixteen thousand images, collected over the past year with a sampling rate of 1 frame every 30 minutes.

Analysis of Criticalities

Preliminary exploratory analysis of the dataset allowed us to identify some criticalities connected to the creation of the automatic flood monitoring solutions.

To begin with, the camera has a non-frontal view with respect with the gauge. Ortho-rectification is required for mimicking frontal view and obtain images where the water line is perfectly horizontal, which will bring to an optimal and reliable water-level estimation. We solved this issue by defining ex-post the ortho-rectification parameters which have been then embedded in parameters of a roto-translation able to ortho-rectify original frames (see Figure 3).

(a) (b)

Figure 3: ortho-rectification of a sample frame snapped in the pilot site.

Another criticality identified regards the quality of data. Night frames are very stable and similar to each other. Day frames suffer from sunrays inclination with respect to the camera, and can possibly bring to overexposed frames, or snap characterized by heavy light spots that could make image analysis harder. Lastly, weather (especially bad weather conditions) has an impact on image quality since raindrops, snowflakes, or fog, could lower image readability and definition (see Figure 4).

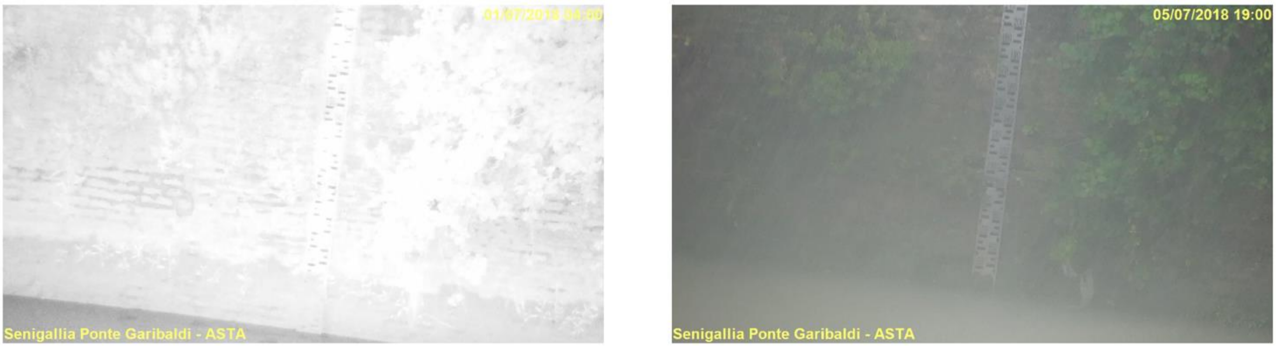

Figure 4: sample overexposed frame and sample bad weather frame.

The general purpose guiding initial solution development has been to keep the analytical solution as simple as possible, in order to be light from a computational point of view and possibly in line with potential edge implementation of the solution as a future step. For this reason, we are going to present the initial sketch solution that mostly ground on traditional image processing techniques. By analyzing weaknesses of this solution, we identified potential improvements that have been included in the final solution, whose performances and characteristics makes it reliable enough, suitable for future edge implementation, and hence a definitive solution.

CHAPTER 2. – Sketch solution 1

In order to compute the water level during the entire day with an accuracy of ±3 cm, image processing should rely on good quality images, but it is easy to have some bad quality frames, as show in the previous section, either due to bad weather or to contingencies connected to the unconstrained and non-standardized context of use. Therefore, we structured the overall solution based on two modules. The first module takes as input a frame snapped by the V-IoT device, and classifies it as either day, night, or bad quality (either overexposed, blurred, or with weather related artifacts), through light and fast computations. The second module has the objective of computing the water level, exploiting as input both the snapped frame and its category computed by the Image quality check module. This is extremely important since day and night frames have completely different characterization, which reflect on different steps required for computing the water level.
The proposed solution is very simple and fast. In the following subsections we are going to analyze performances achieved by this solution.

MODULE 1: Image Category Classifier

The first module involves steps for classifying a frame as either night, day or bad quality, according to the flowchart presented in Figure 5.

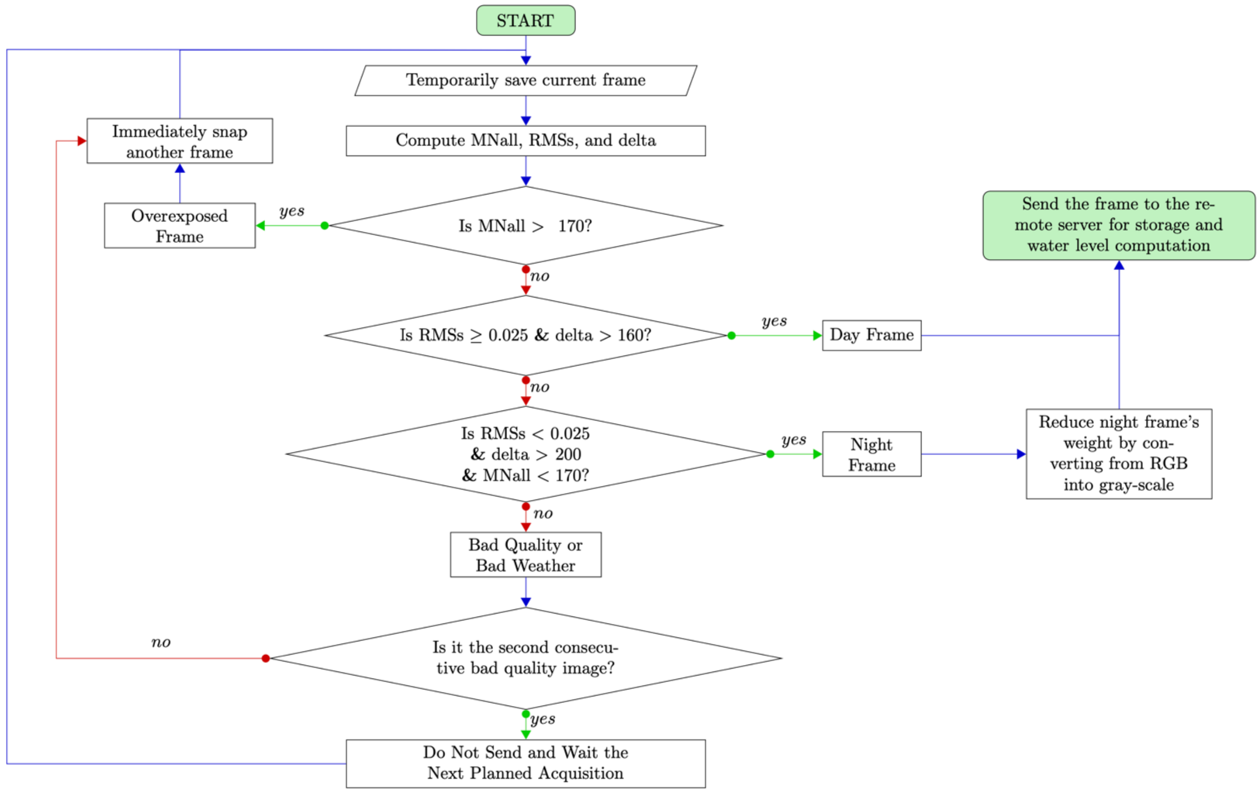

Figure 5:image quality check module flowchart

We analyzed correlation between some metrics extracted from frames snapped, and the image classes, finding some interesting strong relations. The metrics evaluated are:

- Mean of all pixels of RGB color channels (MNall);
- Mean of the saturation channel of the image converted into HSV color model (MNs);
- Root Mean Square of RGB channels (RMSall);
- Root Mean Square of saturation channel (RMSs);
- Maximum inter-pixel difference, computed as the maximum along all the pixel intensities, minus the minimum (delta);
- Variance of the image histogram (VARih);

Specifically, MNall is strictly discerning the overexposed frames from the other classes. MNs and RMSs were strongly different between day frames and the other classes, as could be expected since during the night the camera acquires through auto-IR cut filter and the resulting RGB image is like a gray-scale one. The night frames have very low saturation thus resulting very similar to frames snapped during bad weather or in case of bad quality images, where the saturation gets low too.

Another interesting connection has been found for the delta metric, suitable for discerning between bad quality, against good quality both day or night frames.
In this preliminary attempt, frames categorized as bad quality have been discarded from subsequent analysis.

MODULE 2: Gauge Detection And Water Level Computation

Once an image is categorized as either day or night, the following step is the same for both categories: the ortho-rectification of the image. Through this step, we pass from an image where the gauge is distorted, due to the relative perspective of the camera and the gauge, to a rectified image where the gauge seems frontally framed. Specifically, this procedure has to be set for each site, and starts with the identification of the four rectangle's vertices and the rectified rectangle associated to a perfect frontal view. Subsequent steps are category specific. Specifically, after the rectification of a day frame these steps are performed:

- D1 top hat filtering, using as structuring element a disk of 15 pixels radius;
- D2 adjust image intensities, saturating top and bottom 1% of all pixel values;
- D3 binarize the image using a fixed threshold (45);
- D4 eliminate from the binary image those connected regions having area lower than 50 pixels, to reduce noise;
- D5 fill the holes;
- D6 perform morphological closing using as structuring element a line of 30 pixels;
- D7 compute the percentage of white pixels for each row over the columns;
- D8 compute the adaptive threshold as mean minus one standard deviation of the row percentages;
- D9 start from the bottom and find the first line where the row percentage exceeds the threshold, which is the water level;
- D10 draw a red line corresponding to the computed water level.

On the other hand, after the rectification of a night frame these steps are performed:

- N1 detection of the gauge and cutting the image;
- N2 median filtering the retained portion;
- N3 extraction from it of 5 thresholds of intensity;
- N4 sharpening by a factor of 1.4;
- N5 clustering based on multiple thresholds computed before;
- N6 clusters' edges extraction, using Canny algorithm;
- N7 holes filling;
- N8 morphological closing using as structuring element a rectangle of 4 by 15 pixels;
- N9 eliminate from the binary image those connected regions having area lower than 50 pixels, to reduce noise;
- N10 holes filling;
- N11 compute the sum of black pixels for each row;

N12 assign to each row the value of 0 if the number of black pixels is lower than 70% of row pixels, 1 otherwise;

N13 find the water level which is the first non-zero line.

Results

Night frames proved to be very stable and similar to each other, something that allowed to achieve good results in terms of extracted water level.

Day frames on the contrary, are characterized by strong variance in terms of chromatic characterization, light rays perspective, and many other variables. This turned out to be critical for the reliability pf water level computed.

Results regarding the first module are summarized in the following table:

	Actual	**Detected**
Night	6628	6652
Day	6348	6369
Bad Quality	532	487

Concerning the second module we decided to divide between “correct water level”, if the computed line is no more than 3 cm apart from the actual water line, “small errors” if the computed line is 3 cm to 10 cm apart from the actual water line, and “heavy errors” otherwise. Results are summarized in the following table:

	Correct Water Level	**Small Errors**	**Heavy Errors**
Night	6545	35	48
Day	5300	465	483

CHAPTER 3. – Final Solution

Starting from the analysis of strengths and weaknesses of the preliminary solution just presented, we decided to seek improvements creating a framework for solving the problem of real-time water level monitoring through on-field cameras. The framework aims at both accuracy in the water-level computation and at suitability for being implemented on embedded and low-cost devices, for its future scalability for the development of reliable widespread sensing networks available to public control bodies such as Civil Protection to cope with floods.

The proposed final solution is based on Semantic Segmentation with a Convolutional Neural Network and on image coordinates registration with respect to a reference frame. The network classifies pixels in the two categories background/water and produces an output binary label which is later used to perform level estimation. The rigid transformation was estimated with state-of-art feature matching-based image registration method, a well known, effective and widely used approach in computer vision for image registration in remote sensing applications also. These techniques extract significant image points and their associated feature vectors according to the chosen type of descriptors. Input and reference image features then are matched according to a proper similarity metric thus providing several feature points pairs which are used to estimate geometric transform.

Feature matching

The best trade-off between align errors and computational effort is reached by ORB descriptors which thus represent the best alternative for a scalable framework. As a consequence, ORB were chosen for image rectification in the level estimation experiments.

Semantic Segmentation

In order to check the consistency of our framework, at first three of the most popular Semantic Segmentation DNN in the context of image water level estimation were tested, namely SegNet, FCN8, and Deeplabv3plus:

- SegNet resumes the typical paradigm of Semantic Segmentation CNN with a encoder/decoder architecture. The encoder consists of several blocks each made up of a couple of 3×3 convolutional layers and a final MaxPooling which reduces feature size. The decoder is symmetric with an Upsampling operation at the end of each block which eventually lead to a final feature with the same size as input image.
- FCN8 has an encoder with same structure as SegNet, however the encoder is not symmetric. Furthermore, intermediate encoders' features are re-used by adding them to the outputs of decoding blocks, which are formed by successions of 7×7, pointwise and transposed convolutions which perform upsampling.

- Deeplabv3plus is a high quality Segmentation CNN, it was developed as a solution to several problems in segmentation including object scale variation and accurate localization of boundaries. Its major novelty is the atrous spatial pyramidal pooling (ASPP) module, which performs features' dilated convolutions with different rates and then concatenates the outputs.

Nevertheless, the aforementioned architectures are too computationally and memory expensive to be scaled on embedded devices in practice.

For this reason, in addition to these networks a group of three lightweight architectures was tested, which balance good classification accuracy with high inference speed and low memory occupation:

- ERFNet includes in both encoder and decoder the so called "bottlenecks" which are residual blocks with 2 pairs of horizontal and vertical convolutions in the main path. This strategy allows to capture the same spatial information and to reduce computational cost.
- ENet similarly to ERFNet stacks several bottlenecks in encoder and decoder. ENet bottlenecks have many types of convolutions as standard square filters, dilated, horizontal and vertical. The main difference with respect to ERFNet however is that ENet reduces the number of filters within each bottleneck, leading to thus to a extremely efficient model.
- FastSCNN was specifically optimized to reduce inference time on embedded platforms. It is formed by bottlenecks with depth-wise convolutions between 2 pointwise layers which respectively increase the number of filters and bring it back to the same number of input.

Results

Results of image rectification with feature matching in terms of % of wrong aligned images is summarized in the following table:

	SIFT	**SURF**	**FAST**	**ORB**
day	0.1	1.8	16.4	0.6
night	0.6	0.2	0.6	0.2
bad-weather	2.5	15.6	48.6	14.7

The best trade-off between align errors and computational effort is reached by ORB descriptors which thus represent the best alternative for a scalable framework. As a consequence, ORB were chosen for image rectification in the level estimation experiments.

Segmentation quality was measured by evaluating average pixel classification accuracy and the classical mean Intersection-Over-Union (mIoU) averaged among background/water classes which are reported in the following table:

	Test accuracy			**mIoU**			**Storage cost**	**GFLOPs**
	Day	*Night*	*Bad weather*	*Day*	*Night*	*Bad weather*		
SegNet	99.22	99.61	95.26	97.71	98.88	86.93	22.707	62.83
DeepLabv3+	99.69	99.78	99.78	99.07	99.37	99.28	190.523	26.42
FCN8	99.03	99.56	98.94	97.18	98.74	96.58	1026.068	173.45
FastSCNN	99.60	99.75	99.75	98.82	99.30	99.17	15.174	2.43
ERFNet	99.65	99.75	99.66	98.97	99.28	98.90	24.675	20.91
ENet	99.55	99.72	99.71	98.66	99.19	99.05	5.760	3.13

Results achieved suggest that the considered low resource requiring networks could be suitable for the proposed water level estimation framework. Among these FastSCNN and ENet are the most efficient in terms of storage cost and GFLOPs.

Combining ORB feature matching with ENet, we reach a lightweight solution suitable for embedded and constrained devices. Results achieved by this framework prove to be reliable enough for making this configuration, the one adopted for pilot-site implementation, as will be presented in the next section.

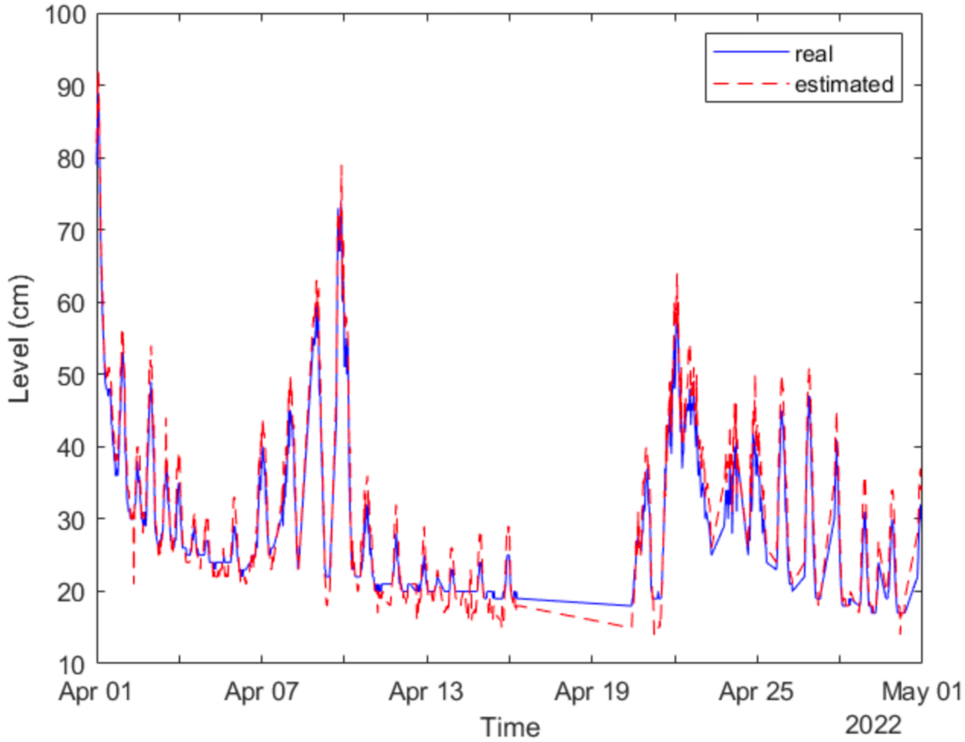

Figure 6: estimated and real water level for some testing images using ENet and ORB features.

CHAPTER 4. – Docker Implementation

Once defined the solution able to meet all requirements set by us, in terms of both water-level computation accuracy and potential to be implemented on embedded devices as a future activity, a pilot implementation has been developed for the pilot site.

Among several ways that could bring us to the concrete implementation of the solution, the docker-based has been selected and implemented. By means of dockers, scalability and ability to work in parallel analyzing frames related to multiple acquisition sites is guaranteed. Moreover, docker implementation is easy to be maintained and shared.

For these reasons, a docker implementing ORB feature matching and using the trained ENet for semantic segmentation of frames snapped has been created and shared with project partners. The docker takes as input an image file name and outputs the computed water level according to ORB feature matching and ENet presented in the previous section.

Docker’s correct functioning has been tested by the institution in charge for image storage.

CHAPTER 5. Output of the Research

Based on the work carried out for this project portion, two research papers, one published and one under review, have been created by the Research team lead by Paola Pierleoni.

- [2021] Sabbatini, L., Palma, L., Belli, A., Sini, F., &, Pierleoni, P. **A Computer Vision System for Staff Gauge in River Flood Monitoring**. *Inventions*, 6(4), 1–16. https://doi.org/10.3390/inventions6040079

- [under review 2023] Pierleoni, P., Falaschetti, L., Manoni, L., Sabbatini, L., Turchetti, C., and Palma, L. **A Semantic Segmentation Based Framework for Water Level Monitoring.** Engineering Applications of Artificial Intelligence (Elsevier).

CHAPTER 6. Discussion and Conclusion

The final solution developed for solving the considered problem proved to be reliable, and suitable for future edge implementation into Visual-IoT nodes.
The proposed solution is site specific but organized in a way that makes it easy the scalability to multiple different sites with minor adaptations. Specifically, adaptations do not refer to the overall framework, which can be considered standardized and general, but only to the network that should be retrained on the specific site, and to the ortho-rectification parameters that can be easily defined for every new acquisition site with very little efforts.

With these considerations in mind, we can conclude that this activity bring us to a satisfactory proof of concepts.