

WP2 Contribution to Milestone **M10.1** Deliverable **TAR2.8**

Preliminary version of algorithms for automatic or semi-automatic extraction of information from digital twin using machine learning technique

Date: March 14th, 2025

Partners: INAF	
Work-Package: WP2	
Milestone: M10	
Responsible partner: Istituto Nazionale di Astrofisica (INAF)	
Author(s): Francesco Franchina, Mauro Imbrosciano, Eva Sciacca	

Document revisions		
<i>Author(s)</i>	<i>Revision content</i>	<i>Date</i>
Eva Sciacca	First draft of the document from TAR2.4 [19]	10/01/2025
Francesco Franchina	Expanding Section 3,4 and 5	23/01/2025
Mauro Imbrosciano, Eva Sciacca	Updated Section 2, 4 and Conclusion	14/03/2025
Mauro Imbrosciano, Eva Sciacca	Final Version	14/03/2025

ABSTRACT

The report aims at describing the preliminary version of an image semantic segmentation algorithm capable of speeding up the detection of flooded areas and related damages. It describes the initial needs, the study of the state of the art algorithms for such a class of problems and the challenges faced during the training and, being still a work in progress, future outlooks for its development.

TABLE OF CONTENTS

1. INTRODUCTION	6
2. Survey on segmentation methods	7
Semantic segmentation	7
Conclusions of the survey	9
3. Dataset	11
FloodNet 2021	11
RescueNet 2021	13
4. Training methods	13
5. Training outcomes	14
6. Transfer learning	19
7. Deployment and Web Service	22
8. CONCLUSIONS	24

LIST OF ACRONYMS and ABBREVIATIONS

HaMMon: Hazard Mapping and vulnerability Monitoring
SAM: Segment Anything Model

WP: Work Package

FCN: Fully Convolutional Network

SAM: Segment Anything Model

PoCI: Proof of Concept Infrastructure

API: Application Programming Interface

CNN: Convolutional Neural Network

MLP: Multi-Layer Perceptron

UAV: Unmanned Aerial Vehicles

1. INTRODUCTION

The Hazard Mapping and vulnerability Monitoring (HaMMon) project aims to develop tools and methodologies to be used in different industrial contexts for the quantification of the impacts of extreme natural events on the Italian territory. In order, also, to support activities focused on monitoring, forecasting and assessment of environmental risks and protection from potentially critical phenomena.

The reference areas are the identification of factors contributing to the risk, forecasts on the occurrence of extreme events in the medium term and the quantification of impacts after catastrophic events. The activities involve intensive use of scientific visualisation and artificial intelligence technologies, especially for assessing and extracting meaningful information on risk-exposed assets.

The WP2 aims to improve the assessment of the damage after a natural disaster, helping to expedite the processing of claims and reduce the time and resources needed for on-site inspections.

The task T2.1 "Workflow for data acquisition and creation of digital twin" is developing a workflow [1] to produce a high-resolution 3D tiled model with centimetre accuracy of the affected area from 2D images captured along proper Unmanned Aerial Vehicles (UAV) flight paths, combining them to create a sparse and dense point cloud, creating a mesh, and generating a texture.

In task T2.4 "Automatic (or semi-automatic) analysis" we are focussing on the development of algorithms for automatic or semi-automatic extraction of information from the digital twin using machine and deep learning techniques. The algorithm will be used to identify and classify different damages within a 3D model. The task involves preparing the data by labelling the buildings, selecting an appropriate machine learning model, training the model on the labelled data, testing the model on a separate set of data, optimising the model performance, and deploying the model to use it within the web application.

This report describes the process of development of an image semantic segmentation algorithm capable of speeding up the detection of flooded areas and related damages. It describes the initial needs, the study of the state of the art algorithms for such a class of problems and the challenges faced during the training and, being still a work in progress, future outlooks for its development.

2. Survey on segmentation methods

The main focus of this R&D activity is to be able to detect patches of aerial images that present some features of interest. Since we are investigating the field of hazardous natural events, we need to identify precisely areas and, specifically, their shapes. For instance, in the case of fire it could be useful to identify dried patches that could easily catch fire, for earthquakes could be useful to analyse the density and the position of the buildings, for the floods could be useful to detect the areas that have been submerged during the episode. Such needs could be addressed easily from the flexibility offered by the segmentation algorithms.

Semantic segmentation

Semantic segmentation is one of the prime research areas in computer vision: the literature is vast and continuously expanding.

The majority of the models implement Fully Convolutional Networks (FCN) architectures inspired by the pioneering paper [3]. This approach leverages the significant benefit of handling inputs of any size and efficiently producing correspondingly sized output maps.

From the perspective of contextual aggregation, segmentation models can be roughly divided into three broad categories: Encoder/Decoder Networks, Pyramid Pooling, Other Schemas.

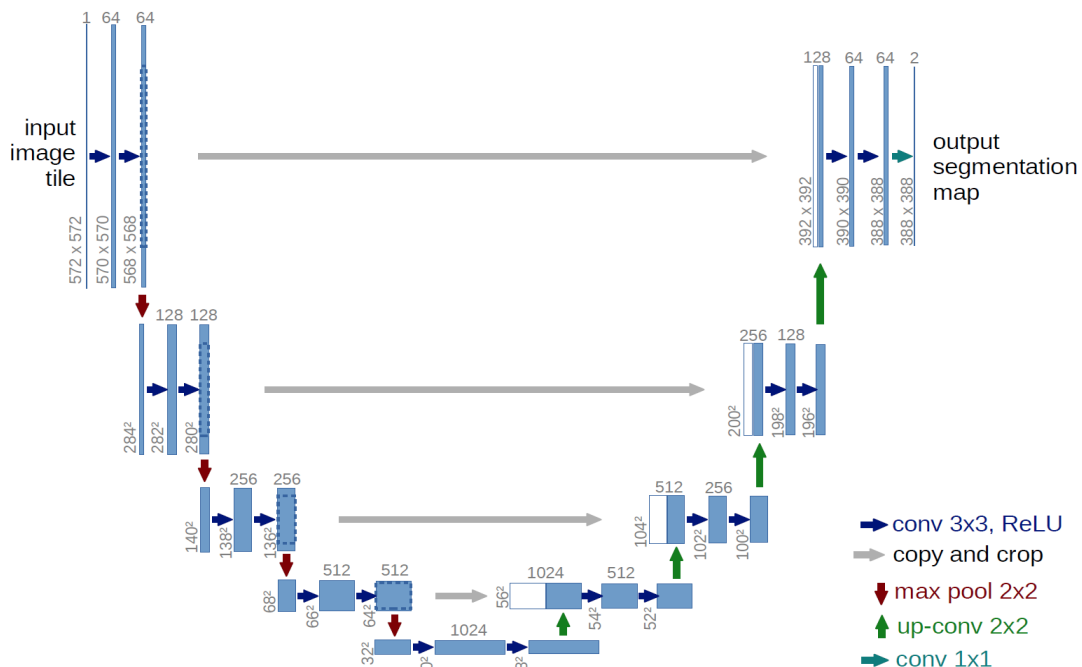


Figure 1: Diagram of a U-Net Architecture, from [5]

Encoder/Decoder Networks

Inspired from the SegNet [11] model, such solutions aim at compressing the information contained in the image into a highly generalised vector that, after being classified, is brought back to the image domain as a collection of masks that describe the different parts of the initial image.

U-Net [5] used symmetric downsampling/upsampling branches, adding skip connections between blocks of the same level to help restore spatial details (a diagram of the model is shown in Fig. 1). It has a very broad usage and became a paradigm in image segmentation. Due to the vast literature and simplicity, U-Net based solutions are very popular and cost-effective, this is one of the reasons that convinced us to investigate such solutions in the first place.

U-nets can be enhanced with Dense Blocks [4] or Harmonic Dense Blocks [12] and recently even the Transformer self-attention mechanism has been integrated into this architecture to improve the feature detection process (see TransUNet [13]).

Pyramid Pooling

PSPNet [14] (Pyramid Scene Parsing Network) marked a big step in the panorama of segmentation since it introduced a new approach. To begin with, the architecture makes use of a pre-trained CNN that builds a feature map of the input image, then PSPNet makes use of a special block in the architecture, the so-called pyramid pooling module. This acts on the feature map in a similar fashion as a pooling layer but does it at the same time on different scales, hence the name pyramid. The network then stacks together all the features along with the upsampled features coming from the pyramids in order to make the inference about the classes of each pixel.

The trick here lies in considering the features with different granularities in order to extract context information.

Other networks based on the principle of combining information from different resolution scales also exist. A notable example is DeepLabv3+ [16], which utilises Atrous Spatial Pyramid Pooling (ASPP). This technique applies atrous (dilated) convolutions with varying dilation rates to effectively capture multi-scale contextual information.

Other schemas

During the course of our research, we explored a number of models that do not align with the categories previously established due to their distinctive features. Let's briefly review the most important of them:

- SegFormer [15] is an effective technique that leverages the advancements in the machine learning field by employing the attention mechanism implemented into the Vision Transformers [8] to make sense of sparse spatial information present in the image. In

particular SegFormer extracts and selects many features at different resolutions of the input image and then passes all the collected features to a lightweight MLP architecture that is able to classify each of its pixels.

- **SAM:** pioneering works has been done by meta's FAIR that, taking advantage of semi-supervised machine learning techniques, developed the Segment Anything Model (SAM) [7], a foundational model for image segmentation capable of building an outstanding dataset of 11 million segmented images. The architecture of SAM is quite complex, and will be briefly discussed in the following section.
- **Attention U-Net:** an enhanced version of U-Net, the Attention U-Net incorporates attention gates (AGs) into the skip connections between the encoder and decoder. Unlike the standard U-Net, which directly transfers features, these AGs filter and refine the feature maps, emphasizing relevant structures while suppressing irrelevant background information.
- **Segmenter:** a segmentation model based on the Vision Transformer (ViT), introduced by Robin Strudel et al. (2021) in [18]. The ViT serves as the backbone for feature extraction, while a Mask Transformer decoder directly predicts segmentation masks, leveraging self-attention mechanisms to model relationships between segments. This approach is particularly effective on complex datasets such as ADE20K and COCO. However, it requires significant computational resources.

Conclusions of the survey

After analysing the former models, we decided to focus on a specific type of U-Net that builds on the experience of other models and outperforms them on most of the aspects, **Tiramisu** [2]. Such a solution would provide a strong baseline since it is still a quite straightforward and interpretable model that can still compete in the landscape of the segmentation solutions.

In parallel we decided to dedicate some time also to the **SAM** model since it is very advanced and can be used to empower or refine simpler methods.

Tiramisu

The model, described in [2], is a fully convolutional neural network [3], therefore adaptable to various input image sizes. It utilises dense blocks [4] for enhanced feature detection and is structured like a U-net [5] to be able to address segmentation tasks. A conceptual diagram of the proposed architecture, taken from the original paper, is shown in Fig. 2.

This model has been successfully used for segmenting radio astronomical images [6].

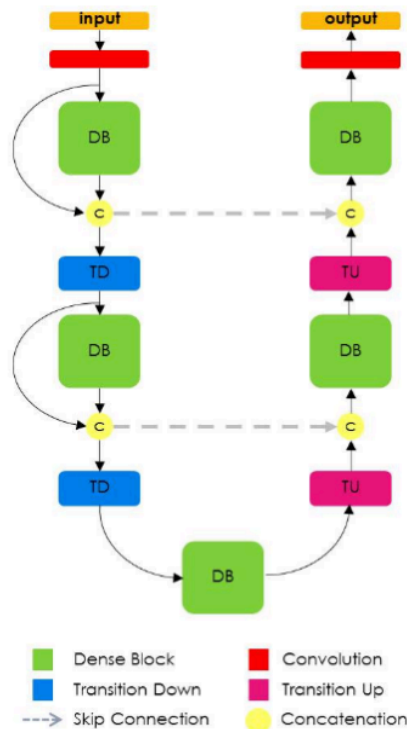


Figure 2: Diagram of Tiramisu architecture for semantic segmentation, taken from the paper [2]

Segment Anything Model

The Segment Anything Model (SAM) [7], developed by Meta's FAIR (Fundamental AI Research), is a complex model inspired by recent developments in Natural Language Processing.

It has several components, with the main building block being a heavy Vision Transformer that computes image encodings [8]. Other components are responsible for encoding prompts of various natures like points, boxes, segmentation masks and even text (see Fig. 3). The model also features a lightweight decoding component, which is crucial for its efficiency and real-time performance. This decoder uses the encoded image features and prompts to generate the final segmentation mask.

SAM is capable of addressing "few or zero-shot tasks", assignments where the training dataset is very limited.

We explore two possibilities:

1. Using SAM to improve the resolution of the output from a lighter and rawer model specialised in coarse tasks, using the latter as a segmentation prompt.

2. Fine-tuning the lightweight decoding branch.

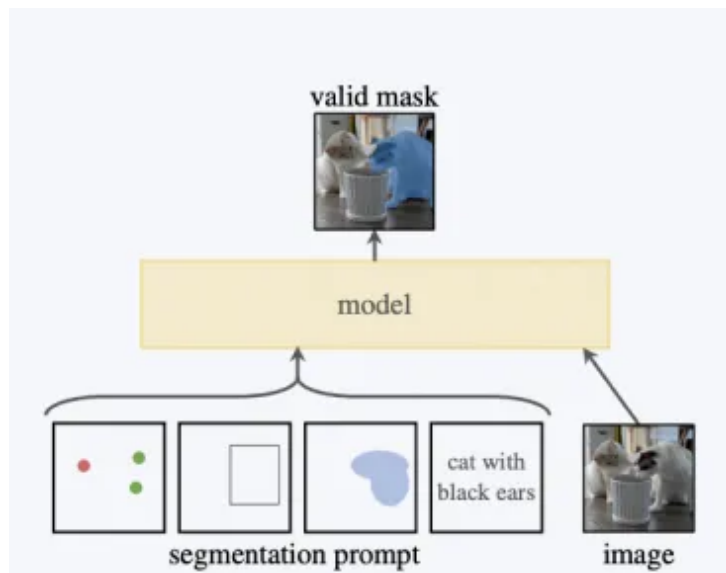


Figure 3: SAM offers promptable segmentation task, empowering data annotation.

3. Dataset

We have evaluated and continue to evaluate various public datasets regarding floods and natural disasters. Most of these use low-resolution satellite images or images taken from social media that are not suitable for the task at hand. For the time being, we have decided to use FloodNet 2021, which is described below.

For the time being, we have decided to use FloodNet 2021 and RescueNet 2022, both described below, for their high-quality images and the relevant chosen classes.

FloodNet 2021

The FloodNet dataset, as described in [9], is a high-resolution aerial imagery dataset captured by small Unmanned Aerial vehicles (sUAV). It is specifically designed to aid in post-flood damage assessment by providing detailed visual data that can be used for three computer vision tasks: classification, semantic segmentation, and VQA (Visual Question Answering).

The flights were conducted at the start of September 2017 in Fort Bend County, Texas, and other areas affected by the passage of Hurricane Harvey. The altitude was approximately 60 metres, thus the images are rich in detail and have better resolution compared to others from public databases of natural disasters, which are primarily composed of satellite observations.

A total of 2343 images have been labelled with 9 classes (see Fig. 4 for a sample image and its related labelling). To distinguish between natural water and flood water, a "water" class has been created, representing any natural water body such as rivers and lakes. Water can also be recognized in the "pool" class. Flooding is detected in the "road-flooded" and "building-flooded" classes: a building is classified as flooded when at least one side of the building is touching the flood water. Other labels include "building-non-flooded," "road-non-flooded," "tree," "vehicle," and "grass."

If more than 30% area of an image is occupied by flood water then that area is classified as flooded, otherwise as non-flooded.

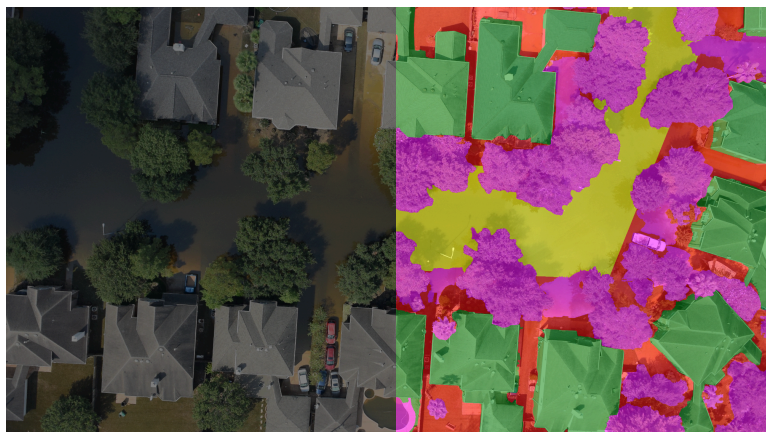


Figure 4: Sample image from the FloodNet dataset (left image) and related labelling (right image).

Regarding the semantic segmentation task, the authors of [9] present interesting performance results calculated on different models. The metric used is the Intersection Over Union (IoU) per-class and averaged across all classes. The results will constitute an upper bound for the Bayes Optimal Error (BOE) of the task.

RescueNet 2021

The RescueNet dataset [10], collects 4494 aerial images captured by two kinds of sUAVs from different Florida's cities struck by the Michael Hurricane in 2018.

The acquisition phase needed around 80 flights carried out with a DJI Mavic Pro quadcopters and Parrot Disco fixed-wing UAVs over a span of several days.

The main aim of such a survey was to document the post-event situation with a specific focus on the entity of the damages caused by the natural event. In particular it wanted to address the issues posed by the damage assessment and the creation of a rescue plan. For tackling such problems, the dataset has been labelled to train classification and segmentation algorithms.

In particular the dataset included 11 classes, some quite similar to the ones of FloodNet (Background, Water, Vehicle, Tree, Pool) but some other, instead, put a significant focus on the possibility of quantifying the amount of damage reported. For instance, the Building class has the possibility of being specified with greater detail by the presence of the subclasses No Damage, Minor Damage, Major Damage, Total Destruction that allow a fine-grained type of information. Something similar has been done with the Road class, some of which are Clear and some others have been Blocked.

Due the peculiarities of this dataset, along with FloodNet, it serves as a benchmark for testing state-of-the-art segmentation models, demonstrating its utility in improving methodologies for assessing natural disaster impacts.

4. Training methods

We are currently verifying how the selected semantic segmentations models behave with a complete training on the dataset for a variable number of epochs. Our aim at this stage is to select the best model that can come out of such training and use it as a baseline for the experimentation on two training techniques that are expected to provide better results: 1) data augmentation and 2) transfer learning.

Data augmentation is a technique used to increase data variety by randomly cropping or rotating parts of the dataset in order to artificially increase the samples on which the neural network is trained.

Transfer learning, on the other hand, is a training technique that pre-trains the weights of the neural network on a wide and robust dataset that exhibits features that are quite similar to the target dataset. Once the training is over, all the layers except the last ones are frozen and the network is retrained on the target dataset. The aim is to learn the basic feature recognition during

the pre-training and then fine-tune the algorithm to specialise it into recognizing the specific features of the target dataset.

More specifically, supported by the encouraging results of the RescueNet paper [10], we intended to train the neural network on the RescueNet dataset since it has more samples and presents all the features considered interesting to address our problem. After a complete training, the idea is to freeze all the weights but the ones of the last layer, then the training will go on using the target dataset, in our case FloodNet. The neural network, in the second phase of the training procedure, should learn the way of interpreting the results of the already learnt convolutional filters in a way that is optimized for FloodNet.

5. Training outcomes

Here we give a detailed overview of the training process and transfer learning that was operated on the architecture built using Tiramisù's paper as a reference[2]. We would like to express our gratitude to the authors of [this repository](#), whose work was fundamental in providing the first implementation of Tiramisù in PyTorch.

In order to be able to measure the differences and improvements across the multiple experiments that we did, we had to fix some hyperparameters. While the complete the source code with all the hyperparameters set is available [in the repository](#), we specify here the most relevant ones and the reason for which we opted for such values:

- the dataset images are rescaled at 950x713 pixels, this is to compress more space into the same image so that the convolutions (since we are not using attention mechanisms, still) are able to capture the complexity and the relationships of the objects present in the image;
- the dataset images are sampled randomly in order to get tiles of 713x713 pixels, this is to get square images that can be slightly different from each other even if they are sampled from the same initial image;
- the data augmentation performed is the same across all the experiments that we did so far, we do the crop specified above and we flip them horizontally;
- the training is carried out for 400 epochs. According to our empirical tests, it was enough to ensure convergence without overfitting.

The final goal of the training is a model that can perform reasonably well on the FloodNet dataset therefore the first step was to get a *baseline* training, against which every comparison should be

made. The metric used to measure the effectiveness of the classifier is the IoU (*Intersection over Unit*) that comes very handy in case of segmentation tasks.

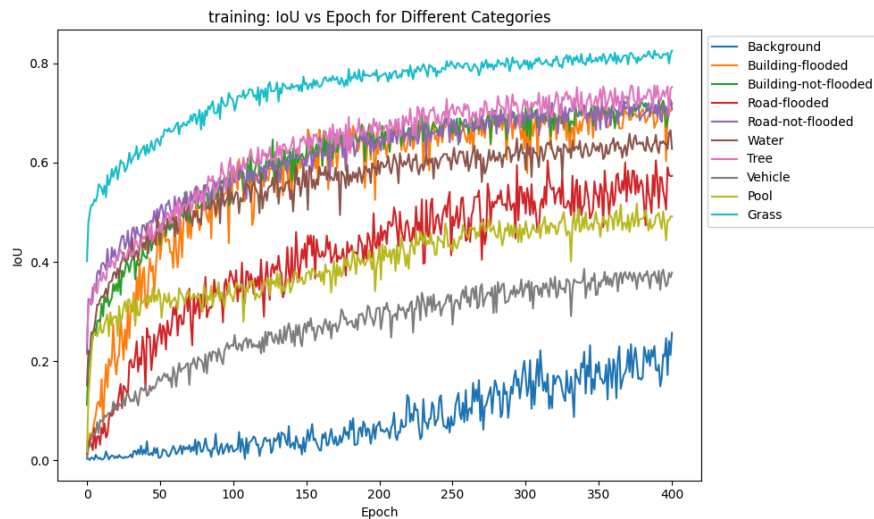


Figure 5: Accuracy of the baseline training, 713x713 tiles from the FloodNet dataset

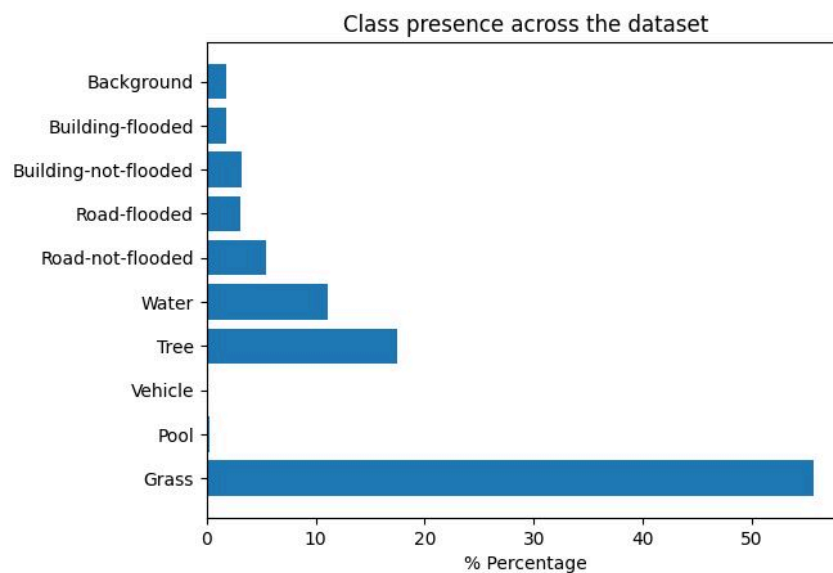


Figure 6: Bar plot that shows how much the classes are represented within the dataset

The plot above (Figure 5) shows that some classes can get a proper accuracy score around 70%, that is a good starting point but we see a lot of variance in the results and that is exactly due to the unbalanceness of the classes gathered in the dataset (Figure 6).

Since the FloodNet dataset has relatively few images and we need a more robust model, we use the *transfer learning* technique and so that the initialization of the weights is based on another model (same architecture) trained on the RescueNet dataset.

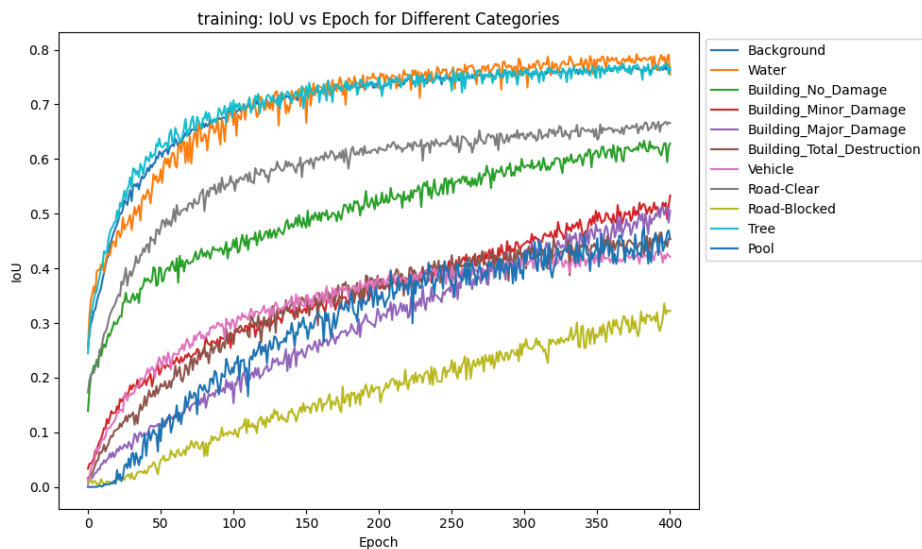


Figure 7: Accuracy of the preliminary training, 713x713 tiles from the RescueNet dataset

We tried the technique of Gradient Accumulation, since batch sizes are very small. Gradient Accumulation is a method used in machine learning to simulate larger batch sizes by accumulating gradients over multiple mini-batches before updating model parameters. This approach did not yield significant results and, in fact, worsened the training metrics. Therefore, we abandoned it.

Initially, we experimented with the optimizer used in the original implementation, RMSprop, testing various parameter values. After evaluating the results, we opted for the Adam optimizer, as described in the PyTorch 2.6 documentation. Additionally, we introduced a learning rate scheduler, 'ReduceLROnPlateau', to adjust the learning rate dynamically based on validation performance.

The training process was conducted for a total of 500 epochs. The results are shown in Figure 8.



Figure 8: Accuracy of the training using the Adam optimizer in combination with the 'ReduceLROnPlateau' learning rate scheduler, 713x713 tiles from the RescueNet dataset

and in validation we obtain the results shown in figure below.

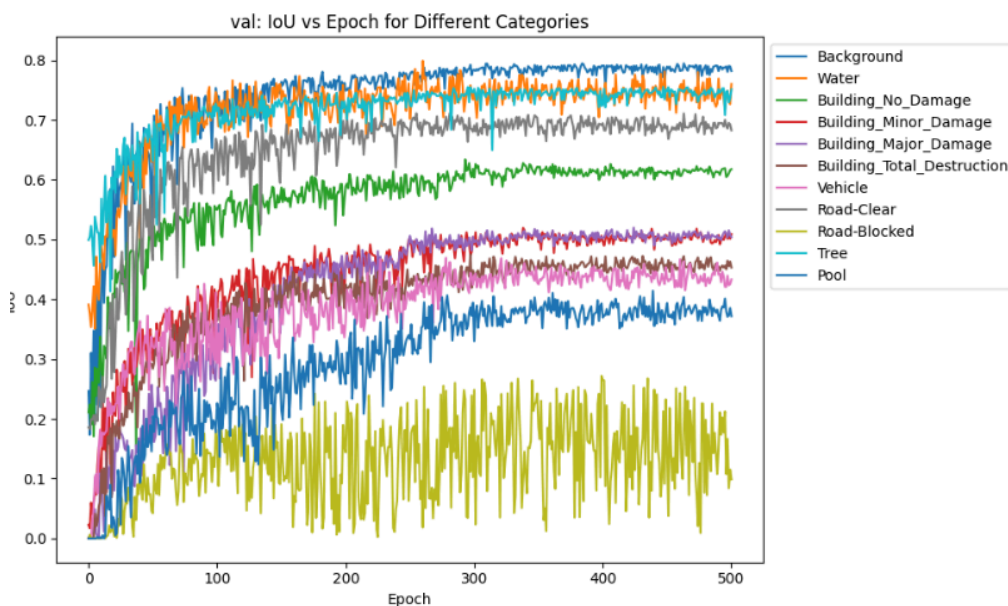


Figure 9: Accuracy of the validation using the Adam optimizer in combination with the 'ReduceLROnPlateau' learning rate scheduler, 713x713 tiles from the RescueNet dataset

The issue of overfitting becomes more evident when analyzing the Intersection over Union (IoU) values for each individual class, comparing training and validation performance. Specifically, overfitting emerges between epochs 200 and 400, depending on the class under consideration.

To mitigate this, we attempted to increase the L2 regularization parameters of the Adam optimizer; however, this did not yield any significant improvements. Alternative regularization techniques, such as dropout or data augmentation, could be explored. However, we opted to forgo these approaches and proceed with evaluating the next set of architectures.

Below are the results of the Tiramisù model on the RescueNet dataset in the test set. Early stopping of 399 and 395 epochs was chosen, appearing to have the most relevant performances.

The comparison is based on the results reported in the RescueNet paper [10].

model	water	build_no_d	build_minor_d	building_major_d	building_total_d	vehicle	road_clear	road_blocked	tree	pool	mIOU
Tiramisù e399	68.7	60.9	51.3	48.9	54.9	40.5	67.2	28.4	71.6	48.3	54.1
Tiramisù e395	75	60.9	50.4	49	55.4	40.3	66.5	7.3	73.9	46.7	52.5
DeepLabv3+	78.4	61.6	49.6	47.3	57.2	49.6	69.7	32.9	78.6	49.4	57.4
Segmenter (ViT-Tiny)	94.8	65.2	44.4	53.2	89.5	47.7	87.8	51.1	97.2	54	68.5
Segmenter (ViT-Small)	97.5	80.7	72.4	79.7	95.8	66.7	94.6	84.1	98.6	87.66	85.8
PSPNet	98.7	95.2	94.4	96.9	98.9	86	98.	94.8	99.4	94.31	95.7
Attention UNet	99.1	99.5	99.6	99.6	99.9	91.2	98.9	98.7	99.6	98.62	98.5

Table 1: Test of the Tiramisù model on test RescueNet dataset in comparison with other models from [10]

The following Figure 6 shows the generated masks on the RescueNet test set.

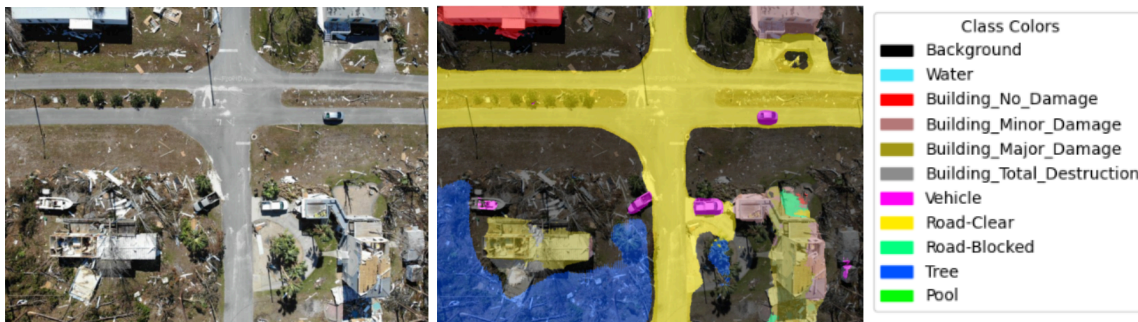


Figure 10: Sample generates mask of Tiramisu model on the RescueNet dataset

6. Transfer learning

RescueNet has different classes and still shows (see e.g. Figure 7, 8) some problems related to the presence of some underrepresented classes. But, since our aim is to use the parameters learned as a starting point for the last training, there is still the possibility that the convolutional filters learned with RescueNet will be useful also on FloodNet.

We made many experiments and comparisons between the models trained on FloodNet using the weights learnt from RescueNet as a starting point, but we present only the two that looked more relevant.

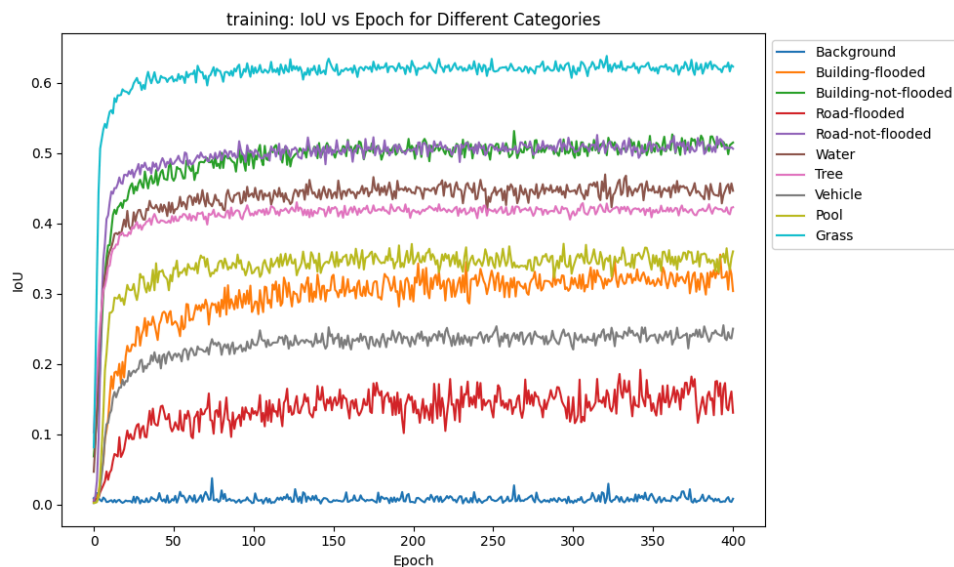


Figure 11: Accuracy of the training with transfer learning of the last layer, 713x713 tiles from the FloodNet dataset

In the first experiment (Figure 11) we froze (prevented from learning) all layers except the last one, the classifier. As expected the learning proceeds very fast and in less than 150 epochs we already reached a quite stable result. The performance of such a model is even worse than the baseline for many classes and this could be explained from the fact that the convolutional filters are still the ones of RescueNet and are kind of incompatible with the ones that would allow the net to perform reasonably well on FloodNet. In order to tackle such an issue, we designed the second experiment.

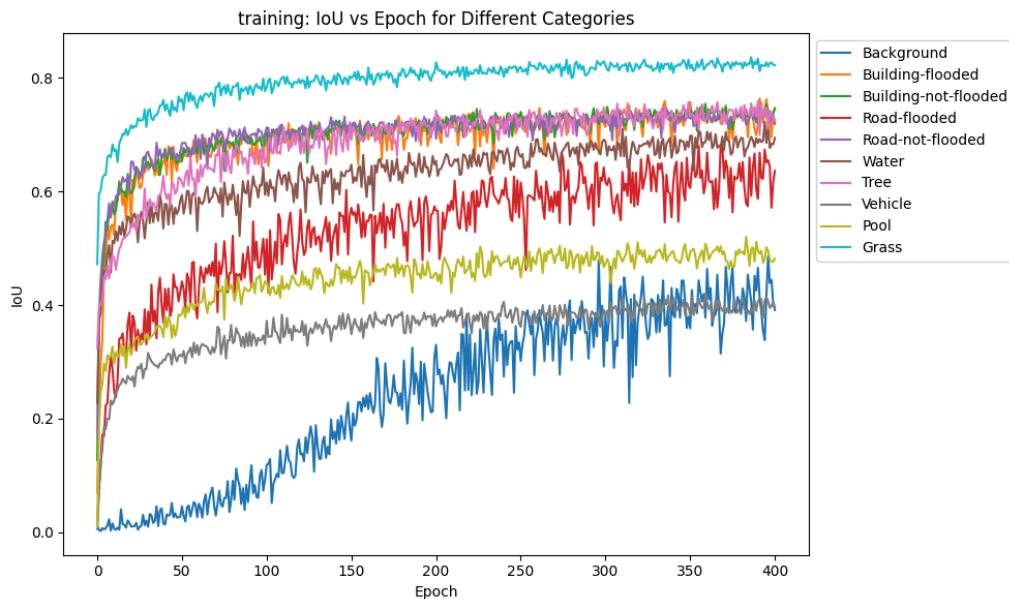


Figure 12: Accuracy of the training with transfer learning of the decoding part, 713x713 tiles from the FloodNet dataset

Here (Figure 12), instead, we froze only the initial encoding part (the first part of Tiramisù that compresses the information) and we let the training operate so that the convolutional filter of the decoding part could be learnt from scratch. Here we can notice a dramatic improvement in comparison with the first experiment and many classes are actually able to converge quite fast.

Currently, an experiment is ongoing, focusing on training based on the best transfer learning configuration and the weights computed on RescueNet, as presented in Table 1. The updated results will be published soon. However, by comparing the results of the last model (Figure 12) with the one of the baseline (Figure 5), we get that overall transfer learning contributed to improve the results but we can definitely say that the advantage is marginal. As a conclusion, the practical application of Tiramisù trained on the examined datasets could be quite limited even after applying transfer learning. Another possibility is to implement attention mechanisms that could, in principle, make more sense of the visual scene and be more efficient in detecting the objects even if they are underrepresented.

7. Deployment and Web Service

The models are currently being trained on internal INAF's resources, in particular we are using the cluster Pleiadi¹ that allows us to exploit the parallelism offered by the GPU accelerator.

We are transferring the training environment from the previous infrastructure to the Kubernetes cluster provided by the University of Turin as a PoCI [17]. This step involves creating proper Docker images and Kubernetes projects to establish a convenient development environment. In synergy with Work Package 1, we developed a container image with the appropriate CUDA-enabled PyTorch libraries wrapped in a Conda environment, ready to execute the training scripts. Additionally, we implemented a set of tools, to facilitate a smooth testing process, which will be organised into Helm Charts for ease of use.

We are also working on a GUI that will speed up the selection and evaluation of the models that are going to be trained during the activity.

In particular we would like to be able to submit manually and quickly some specific inputs without having to copy them in the accelerated environment and run the inference via command line. We are using a Python library called Gradio² for such a purpose since it allows us to wrap the whole inference pipeline into a handy web GUI (see Fig. 13) and, if needed, also a REST API able to run the inference.

¹ <https://pleiadi.readthedocs.io/en/latest/>

² <https://www.gradio.app/>

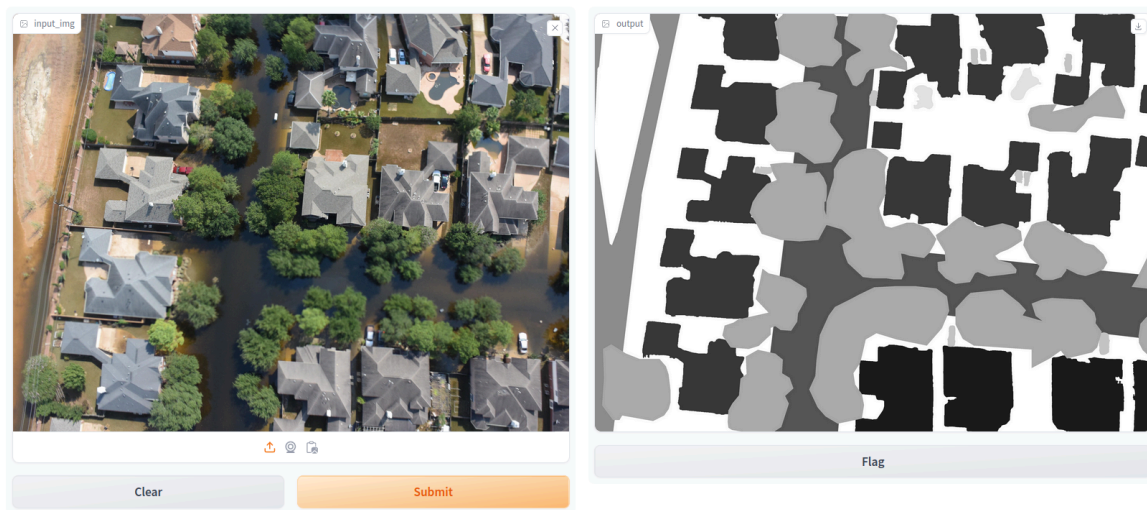


Figure 13: Example of the interface developed with Gradio that allows to check the results very fast and in a very intuitive way.

We plan to integrate such interfaces in the container in order to speed up the evaluation of the models and the development iterations.

8. CONCLUSIONS

This document reports the process of development of an image semantic segmentation algorithm capable of speeding up the detection of flooded areas and related damages. It describes the study of the state of the art algorithms for such a class of problems and the challenges faced during the training and the training dataset selection.

The path we decided to take in order to achieve the goal looks promising: we are currently getting encouraging results and we have a robust plan for raising the bar step-by-step as we proceed with the development.

The next phase will be on experimenting on further models such as the Attention U-Net and Segmenter and doing some benchmarks on both drone and satellite sourced images and check if the model behaves consistently as we expect.

The final step will be the integration of our solution with the digital twin developed by other tasks in the WP2.

REFERENCES

- [1] L. Pelonero and F. R. Vitello, "Algorithm for creation of digital twin from UAV," Feb. 2024. [Online]. Available: <https://doi.org/10.15161/oar.it/143691>
- [2] S. Jégou, M. Drozdal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2017, pp. 11–19.
- [3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.
- [4] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.
- [5] O. Ronneberger, P. Fischer, and T. Brox, "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. Springer, 2015, pp. 234–241.
- [6] R. Sortino, D. Magro, G. Fiameni, E. Sciacca, S. Riggi, A. DeMarco, C. Spampinato, A. M. Hopkins, F. Bufano, F. Schillirò et al., "Radio astronomical images object detection and segmentation: a benchmark on deep learning methods," Experimental Astronomy, vol. 56, no. 1, pp. 293–331, 2023.
- [7] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo et al., "Segment anything," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 4015–4026.
- [8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly et al., "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv preprint arXiv:2010.11929, 2020.
- [9] Rahnemoonfar, Maryam & Chowdhury, Tashnim & Sarkar, Argho & Varshney, Debvrat & Yari, Masoud & Murphy, Robin. (2020). FloodNet: A High Resolution Aerial Imagery Dataset for Post Flood Scene Understanding.
- [10] Rahnemoonfar, M., Chowdhury, T., and Murphy, R., "RescueNet: A High Resolution UAV Semantic Segmentation Dataset for Natural Disaster Damage Assessment", vol. 10, no. 1, Art. no. 913, 2023. doi:10.1038/s41597-023-02799-4.

- [11] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. (2015). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. arXiv preprint arXiv:1511.00561.
- [12] Chao, P.-H., Kao, C.-Y., Ruan, Y.-S., Huang, C.-H., & Lin, Y.-L. (2019). HarDNet: A Low Memory Traffic Network. arXiv preprint arXiv:1909.00948.
- [13] Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., ... & Zhou, S. K. (2021). TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. arXiv preprint arXiv:2102.04306.
- [14] Zhao, H., Shi, J., Qi, X., Wang, X. and Jia, J. (2017). Pyramid Scene Parsing Network. arXiv:1612.01105 [cs].
- [15] Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J. M.; Luo, P. (2021) SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. arXiv:2105.15203.
- [16] Chen, L.-C., Papandreou, G., Schroff, F., & Adam, H. (2017). Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. arXiv preprint arXiv:1706.05587.
- [17] Colonnelli, I. et al “HaMMon PoC PaaS Technical report.” Internal technical report. March 2024
- [18] Strudel, R., Garcia, R., Laptev, I., & Schmid, C. (2021). Segmenter: Transformer for Semantic Segmentation. arXiv:2105.05633.
- [19] Francesco, Franchina, Sciacca, Eva, & Imbrosciano, Mauro. (2024, July 9). Development of a solution for image segmentation with focus on flooding (Version 1.0).
<http://doi.org/10.15161/oar.it/211549>