Interreg Co-funded by the European Union

Italy – Croatia

STRENGTH

D.1.2.1
Survey of appropriate Machine Learning (ML) /Artificial Intelligence system

Interreg Co-funded by the European Union

Italy – Croatia

STRENGTH

Project acronym	STRENGTH
Project full title	STRategies for assessing climate change and natural hazards' impact on urban ecosystems, increasing resilience to ENvironmental hazards, and promoting territorial GrowTH
Programme	Interreg Italy-Croatia 2021-2027
Start date	01/04/2024
End date	30/09/2026
Project ID	ITHR0200318

Deliverable Title	D.1.2.1 - Survey of appropriate Machine Learning (ML) /Artificial Intelligence system
Activity	Digital monitoring to detect seismic vulnerability
WP	1
WP Leading Partner	FESB
Contributing Partners	FGAG, UNIFE
Dissemination level	Public
Version	Finalised
Date	30/09/2025

Table of contents

Executive Summary

This document provides an expanded survey and implementation guide for a practical two-stage façade analysis pipeline using thermal and RGB imagery. It covers network families, training feasibility in MATLAB/Python, recommended architectures from literature, a workflow diagram, tables, and references.

1. Introduction

Accurate interpretation of building façades from thermal and RGB imagery is usually being achieved through a two-stage process, where façade layout parsing and fine-grained masonry material segmentation are carried out in sequence. Automated structural assessment, material mapping, and digital-twin generation are being enabled when these stages are completed in a reliable way, although in practice the results can depend on the dataset condition. Deep learning is generally regarded as the most effective framework for this purpose, since very high performance has been demonstrated in both pixel-level segmentation and object-level detection tasks, even if some models can behave a bit differently under varying illumination. In this report, foundational segmentation architectures such as U-Net [1] and DeepLab [2] are being taken into account, while state-of-the-art transformer-based models, including SegFormer [3], are also being considered because their capability for more global contextual reasoning has been shown.

For the detection of façade components like windows and doors, established object-detection methods—including Faster R-CNN [4] and the YOLO family [5][6]—are being incorporated. When combined, these models are used for forming a practical and trainable pipeline that is compatible with MATLAB and Python environments for more comprehensive façade analysis, even if some additional tuning is usually needed. In general, accurate interpretation of building façades from thermal and RGB images is being understood as requiring these two main stages: façade layout parsing and fine-grained masonry material segmentation. Deep learning is therefore being viewed as the most robust approach, and the architectures that can be trained in MATLAB or Python are being summarised in this document together with a practical pipeline for their implementation.

2. Literature review

Early façade analysis was predominantly conducted using handcrafted features, thermal thresholding techniques, and rule-based image processing. These traditional approaches were often constrained by their sensitivity to environmental variation, imaging noise, and inconsistencies in façade texture. With the emergence of deep learning, a pronounced shift in both accuracy and robustness was observed. Convolutional neural networks, most notably U-Net [1], were shown to deliver exceptional performance on structural textures owing to their encoder–decoder architecture, which had been designed specifically for dense prediction tasks. Subsequent enhancements were introduced through the DeepLab family of architectures [2], in which multi-scale semantic understanding was improved by the incorporation of atrous (dilated) convolutions and atrous spatial pyramid pooling.

Further progress was achieved with the adoption of transformer-based frameworks, such as SegFormer [3] and HRNet [7]. In these models, high-resolution spatial representations were deliberately preserved, and long-range dependencies were captured more effectively, resulting in improved segmentation of complex façade surfaces. Owing to their global receptive fields and highly efficient feature-fusion mechanisms, substantial gains were realised in boundary delineation and texture differentiation.

In parallel, significant developments were observed within the field of object detection. Notable advances were made when Faster R-CNN [4] was introduced, as region proposal networks were employed to generate high-quality candidate regions in a fully trainable manner. The YOLO family of architectures [5][6] subsequently demonstrated that object detection could be consolidated into a single, end-to-end predictive pipeline capable of real-time operation. These detectors have since been widely adopted for the identification of façade components including windows, doors, and other architectural openings.

More recently, instance-level segmentation has been advanced through architectures such as Mask2Former [9], in which a unified transformer backbone has been employed to address semantic, instance, and panoptic segmentation within a single framework. Through this design, substantial flexibility and strong generalisation capacity have been achieved, enabling a wide range of façade-related segmentation tasks to be addressed consistently and with improved accuracy.

3. Dataset preparation

High-quality datasets are usually considered essential for achieving more reliable segmentation, and because of that several preparation steps are normally carried out.

- Data Collection: RGB and thermal imagery of façades are captured under stable lighting conditions, and thermal calibration is ensured.
- Annotation Tools: Pixel-level labelling will be performed by using tools such as CVAT or LabelMe. The façade areas, windows, doors, and material classes like brick, stone, concrete and plaster will be annotated.
- Patch Extraction: Overlapping patches (256–512 px) are created so that the GPU memory can be more balanced with the needed representation.
- Data Balancing: Material classes are sampled more evenly in order that underfitting of smaller classes.
- Augmentation: Flips, rotations, and mild brightness changes will be applied. However, more extreme augmentations, especially for thermal images, will be avoided since they can disturb the temperature information, what is not desirable.

4. Methodology and full data pipeline description

1. Methodology

Methodology follows a modular two-stage pipeline:

Stage 1 – Façade Layout Parsing:

- Use U-Net/DeepLabv3+ for full-scene segmentation or YOLO/Faster R-CNN for window/door detection.
- Output: façade mask, window masks, door masks.

Stage 2 – Masonry Material Segmentation:

- Apply U-Net++, DeepLabv3+, or SegFormer for material classification only within façade regions.
- Classes include brick, stone, concrete, plaster, background.
- Post-Processing:
- Morphological smoothing for segmentation refinement.
- Connected component labeling to remove noise.
- Material proportion quantification.

2. Data Pipeline

The data workflow is as follows:

1. Acquire thermal (and optionally RGB) façade imagery.
2. Preprocess images: alignment, denoising, normalization.
3. Stage 1 models compute façade mask + detect openings.
4. Mask out non-façade regions.
5. Stage 2 model segments masonry materials.
6. Post-processing: smoothing, connected components, area calculation.
7. Generate façade maps and material statistics.

This pipeline ensures modularity and strong performance even with moderate dataset sizes.

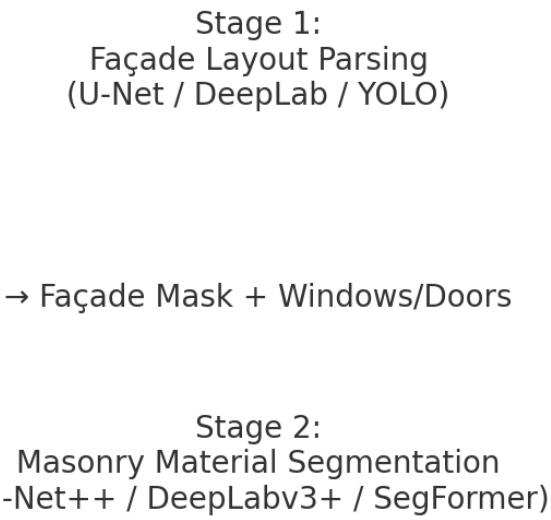

Figure 1. Visual Representation of data pipeline

5. Implementations

MATLAB Implementation:

- Use Deep Learning Toolbox for training U-Net, SegNet, DeepLab.
- Use built-in apps for data labeling.
- Compatible models: U-Net, SegNet, DeepLabv3+, Faster R-CNN.

Python Implementation:

- Use PyTorch with segmentation-models-pytorch for U-Net++, DeepLab, HRNet, SegFormer.
- Use Ultralytics for YOLOv8 and YOLOv8-Seg.
- Export trained Python models to ONNX for MATLAB inference compatibility.

Table 1. Comparisons of Matlab/Python trainableity

Model	Type	Trainable in MATLAB?	Trainable in Python?
U-Net / U-Net++	Segmentation	Yes	Yes
SegNet	Segmentation	Yes	Yes
DeepLabv3+	Segmentation	Yes	Yes
SegFormer	Segmentation (Transformer)	Inference only	Yes
Mask2Former	Segmentation (Transformer)	No	Yes
YOLOv8	Object Detection	Partial	Yes
YOLOv8-Seg	Instance Segmentation	No	Yes
Faster R-CNN	Object Detection	Yes	Yes

6. Conclusion

The reviewed architectures are being shown as highly feasible for façade-level segmentation tasks, and good performance was observed across different data conditions. It is often noted that U-Net and DeepLab are performing well even when smaller datasets are used, while transformer-based methods are usually requiring larger training sets but are giving more superior global context modelling. Detection networks such as YOLOv8 are being considered advantageous when more precise window and door localisation is needed, although sometimes their behaviour can be a bit sensitive to difficult lighting.

Thermal–RGB fusion is also being regarded as an additional layer of interpretability, especially useful for material classification where emissivity differences are complementing texture information from RGB images, even if the alignment between the two modalities is not always perfect in practice.

Overall, a complete pipeline for façade layout parsing and masonry material segmentation is being consolidated in this report. The surveyed architectures—including U-Net [1], DeepLab [2], SegFormer [3], Faster R-CNN [4], YOLO models [5][6], HRNet [7], SegNet [8], and Mask2Former [9]—are providing strong foundations for constructing a more robust classification system that can be deployed in both MATLAB and Python environments, although some additional tuning would usually be needed depending on the dataset.

7. References

[1] Ronneberger et al., U-Net: Convolutional Networks for Biomedical Image Segmentation, 2015.

[2] Chen et al., DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, 2017.

[3] Xie et al., SegFormer: Simple and Efficient Design for Semantic Segmentation, 2021.

[4] Ren et al., Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, 2015.

[5] Bochkovskiy et al., YOLOv4: Optimal Speed and Accuracy of Object Detection, 2020.

[6] Jocher et al., YOLOv5/YOLOv8 Documentation, Ultralytics.

[7] Zhao et al., HRNet: High-Resolution Representations for Vision, 2020.

[8] Badrinarayanan et al., SegNet: Encoder-Decoder Architecture, 2017.

[9] Cheng et al., Mask2Former: Unified Segmentation Framework, 2022.