# STUDY OF A SEMANTIC SEGMENTATION ALGORITHM FOR DISASTER ASSESSMENT IN AN EDGE COMPUTING ENVIRONMENT

**Gianluca Parnisari**
CTO Training System
TXT E-TECH s.r.l.
Cologno Monzese, Italy

**Enrico Targhini**
Doctoral Researcher
Politecnico di Milano
Milano, Italy

## ABSTRACT

When a natural disaster strikes, it often causes widespread destruction, making timely and accurate information essential for effective response and recovery. Traditional assessment methods, which rely on manual interpretation of aerial and satellite imagery, are limited by delays, high operational costs, and a high risk of human error. This project aims to enhance disaster assessment by leveraging deep learning and edge computing to enable fast, reliable analysis during emergencies. The primary objective is to develop a system capable of accurately identifying objects and classifying them based on the amount of reported damage. This system will then be adapted for real-time, automated semantic analysis on edge devices such as drones and IoT sensors. The goal is to minimize reliance on remote data centers, improve on-site decision-making, and support efficient disaster response in resource-constrained environments by balancing accuracy with computational efficiency.

## NOTATION

*

Acronyms

**DL** Deep Learning

**CV** Computer Vision

**CNN** Convolutional Neural Network

**FCN** Fully Convolutional Network

**WCCE** Weighted Categorical Cross-Entropy

**SGD** Stochastic Gradient Descent

**IoU** Intersection over Union

## INTRODUCTION

Natural disasters such as earthquakes, floods, and wildfires are catastrophic events that cause significant loss of life and widespread devastation to infrastructure, necessitating rapid and accurate damage assessment for effective emergency response. The introduction of deep learning (DL) models has significantly enhanced the accuracy and efficiency of damage estimation by automating feature extraction and enabling faster segmentation of affected areas (Ref. 1). Among the various Computer Vision (CV) techniques, semantic segmentation is particularly valuable in this scenario, as it assigns a label to each pixel in an image, boosting the classification of different structures and, potentially, the assessment of damage severity. However, despite the advancements brought by modern DL models, these scenarios still pose several challenges, including class imbalance, dataset occlusions, and the need for real-time computation to process data rapidly near the source.

Semantic segmentation is a core technique in damage assessment from aerial and satellite imagery. The first Convolutional Neural Network (CNN) models, introduced for the first time in 1998 with the LeNet model (Ref. 2), laid the foundation for modern segmentation methods, leading to the development of Fully Convolutional Networks (FCN), which preserve spatial information throughout the network to produce dense, high-resolution predictions.

A major breakthrough came with U-Net, an encoder-decoder architecture with skip connections that enhanced the retention of spatial details and greatly improved segmentation quality (Ref. 3). This particular architecture, as the name suggests, is composed by two main parts:

- An encoder, which modifies the original image to extract relevant features from objects in it. These features will then be used to discriminate different objects in the pic-

ture and classify better similar objects in different images.

- A decoder, which uses the extracted features to reconstruct the original image, with all pixels labeled to correctly identify the category they represent (example on image 1.)

Building on this foundation, advanced architectures such as DeepLabV3+, PSPNet, and transformer-based models like Segmenter have pushed the state of the art by capturing multi-scale features and global context, further improving performance on complex segmentation tasks.

In disaster assessment applications, these models have been widely used for tasks like flood delineation, building damage classification, and debris detection. U-Net and DeepLab, in particular, are frequently applied due to their ability to capture fine-grained structural details and delineate boundaries effectively. However, despite their success, these models are typically large and computationally expensive, making them unsuitable for real-time processing, especially on edge devices with limited resources.

These limitations forced recent research to shift toward lightweight and efficient architectures. Models such as ESPNet and LETNet have been introduced to reduce computational load while preserving acceptable segmentation accuracy. ESPNet (Ref. 4) employs efficient spatial pyramid modules for multi-scale feature extraction, while LETNet (Ref. 5) combines CNN-based processing with transformer components to enhance global feature understanding. These models demonstrate promising results in terms of speed and efficiency, but often still face trade-offs in segmentation accuracy or require extensive training on large-scale datasets, which may not be feasible in real-world disaster scenarios with limited data availability.

To face these challenges, this paper introduces U-YOLO, a novel DL model for semantic segmentation in disaster assessment, designed to balance high accuracy with low computational effort. The architecture builds upon YOLOv11, a state-of-the-art object detection framework (Ref. 6), and adapts it for segmentation tasks. By incorporating an optimized segmentation head inspired by U-Net, U-YOLO achieves significant improvements in efficiency while maintaining competitive accuracy compared to established segmentation models.

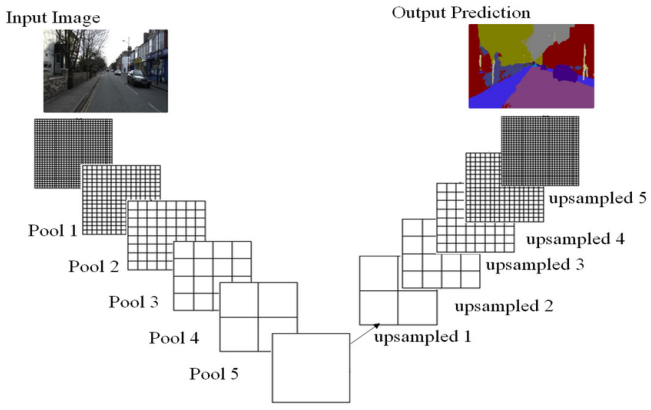


**Figure 1. Example of a classic encoder-decoder structure for semantic segmentation tasks.**

## METHODOLOGY

To fully evaluate the performance of the newly introduced model and analyze its strengths and weaknesses, it has been tested and compared against two state-of-the-art architectures: Attention U-Net and Segmenter. These models were selected as baseline comparisons for two main reasons. First, they represent modern yet fundamentally different architectures. This diversity allows to provide deep insights into how various components and design choices impact performance on the given task, leading to a sequential informed selection of features when designing the new model. Second, these architectures are considered the top performing models on the RescueNet dataset, as reported by its creators (Ref. 7). This makes them ideal benchmarks for assessing the effectiveness of the proposed approach.

### Attention U-Net

This architecture is a modified version of the well-known U-Net model, retaining its core structure while introducing enhancements to improve segmentation accuracy. While the original U-Net employs symmetric encoder-decoder paths with skip connections to retain spatial information, it treats all features equally when transferring information from the encoder to the decoder. In contrast, Attention U-Net enhances this mechanism by introducing Attention Gates within the skip connections, such as shown in 2. These gates learn to focus selectively on the most relevant regions of the feature maps, effectively filtering out irrelevant background information and highlighting salient structures—particularly important in complex or cluttered images such as post-disaster scenes (Ref. 8).

By dynamically weighting features based on their contextual importance, Attention U-Net improves the precision of segmentation boundaries, especially for small or ambiguous regions that may be overlooked in the standard U-Net. Importantly, this enhancement is achieved without a significant increase in computational cost, making it a practical upgrade for applications requiring high segmentation accuracy. In disaster assessment tasks, this ability to focus on critical damage areas while suppressing noise from surrounding regions makes Attention U-Net particularly effective compared to the baseline U-Net.

With a total of 49.9 million parameters, this model is highly effective in capturing fine details in pictures but comes at the cost of high computational complexity.

### Segmenter

This model heavily relies on the Vision Transformer architecture, representing a significant shift from traditional convolutional neural networks. A Vision Transformer is a modified

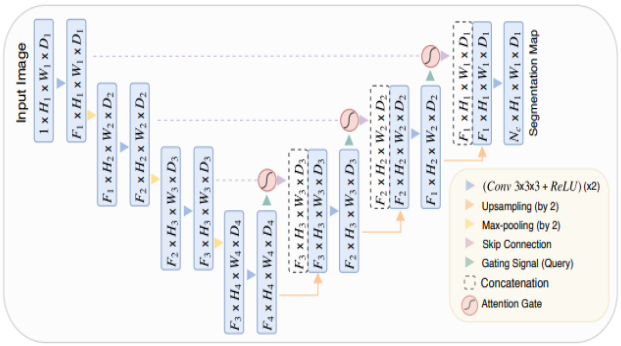


**Figure 2. Attention U-Net full model. Picture taken from Attention U-Net's paper (Ref. 8).**

version of the transformer framework originally developed for natural language processing, adapting it for dense prediction tasks like semantic segmentation. Segmenter processes input images by dividing them into fixed-size, non-overlapping patches, each of which is treated as an input "token". These tokens are then passed through a standard transformer encoder, allowing the model to capture global contextual relationships across the entire image (Ref. 9).

One of Segmenter's key advantages is its ability to model long-range dependencies effectively, which is particularly useful for segmenting objects that are spatially distant but semantically related. Additionally, the architecture is highly scalable and benefits from advances in large-scale pretraining on extensive image datasets, making it a powerful tool for high-level vision tasks.

However, Segmenter also comes with notable limitations. Its patch-based representation loses fine-grained spatial information, particularly when small objects or subtle boundaries are present. Unlike CNNs, which naturally preserve local detail through consecutive feature extraction, Segmenter's reliance on fixed patch sizes can lead to coarse outputs and reduced accuracy in segmenting fine details or small structures. Moreover, the model's computational complexity and memory demands are considerably higher than those of lightweight CNN-based alternatives, making it less suitable for real-time or edge-device deployment in time-critical scenarios like disaster response.

### The proposed model: U-YOLO

This new model is designed to directly tackle the key challenges of real-time disaster assessment in edge computing environments. It follows an encoder-decoder architecture, where the encoder utilizes YOLOv11's backbone for feature extraction, while the decoder, inspired by U-Net, reconstructs segmentation masks with precise pixel-wise classification while maintaining low computational overhead.

YOLO (You Only Look Once) is a widely adopted deep learning framework for object detection, known for its ability to perform detection in a single forward pass of the network. Unlike traditional detectors, which separate region proposal (which delimitate objects' boundaries in the image) and the actual category classification tasks, YOLO directly predicting bounding boxes and class probabilities from the input image. This unified architecture allows for extremely fast and accurate detection, making it highly effective for real-time applications.

Over the years, the YOLO family has evolved through multiple iterations—from the early YOLOv1 to more recent versions like YOLOv4, YOLOv5, and YOLOv8—each improving upon speed, accuracy, and architectural efficiency. Later versions introduced innovations such as better backbone networks and enhanced feature fusion techniques. However, many of these improvements were tailored to object detection tasks and not directly applicable to dense prediction problems like semantic segmentation (Ref. 12).

YOLOv11 (Ref. 6), the latest version in the YOLO (You Only Look Once) family of models, was selected as the foundation for the proposed approach due to its excellent balance of speed, accuracy, and efficiency. Originally designed for object detection, YOLO models are known for processing images extremely quickly by detecting all objects in a single pass. YOLOv11 builds on this legacy with improved feature handling and the ability to recognize patterns at multiple scales, allowing it to capture both small details and broader structures within an image.

Compared to earlier versions, YOLOv11 introduces enhancements that make it more accurate and lightweight, especially in challenging environments like disaster zones. One of its key advantages is its attention mechanism, which helps the model focus on the most important parts of an image while ignoring irrelevant background areas—this improves precision without slowing down performance. These strengths make YOLOv11 a powerful and adaptable choice for tasks like damage assessment, where both speed and detail matter.

Since YOLOv11 is primarily designed for object detection, U-YOLO replaces its detection head with a lightweight segmentation decoder. The new classifier is based on the well-known decoder of the U-Net architecture, utilizing a sequence of upsampling layers to gradually reconstruct the original image. It is further enhanced with skip connections, which link the YOLO backbone's feature maps to corresponding decoder layers, helping to restore spatial information lost during downsampling and improving boundary precision.

Additionally, the decoder is designed to adapt to the model size, preventing abrupt upsampling while keeping the number of parameters low. This hybrid approach combines YOLOv11's efficiency with U-Net's segmentation strength, enabling fast and accurate segmentation suitable for edge devices such as UAVs and IoT-based monitoring systems.

### Dataset

The dataset selected for this project is called RescueNet (Ref. 7), a high-resolution natural disaster dataset. The dataset was assembled by the Center for Robot-Assisted Search at

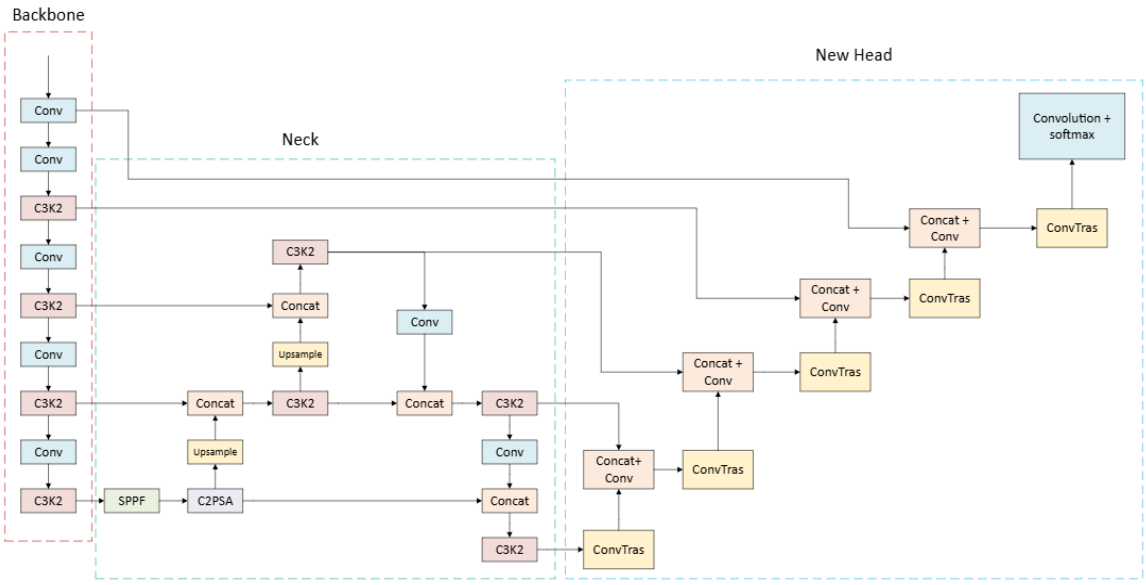


**Figure 3. Architecture of U-YOLO.**

Mexico Beach, Florida, where Hurricane Michael reached the coast as a category 5 hurricane on October 10, 2018. All pictures were taken using small UAVs operating at an altitude of over 400 feet above ground level. Each photo has a resolution of $3000 \times 4000$ pixels, with an approximate ground-level resolution of 4.5–5 cm per pixel. This high resolution ensures an exceptional accuracy, allowing for the detailed identification of structural features, damage levels, and finer elements within the scene, making the dataset particularly suited for capturing small details in underrepresented classes.

RescueNet provides a pixel-level annotation for 10 different classes, making it suitable for complex semantic segmentation tasks. The classes represent six main categories:

- **Tree**: This category includes all standing trees or small forests within the scene. These elements typically represent the natural green environment surrounding urban or suburban regions. This category comprehends the *22.5%* of total pixels, making it the most represented class.
- **Water**: This class includes all visible water bodies such as ponds, artificial canals or the sea. This category, like the tree class, is also largely represented, with *8.1%* of total pixels, which is plausible since photos were taken near the ocean.
- **Building**: This is one of the most critical classes, since it represent all the buildings present in the scene stroke by the storm. In order to correctly evaluate the amount of damages reported by buildings, this class is further subdivided into four distinct subclasses based on the degree of structural damage. This subdivision is done according to FEMA guidelines (Ref. 10), a document drafted by the U.S. Department of Homeland Security to evaluate damage assessment and determine consequential individual or public assistance in different cases. The obtained classes are:
  - **Building No Damage**: Refers to buildings that show no apparent structural damage and remain intact and fully functional. Between buildings categories, this is the most present in the dataset, with *2.65%* of total pixels.
  - **Building Medium Damage**: These buildings exhibiting partial damage, such as missing roof tiles, broken windows, or cracks in walls. This category has a similar distribution of the previous class, with *2.61%* of total pixels.
  - **Building Major Damage**: These structures are significantly damaged and may require extensive repairs before they can be safely reoccupied. These buildings are less present in the dataset, with only *1.65%* of pixels.
  - **Building Total Destruction**: Refers to buildings that have collapsed entirely or are no longer standing. Only the foundations or scattered debris are typically visible. These is the least represented building class, with *1.42%* of total pixels.
- **Road**: This class represents road networks within the scene and is divided into two subclasses based on their condition after a disaster. Similarly to the buildings classes, it is further divided in two subclasses.
  - **Road Clear**: Roads that remain unobstructed and passable for vehicles and pedestrians. This cate-

gory is largely present in the dataset (*6.98%*), meaning that overall roads are not largely affected by the storm.

  - **Road Blocked**: Roads that are obstructed due to rubble, debris, or other obstacles, rendering them non-navigable. The distribution of this class is consistently lower that the other road class (*1.57%*).

- **Vehicle**: This class includes visible motor vehicles such as cars, trucks, or vans. Vehicle presence can provide insights into evacuation patterns or the status of urban mobility. Due to the smaller size of vehicles respect to buildings or natural surroundings, this class is poorly represented in the dataset, with only *0.33%* of total pixels.

- **Pool**: Refers to private swimming pools present in the houses' garden. Since swimming pools are not so usual, this class is the least represented one in the whole dataset (*0.06%* of pixels).

Overall, all these characteristics make the dataset perfect for the task. It presents different categories depending on the severity of structural damage, it shows an obvious class imbalance and it has an extremely high resolution, which allows the models to capture finer segmentation aspects. A graphical example of three dataset's samples is reported in figure 4.

A major concern with the dataset is the presence of annotation inconsistencies. Some images exhibit incoherent debris labeling, since rubble is sometimes annotated as background and other times as part of buildings or roads. Additionally, irregular building boundaries in certain images may lead to misclassifications and poor generalization during model training. For example, in picture 2 of figure 4, it can be seen that the road is clearly obstructed by sand and debris, but it is annotated as clear. Moreover, looking at the right building, it is annotated as undamaged while it is clearly collapsed. These challenges must be carefully addressed to ensure fair model evaluation and a reliable generalization degree.

## EXPERIMENTAL SETTINGS

To ensure a fair and consistent comparison between models, all experiments were conducted under the same settings. These choices reflect commonly used practices in machine learning to improve training stability, performance, and generalization.

The models were trained using a loss function known as **Weighted Categorical Cross-Entropy (WCCE)**. A loss function measures how well the model is performing by comparing its predictions to the correct answers. WCCE is a modified version of the standard categorical cross-entropy loss, a well-known loss function which assigns a penalization to prediction performed over wrong classes (Ref. 13). The WCCE function behave exactly like the standard CCE, but also assigns different importance values, called weights, to each class. This is especially useful when some classes appear less

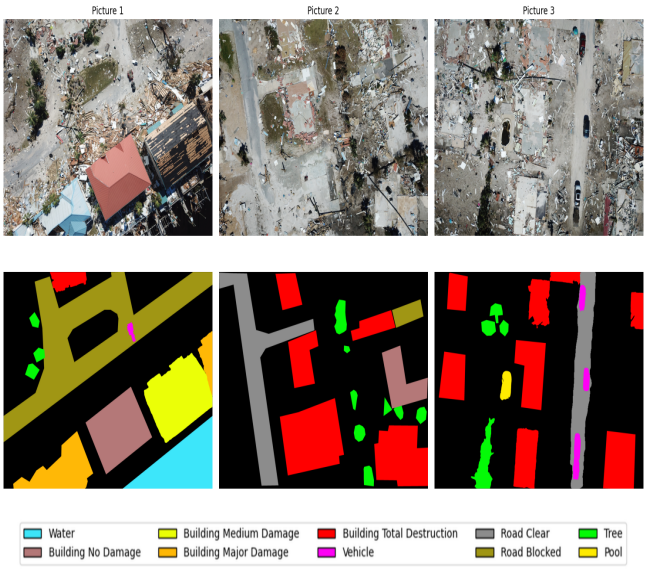


**Figure 4. Three dataset examples, with the original image above and the corresponding label below.**

frequently or are more difficult to learn. By assigning higher weights to these underrepresented or challenging classes, the model is encouraged to learn them better and not ignore them. The WCCE is defined as in 1:

$$L_{\text{WCCE}} = -\sum_{i=1}^{C} w_i y_i \log(\hat{y}_i) \tag{1}$$

where:

- C is the number of classes,
- $y_i$ is the true label for class,
- $\hat{y}_i$ is the predicted probability for class,
- $w_i$ is the weight assigned to class.

For optimization, the **Stochastic Gradient Descent (SGD)** algorithm with momentum was used as the main optimizer. In deep learning, an optimizer is responsible for updating the internal parameters of the model (also known as weights) in order to reduce the prediction error. A key component of this process is the **Learning Rate**, which is a small positive value that controls how big each update step should be. If the learning rate is too high, the model may overshoot the optimal solution and fail to converge. If it's too low, training becomes very slow and may get stuck in suboptimal results.

SGD performs these updates using only a small, random subset (or batch) of the training data at each step, instead of using the entire dataset. This makes the training process faster and requires less memory (Ref. 14). The addition of **momentum** further enhances the optimizer by allowing it to accumulate the direction of past gradients. This helps smooth out noisy

updates, avoid local minima, and accelerate convergence by maintaining a consistent direction in the parameter updates.

To improve the model's ability to generalize to new, unseen data, data augmentation was applied during training. Data augmentation is a common machine learning technique, which add some randomness and unpredictability to the dataset by randomly zoom, rotate, or flip the images, each with a 50% probability. These transformations simulate variations in real-world scenarios and help the model become more robust by preventing it from overfitting to the training data.

Model performance was evaluated using three metrics: **Intersection-over-Union (IoU)**, **precision**, and **recall**.

IoU is commonly used in image segmentation and object detection tasks (Ref. 15). It's calculated as the area of the intersection between the predicted and ground truth regions divided by the area of their union. The equation 2, where $A$ is the set of pixels in the predicted region and $B$ is the set of pixels in the ground truth region. As the function suggests, a value of 1.0 indicates a perfect match (complete overlap), while a value of 0 means no overlap at all. This makes IoU a reliable and interpretable metric for assessing how well a model identifies and localizes different objects in an image.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{2}$$

Precision and recall are standard evaluation metrics in classification and segmentation tasks. Precision measures the proportion of correctly predicted positive samples (true positives) out of all samples predicted as positive. Recall, on the other hand, measures the proportion of correctly predicted positive samples out of all actual positive samples in the ground truth. The definitions are shown in Equations 3 and 4, where **TP**, **FP**, and **FN** denote true positives, false positives, and false negatives, respectively. A high precision indicates few false positives, while a high recall indicates few false negatives. Both are crucial for evaluating model performance, especially in imbalanced datasets or safety-critical applications.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{4}$$

## RESULTS

### IoU comparison

The table 1 presents the IoU percentage for each class, comparing, from the top row, the U-YOLO, Attention U-Net, and Segmenter models. The first two values represent the raw mean IoU and the weighted mean IoU, which are computed by multiplying each singular score by to the proportion of pixels belonging to the respective class in the dataset.

**Table 1. IoU Comparison Across Different Classes.**

| IoU | U-YOLO | Attention U-Net | Segmenter |
|---|---|---|---|
| Mean | 54.8 | **57.1** | 38.3 |
| W. Mean | **58.7** | 56.8 | 50.9 |
| Water | **73.1** | 71.3 | 71.5 |
| B. No | 61.5 | **63.4** | 47.9 |
| B. Low | 48.6 | **52.6** | 33.0 |
| B. High | 49.1 | **52.2** | 27.9 |
| B. Total D. | 58.9 | **61.2** | 28.8 |
| Vehicle | 41.9 | **56.2** | 20.3 |
| Road Cl. | **70.6** | 68.9 | 56.7 |
| Road Bl. | **37.7** | 29.3 | 20.0 |
| Tree | **53.9** | 50.6 | 50.5 |
| Pool | 52.7 | **64.9** | 26.5 |

Analyzing the results, while Attention U-Net model is able to achieve the highest overall IoU, demonstrating strong performance across all classes, the new U-YOLO achieves the highest weighted IoU (58,7%), indicating that it correctly segments a larger amount of dataset's pixels. This behavior suggests that Attention U-Net is more precise in identifying small and less sampled classes, while U-YOLO performs better on dominant classes. The Segmenter model, instead, shows limited performances over this dataset, with a strong bias toward over-sampled classes, as shown by the large difference between weighted IoU and mean IoU.

Looking at specific classes, U-YOLO demonstrates the best segmentation results over more sampled classes, achieving an IoU value of 73.1% for water and 53.9% for trees, reinforcing its strength in capturing large, continuous regions. In particular, the water class is the best segmented class of all models, probably due to its well defined and distinct features.

In building segmentation, probably the most important classes for disaster assessment, Attention U-Net achieves the highest performance. The model is able to reach over 60% of IoU over untouched and completely destroyed buildings, and reaching a more moderate value in buildings with medium and major damage values, indicating that these types of edifices are highly confused between themselves, plausible due to the presence of common features. The U-YOLO model, while not being able to surpass the state-of-the-art performance, is still able to get close to them, with a difference of at most 4%. This suggests that Attention U-Net is best suited for distinguishing between different structural damage levels, while U-YOLO, although being highly competitive, does not reach the same level of segmentation accuracy in these categories, probably due to the lower magnitude of weight.

A different trend is shown for roads classification, where U-YOLO excels in segmenting both blocked roads and clear roads, with a peak of 70.6% IoU for the second one. A peculiar behavior can be observed for the classification of obstructed roads, where all the models are not able to reach sufficiently high performances. This behavior happens for two

main reasons: first, these roads are particularly covered with debris, bricks and tiles which, in most cases, can also be found along the road perimeter. This characteristic leads to various mispredictions when segmenting boundaries, leading to over-classification in areas with a large presence of rubble. Second, there are several cases in dataset where some streets are annotated as clear while reporting some sands or other types of detritus over them, leading to misclassification and a lack of correct generalization.

Lastly, vehicles and pools represent the classes with the fewest samples, respectively covering 0.3% and 0.1% of the total pixels. The obtained results show that Attention U-Net clearly outperforms the other models in these categories, effectively recognizing them thanks to its ability to capture fine details, even in underrepresented classes. While U-YOLO does not reach the same level of quality, it is still able to identify and segment these classes, achieving particularly good segmentation accuracy for pools, reaching 52.7%.

Considering now the Segmenter model, it clearly struggles to deliver effective performance across all classes. While it performs relatively well in dominant classes such as water, trees, and clear roads, its accuracy in other areas is insufficient, with some relevant classes achieving less than 30% IoU. This behavior is due to how the model performs feature extraction, where its high reliance on ViT patch processing and abrupt upsampling prevent it from correctly identifying key class features, leading to sparse and inaccurate segmentation.

### Precision and Recall

In this section, since the previous results highlights the limitations of Segmenter for this particular task, the following results' analysis will be done considering only Attention U-Net and U-YOLO models. The table 2 reports the precision and recall for all the classes highlighted before.

Looking at the precision's values, largely sampled classes such as water, trees achieve the highest precision, with both models performing well. In particular, U-YOLOn and Attention U-Net reach almost a precision of 80% for these classes. These high performances highlights that models can identify clearly most relevant features for these classes, producing a low amount of false positive annotations, probably due to the distinctive features of these classes.

Buildings instead present more variability. U-YOLOn performs slightly better for undamaged buildings, while Attention U-Net has is superior in damage categories. However, the precision difference between these two models is between 65% and 75% for both models, suggesting a similar level of uncertainty in building classification. For roads, U-YOLOn demonstrates stronger performance in detecting clear roads without huge confusions, whereas Attention U-Net is better in detecting blocked roads, with a very low number of false positives for this class.

For less sampled classes such as vehicles and pools, U-YOLO struggles to clearly identify them without confusion, reaching a precision below 60% for these classes, highlighting that this model struggles in identifying clearly small classes. On the other hand, Attention U-Net reaches a better level of precision especially in identifying pools in the dataset. This results further validate that Attention U-Net excelling in finer details due to its bigger amount of parameters compared to the YOLO model.

**Table 2. Precision and Recall Comparison Across Categories.**

| Precision | | Recall | |
|---|---|---|---|
| YOLO | At. U-Net | YOLO | At. U-Net |
| 76.0 | **83.0** | **89.1** | 83.5 |
| **74.4** | 71.4 | 79.2 | **85.1** |
| 67.7 | **70.2** | 66.3 | **67.8** |
| 64.8 | **66.1** | 69.9 | **71.2** |
| 69.4 | **72.5** | 78.4 | **79.6** |
| 46.2 | **61.0** | 77.6 | **85.0** |
| **83.7** | 80.3 | 80.9 | **83.0** |
| 42.5 | **56.4** | **70.3** | 37.8 |
| 84.2 | **87.0** | **90.2** | 83.9 |
| 59.6 | **76.5** | **86.9** | 82.2 |

Looking at the recall, the table 2 reinforces the trends observed in previous evaluations, with water, pools, and trees categories achieving the highest recall values. In particular, water and trees, being the most represented classes, maintain a recall above 80% across all models, indicating a low occurrence of false negatives for these categories.

In buildings classes, U-YOLOn and Attention U-Net are able to handle undamaged and destroyed buildings effectively, with Attention U-Net maintaining an advantage in more severely damaged categories. However, both models exhibit some inconsistencies, reflecting the trend to confuse buildings classes between themselves. For roads, clear roads are well segmented by all models, while Attention U-Net shows a significant drop in performance for blocked roads, barely reaching 40% recall, compared to U-YOLOn's 70%, highlighting a notable amount of false negative samples in this category.

Smaller and more detailed classes, such as vehicles, show the most pronounced differences. Attention U-Net achieves the highest recall for vehicles, confirming its effectiveness in detecting small objects, while Segmenter struggles significantly. A similar trend appears in pools, but in this case U-YOLOn outperform the other models.

Overall, Attention U-Net demonstrates strong performance across all classes, showing a remarkable ability to identify small classes and highly similar categories, such as buildings with various types of damages. In contrast, the U-YOLO model achieves better segmentation for dominant classes while using significantly fewer parameters.

### Models efficiency

Table 3 compares the models in terms of size and computational efficiency, reporting the number of Giga Floating Point

**Table 3. Hardware requirements comparison.**

| | GFLOPs | FPS | Params |
|---|---|---|---|
| **U-YOLO** | 42.85 | 16.59 | 3.3M |
| **Att. U-Net** | 2311.97 | 9.88 | 49.9M |
| **Segmenter** | 387.05 | 13.53 | 22.4M |

Operations (GFLOPs), frames per second (FPS), and total parameters.

U-YOLO clearly emerges as the most lightweight model, requiring only 43 GFLOPs while achieving significantly higher speed than the other models. This makes it particularly effective for real-time disaster response scenarios. A potential middle ground between the two extremes in parameter size is represented by the Segmenter model; however, its poor performance suggests that it is the least effective option in this evaluation. Thus, while Attention U-Net likely remains the state-of-the-art for damage assessment, the new model achieves highly competitive performance with a substantially lower number of parameters, offering a more efficient and practical trade-off for edge computing scenarios.

## ABLATION STUDY

To validate the model architecture and justify key design choices, this work presents a series of ablation studies, specifically designed to assess the impact of individual components and testing alternative configurations. Specifically, three key aspects were analyzed: the integration of an additional module in the decoder network, the replacement of the WCCE loss function with alternative formulations better suited for handling class imbalance, and the adoption of a different learning rate optimizer. These experiments provide valuable insights into the trade-offs between segmentation accuracy, model efficiency, and generalization ability, helping to refine the proposed approach and optimize its performance for real-world disaster assessment applications. In order to simplify the comparison, table 4 reports the mean and weighted mean IoU of the presented model compared with the baseline U-YOLO.

### Adding Attention Gate module

The first study conducted explores the integration of the AG module into the architecture, specifically to filter out irrelevant spatial information in the skip connections. This modification stems from the previously seen architecture of Attention U-Net, where AGs play a crucial role in enhancing feature relevance during decoding. However, in the initial design of U-YOLO, these components were deliberately omitted to maintain a lightweight and simplified decoder. The rationale behind revisiting their inclusion was to investigate whether this selective attention mechanism could improve segmentation accuracy, particularly for less prominent features.

Experimental results indicate that both the original and the AG-enhanced models achieve the same mean-IoU, suggesting comparable overall segmentation performance. Nevertheless, the original model shows better results in terms of weighted IoU, highlighting its strength in accurately segmenting the most frequent classes. On the other hand, the AG-enhanced version appears to be slightly more effective at capturing finer details, which may benefit the segmentation of smaller or less represented classes. Despite this, the marginal gains in detail recognition do not compensate for the increased computational complexity introduced by the AG module. Therefore, this modification is ultimately deemed not viable, as it does not offer a significant improvement in performance relative to its cost.

### Modify the loss function

The second set of studies aims to compare and evaluate the model using two different loss functions: (Ref. 16) and the Unified Focal Loss (Ref. 17).

The Tversky loss is a function specifically designed to address class imbalance in segmentation tasks. It modifies the traditional Dice loss by introducing two parameters, $\alpha$ and $\beta$, which control the relative weighting of false positives and false negatives, respectively (Ref. 11). This flexibility makes it particularly useful when dealing with datasets where certain classes are underrepresented. However, experimental results reveal that the Tversky loss struggles to achieve stable convergence in this context. Despite its theoretical potential, the lack of convergence makes it challenging to obtain consistent improvements, limiting its practical applicability in the current setup.

The Unified Focal loss is a more recent and sophisticated alternative that combines two advanced loss functions: the Tversky Focal loss, which builds upon the Tversky loss by incorporating a focusing parameter $\gamma$ to emphasize hard-to-classify examples, and the Categorical Focal loss, which applies a similar concept to the categorical cross-entropy (CCE) function (Ref. 11). This unified approach is designed to jointly address class imbalance and sample difficulty. The results are encouraging, showing that the Unified Focal loss not only converges reliably but also exhibits better performance. These findings suggest that, with further parameter tuning, this loss function holds the potential to surpass the original model's performance, particularly in scenarios requiring greater sensitivity to minority classes.

### Modify the LR optimizer

The final analysis investigates the impact of modifying the LR optimizer by replacing SGD with AdamW, an adaptive optimization algorithm widely adopted in modern deep learning applications (Ref. 18). Notably, AdamW is also the optimizer used in the original YOLO architecture, motivating its consideration in this context to ensure architectural consistency.

While SGD is valued for its simplicity and strong generalization capabilities, it lacks adaptive learning rate adjustments, which can be limiting in complex optimization landscapes. AdamW, on the other hand, introduces per-parameter learning rates and decoupled weight decay, enabling more flexible

**Table 4. Ablation studies results**

| | Mean | W. Mean |
|---|---|---|
| U-YOLO | **54.8** | **58.7** |
| U-YOLO-AG | **54.8** | 57.7 |
| Tversky Loss | 42.6 | 47.4 |
| Unified F. Loss | 53.4 | 54.7 |
| AdamW | 48.7 | 52.9 |

and potentially faster convergence through historical gradient tracking.

Despite these theoretical and practical advantages, experimental results show that AdamW consistently underperforms compared to the original SGD-based configuration in this specific segmentation task. This suggests that the stability and regularization offered by SGD are more suitable in the current setting. As a result, the use of AdamW—though standard in YOLO—is not considered beneficial for the modified architecture.

## CONCLUSION

The introduction of the new model established a strong foundation for balancing computational efficiency and segmentation accuracy. Its primary strength lies not only in significantly reduce resource consumption, requiring approximately 15 times fewer parameters than Attention U-Net, but also to deliver strong and competitive performance across various and different classes, it is even capable of outperforming state-of-the-art models in well-sampled and clearly defined classes. However, due to its compact size, the model has some limitations, such as reduced ability to refine boundaries and difficulties in precisely distinguishing between building types.

The assessment and evaluation of different configurations, including new modules and different training techniques, not only validates the model's structure, but also opened the way for potential future improvements. One promising direction is further refinement of the Unified Focal Loss function. Additionally, integrating alternative decoder structures, such as multi-scale feature extraction or more advanced skip connection mechanisms, could further enhance segmentation accuracy.

Overall, this study has achieved solid results, introducing an innovative and efficient model for disaster assessment in edge computing environments.

Gianluca Parnisari gianluca.parnisari@txtgroup.com Enrico Targhini enrico.targhini@polimi.it

## ACKNOWLEDGMENTS

I'd like to thank TXT-Group for the opportunity to work in an incredible and stimulating environment. A special thanks to Gianluca and Raffaele, who helped me developing this architecture and guided me during this journey.

A thank also to professor Piero Fraternali, who helped me for technical details and organization.

A last acknowledge also to Aleksandra Pavlovic, who's always been by my side and I know will be there when I need her.

## REFERENCES

1. Akhyar, A., Zulkifley, M. A., and Lee, J., "Deep artificial intelligence applications for natural disaster management systems: A methodological review," *Ecological Indicators*, Vol. 163, 2024, pp. 112067. https://doi.org/10.1016/j.ecolind.2024.112067

2. Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P., "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, Vol. 86, Dec. 1998, pp. 2278–2324. https://doi.org/10.1109/5.726791

3. Ronneberger, O., Fischer, P., and Brox, T., "U-Net: Convolutional Networks for Biomedical Image Segmentation," arXiv preprint arXiv:1505.04597, 2015. URL: https://arxiv.org/abs/1505.04597

4. Mehta, S., Rastegari, M., Caspi, A., Shapiro, L., and Hajishirzi, H., "ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation," arXiv preprint arXiv:1803.06815, 2018. URL: https://arxiv.org/abs/1803.06815

5. Xu, G., Li, J., Gao, G., Lu, H., Yang, J., and Yue, D., "Lightweight Real-time Semantic Segmentation Network with Efficient Transformer and CNN," arXiv preprint arXiv:2302.10484, 2023. URL: https://arxiv.org/abs/2302.10484

6. Khanam, R., and Hussain, M., "YOLOv11: An Overview of the Key Architectural Enhancements," arXiv preprint arXiv:2410.17725, 2024. URL: https://arxiv.org/abs/2410.17725

7. Rahnemoonfar, M., Chowdhury, T., and Murphy, K., "RescueNet: A High Resolution UAV Semantic Segmentation Dataset for Natural Disaster Damage Assessment," *Scientific Data*, No. 913, 2023. https://doi.org/10.1038/s41597-023-02799-4

8. Oktay, O., Schlemper, J., and Le Folgoc, L., "Attention U-Net: Learning Where to Look for the Pancreas," *CoRR*, Vol. abs/1804.03999, 2018. URL: http://arxiv.org/abs/1804.03999

9. Strudel, R., Garcia, R., Laptev, I., and Schmid, C., "Segmenter: Transformer for Semantic Segmentation," *CoRR*, Vol. abs/2105.05633, 2021. URL: https://arxiv.org/abs/2105.05633

10. FEMA, "FEMA Preliminary Damage Assessment Guide," 2020. URL: https://www.fema.gov/sites/default/files/2020-07/fema_preliminary-disaster-assessment_guide.pdf

11. Terven, J., Cordova-Esparza, D. M., and Ramirez-Pedraza, A., "Loss Functions and Metrics in Deep Learning," arXiv preprint arXiv:2307.02694, 2024. URL: https://arxiv.org/abs/2307.02694

12. Jegham, N., Koh, C. Y., Abdelatti, M., and Hendawi, A., "Evaluating the Evolution of YOLO (You Only Look Once) Models: A Comprehensive Benchmark Study of YOLO11 and Its Predecessors," arXiv preprint arXiv:2411.00201, 2024. URL: https://arxiv.org/abs/2411.00201

13. Mao, A., Mohri, M., and Zhong, Y., "Cross-Entropy Loss Functions: Theoretical Analysis and Applications," arXiv preprint arXiv:2304.07288, 2023. URL: https://arxiv.org/abs/2304.07288

14. Bottou, L., "Stochastic Gradient Descent Tricks," in *Neural Networks: Tricks of the Trade: Second Edition*, edited by Montavon, G., Orr, G. B., and Müller, K.-R., Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 421–436. https://doi.org/10.1007/978-3-642-35289-8_25

15. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., and Savarese, S., "Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression," arXiv preprint arXiv:1902.09630, 2019. URL: https://arxiv.org/abs/1902.09630

16. Salehi, S. S. M., Erdogmus, D., and Gholipour, A., "Tversky loss function for image segmentation using 3D fully convolutional deep networks," arXiv preprint arXiv:1706.05721, 2017. URL: https://arxiv.org/abs/1706.05721

17. Yeung, M., Sala, E., Schönlieb, C.-B., and Rundo, L., "Unified Focal loss: Generalising Dice and cross entropy-based losses to handle class imbalanced medical image segmentation," arXiv preprint arXiv:2102.04525, 2021. URL: https://arxiv.org/abs/2102.04525

18. Loshchilov, I., and Hutter, F., "Decoupled Weight Decay Regularization," arXiv preprint arXiv:1711.05101, 2019. URL: https://arxiv.org/abs/1711.05101